UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE

FAKULTA PRÍRODNÝCH VIED

IDENTIFIKÁCIA SPRÁV TYPU „FAKE NEWS“ Z TEXTOV V SLOVENSKOM JAZYKU

DIPLOMOVÁ PRÁCA

2021 **Bc. BENEDIKT PECUCH**

UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE

FAKULTA PRÍRODNÝCH VIED

IDENTIFIKÁCIA SPRÁV TYPU „FAKE NEWS“ Z TEXTOV V SLOVENSKOM JAZYKU

DIPLOMOVÁ PRÁCA

Študijný odbor:	9.2.9 Aplikovaná informatika
Študijný program:	Aplikovaná informatika
Školiace pracovisko:	Katedra informatiky
Školiteľ:	doc. PaedDr. Jozef Kapusta, PhD.

Nitra 2021 Bc. Benedikt Pecuch

Univerzita Konštantína Filozofa v Nitre
Fakulta prírodných vied

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Benedikt Pecuch
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: Diplomová práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Identifikácia správ typu "fake news" z textov v slovenskom jazyku

Anotácia: Falošné správy (tzv. Fake News) sú v súčasnosti strašiakom vyspelého sveta. Popri náraste využívania sociálnych sietí, hlavne na komunikáciu, pozorujeme aj vysoký nárast šírenia falošných správ, hoaxov a iných poloprávd v periodikách, ako aj v spoločnosti.
Cieľom práce je vytvoriť návrh metódy identifikácie falošných správ. V práci budú extrahované základné charakteristiky správ a budú vybrané vhodné metódy z dostupných metód strojového učenia. Práca bude zameraná na identifikáciu falošných správ v slovenských textoch, pričom bude potrebné vyriešiť niekoľko problémov typických pre flektívne typy jazykov, do skupiny ktorých patrí aj slovenčina.

Charakter práce: aplikačný.

Požiadavky na obsah podľa charakteru práce (nemyslí sa tým šablóna, tá je daná bez ohľadu na charakter práce): popis riešeného problému, návrh systému / hardvérového riešenia a pod. (modely, ..), metodika vývoja/tvorby, implementácia, popis vytvoreného riešenia, testovanie.

Školiteľ: doc. PaedDr. Jozef Kapusta, PhD.
Oponent: prof. RNDr. Michal Munk, PhD.
Katedra: KI - Katedra informatiky

Dátum zadania: 01.10.2019

Dátum schválenia: 15.10.2019

RNDr. Ján Skalka, PhD., v. r.
vedúci/a katedry

POĎAKOVANIE

Touto cestou vyslovujem poďakovanie doc. PaedDr. Jozefovi Kapustovi, PhD. za usmernenie, pripomienky a odbornú pomoc pri vypracovaní diplomovej práce.

ABSTRAKT

PECUCH, Benedikt: Identifikácia správ typu „fake news“ v textoch slovenského jazyka. [Diplomová práca]. Univerzita Konštantína Filozofa v Nitre. Fakulta prírodných vied. Školiteľ: doc. PaedDr. Jozef Kapusta, PhD. Stupeň odbornej kvalifikácie: Magister odboru Aplikovaná informatika. Nitra: FPV, 2021. 52 s.

Falošné správy (tzv. Fake News) sú v súčasnosti strašiakom vyspelého sveta. Popri náraste využívania sociálnych sietí, hlavne na komunikáciu, pozorujeme aj vysoký nárast šírenia falošných správ, hoaxov a iných poloprávd v periodikách, ako aj v spoločnosti. Cieľom práce je vytvoriť návrh metódy identifikácie falošných správ. V práci budú extrahované základné charakteristiky správ a budú vybrané vhodné metódy z dostupných metód strojového učenia. Práca bude zameraná na identifikáciu falošných správ v slovenských textoch, pričom bude potrebné vyriešiť niekoľko problémov typických pre flektívne typy jazykov, do skupiny ktorých patrí aj slovenčina.

Teoretická časť práce je zameraná na definíciu pojmu falošné správy, históriu falošných správ, vlastnosti a typy falošných správ. V praktickej časti ponúkame analýzu súčasného stavu, metodiku výskumu a implementáciu navrhovaného programu v jazyku Python.

Kľúčové slová: Fake news. Hoax. Detekcia dezinformácií. Binárne stromy. Neurónové siete. Flektívne jazyky. Python.

ABSTRACT

PECUCH, Benedikt: Identification of "fake news" type messages in Slovak language texts. [Thesis]. Constantine the Philosopher University in Nitra. Faculty of Natural Sciences. Supervisor: doc. PaedDr. Jozef Kapusta, PhD. Degree of professional qualification: Master of Applied Informatics. Nitra: FPV, 2021. 52 p.

Fake News is currently serious issue of the developed world. During increation of the usage of social networks, especially for communication, we also observe a high increase in the spread of false news, hoaxes and other half-truths in periodicals as well as in society. The aim of the work is to create a proposal for a method of identifying false messages. The basic characteristics of messages will be extracted in the work and suitable methods will be selected from available machine learning methods. The work will focus on the identification of false messages in Slovak texts, while it will be necessary to solve several problems typical of inflected types of languages, to which Slovak also belongs.

The theoretical part of the work is focused on the definition of the concept of fake news, the history of fake news, properties and types of fake news. In the practical part we offer an analysis of the current state, research methodology and implementation of the proposed program in Python.

Keywords: Fake news. Hoax. Detection of disinformation. Binary trees. Neural networks. Inflected languages. Python.

Obsah

ÚVOD

„Pravda a lož sú ako jednovaječné dvojčatá. Často býva problémom rozoznať jednu od druhej.“

Lech Przeczek, český básnik a spisovateľ

Koncom minulého storočia svetoví prognostici predpovedali, že tretie tisícročie sa bude niesť v znamení revolúcie informačnej spoločnosti. Neuveriteľne rýchly vývoj informatiky či už v oblasti hardware alebo software zapríčinil, že mobilné telefóny doplnené o množstvo doplnkových funkcií sa stali bežnou súčasťou života ľudí na celom svete. Následný rozvoj sociálnych sietí umožnil užívateľom pracovať s informáciami na celkom novej vyššej úrovni. Ide o rýchly, takmer okamžitý, prístup k informáciám a možnosť ich ďalšieho spracovania a najmä ich zdieľania. Funkciou zdieľania, ktoré bolo umožnené každému bežnému používateľovi, sa tento stal prakticky spolutvorcom a šíriteľom informácií. Okrem samozrejmých prínosov táto skutočnosť viedla aj k tomu, že sa internetom začali šíriť neoverené, nepravdivé a zmätočné informácie vo forme falošných správ.

Podľa štatistík sa falošné správy šíria po internete šesť krát rýchlejšie ako pravdivé správy. Rovnako s tým ako rastie počet používateľov na sociálnych sieťach je zrejmé, že klasické manuálne spôsoby kontroly a nápravy dopadov falošných správ sú v dnešnej dobe už nevyhovujúce. Udalosti 6.januára 2021 v Capitole po prezidentských voľbách v USA jasne potvrdili, že šírenie dezinformácií dosiahlo kritickú hodnotu. V prípade, že sa nepodniknú účinné kroky v zamedzení ich šírenia, sú ohrozené základné demokratické princípy našej spoločnosti ako takej.

Cieľom diplomovej práce je vytvoriť návrh metódy identifikácie falošných správ. Práca bude zameraná na identifikáciu falošných správ v slovenských textoch, pričom bude potrebné vyriešiť niekoľko problémov typických pre flektívne typy jazykov, do skupiny ktorých patrí aj slovenčina.

Diplomová práca pozostáva z teoretickej časti, analýzy súčasného stavu a praktickej časti. Práca je rozdelená na štyri kapitoly. Prvá kapitola diplomovej práce je zameraná na definíciu, vysvetlenie, históriu falošných správ a ich charakteristiku. V druhej kapitole popisujeme metódy detekcie falošných správ a ponúkame analýzu dostupných prác zaoberajúcich sa touto tematikou. Tretia kapitola je venovaná metodike

výskumu, ktorý budeme realizovať. V poslednej štvrtej kapitole navrhujeme spôsob detekcie falošných správ pomocou dostupných metód strojového učenia, analyzujeme a vyhodnocujeme získané výsledky.

1 FALOŠNÉ SPRÁVY

Falošné správy alebo po anglicky fake news sa stali súčasťou našej reality v informačnom veku. Keďže ich negatívny dopad na spoločnosť je neodškriepiteľný je nevyhnutné venovať sa im zo všetkých možných pohľadov od sociologického, psychologického, ekonomického, politického až po technický pohľad.

1.1 DEFINÍCIA POJMU FALOŠNÉ SPRÁVY

Pojem falošná správa (po angl. fake news) sa používa vo význame vymyslenej správy. Táto správa zvyčajne obsahuje nepravdivé alebo skreslené informácie. Šírené môžu byť neúmyselne (takzvaná misinformácia) alebo úmyselne (takzvaná dezinformácia) a tiež podľa toho, či sú založené na názore alebo faktoch. Najpoužívanejším médiom na ich šírenie v súčasnosti je internet a sociálne siete, ktoré potlačili do úzadia dosah klasických printových médií. Najrozšírenejšími sociálnymi sieťami, ktoré sú takto zneužívané sú Facebook, Twitter a Youtube. Dôvodom ich vytvárania je získať osobný, ekonomický alebo politický profit.

Pod osobným profitom alebo výhodou sa myslí získanie pozornosti a dôležitosti zväčša u konkrétnych osôb, ktoré trpia pocitom svojej nedôležitosti alebo naopak trpia pocitom svojej výnimočnosti a nadradenosti. Zdieľaním nepravdivých správ a ich zápalistým komentovaním si zväčša snažia podľa psychológov kompenzovať nedostatok sebavedomia, alebo potlačiť pocit strachu napríklad z prebiehajúcej pandémie Covid 19 (Jurkovič, Čavojová, Brezina, 2019).

Ekonomické výhody vedú k ich tvorbe z toho dôvodu, že tieto správy sú prezentované ako senzačné a tým zabezpečujú vysokú sledovanosť, ktorá prináša vysoké príjmy z reklamy alebo predaja propagovaných výrobkov. Typickým príkladom je na Slovensku stránka *Badatel.net*. Táto sa popri zverejňovaní senzačných a kontroverzných článkov snaží predávať rôzne vitamínové prípravky a doplnky stravy, ktorých predaj je priamo úmerný počtu čitateľov ich stránky. Navyše algoritmy sociálnych sietí sú navrhnuté tak, aby sa maximalizoval čas užívateľov pri mobile alebo počítači a aby sa automaticky ponúkali výrobky a služby, ktoré boli v minulosti vyhľadávané (Orlowski, 2020).

Tretím dôvodom je spomínaný politický profit. Najznámejšími príkladmi, kde šírením falošných správ získali jej šíritelia politickú výhodu je Brexit v Anglicku a voľby v USA z roku 2016. Vo Veľkej Británii počas referenda o vystúpení z Európskej únie

došlo k systematickému ovplyvňovaniu účastníkov referenda cez špecializované nástroje firiem ako napríklad *Cambridge Analytica* a to využitím osobných dát zo sociálnych sietí Facebook a Twitter. Podobne sa angažovala táto firma aj pri podpore zvolenia Donalda Trumpa, pričom sa v jeho prospech šírila najzdieľanejšia vymyslená správa týchto volieb a to, že jeho zvolenie za prezidenta USA údajne podporuje samotný pápež. Dlhodobo je známe z výročných správ slovenských spravodajských služieb, že na našom území sa angažujú ruské prokremeľské informačné média, ktoré sa snažia v rámci tzv. hybridnej vojny o vytvorenie nedôvery voči Európskej únii a podporujú snahy o vystúpenie z tohto spoločenstva (Správa o činnosti Slovenskej informačnej služby, 2020).

1.2 HISTÓRIA FALOŠNÝCH SPRÁV

História šírenia falošných správ siaha až do staroveku. Už v Grécku a Egypte sa zaznamenali jedny z prvých prípadov šírenia nepravdivých správ. Najznámejšie prípady však zaznamenávame v posledných niekoľko sto rokov. V 18. storočí sa dala panovníčka Mária Terézia zaočkovať proti ovčím kiahňam, aby vyvrátila klamstvá, ktoré sa o očkovaní šírili.

Začiatkom 20.storočia v Rusku za účelom vyburcovania protižidovských nálad vytvorila pravdepodobne cárska tajná služba dokument Protokoly sionských mudrcov. V ňom bol popísaný údajný plán Židov na ovládnutie sveta. Tento bol následne použitý v nacistickom Nemecku na obhajobu zákonov, ktoré obmedzovali práva a slobody tohto etnika. Dodnes sa téma ovládnutia sveta niekoľkokrát opakovala až do dnešných dní v podobe hoaxov o Georgovi Sorosovi a Billovi Gatesovi (Michelis, 2004).

Zo Škótska sú známe správy o príšere v jazere Loch Ness, kde v roku 1934 sa kvôli fotografii britského lekára rozpútali vášnivé debaty o jej pravosti. Pravda vyšla na svetlo sveta až o 60 rokov neskôr, keď sa nevlastný syn fotografa priznal, že príšera bola v skutočnosti hračka.

V roku 1938 v USA vysielal známy režisér Orson Welles rozhlasovú hru Vojna svetov o napadnutí našej planéty mimozemšťanmi z Marsu. Keďže hra bola napísaná formou sugestívnych správ, množstvo ľudí sa domnievalo, že ide o skutočnosť a to spôsobilo u nich paniku a strach.

Keď sa pozrieme na falošné správy v období počas druhej svetovej vojny narazíme na jednu z najdôležitejších udalostí a tou bolo vylodenie spojencov v Stredomorí. Komplikovanou konšpiráciou s utopeným britským majorom, ktorý nikdy

neexistoval, sa podarilo presvedčiť Nemcov, že k vylodeniu dôjde na Sicílii. Na základe falošnej správy Nemci presunuli svoje vojenské sily, čo veľmi pomohlo Američanom a Britom.

Z povojnových falošných správ patrí medzi najznámejšie pitva mimozemšťana z Roswellu. V roku 1947 došlo v americkom štáte Nové Mexiko k udalosti, ktorá spôsobila šírenie správy o páde rakety z vesmíru. Čím viac americká vláda tieto fámy vyvracala, tým viac sa rozširoval počet ľudí, ktorí verili nepravde. Na potvrdenie tejto udalosti bol v roku 1995 odvysielaný film o pitve mimozemšťana. Ako sa neskôr ukázalo išlo o podvod a film nebol natočený v Roswelli, ale v Londýne, pričom údajné telo mimozemšťana vytvoril sochár (Bartholomew, Radford, 2003).

Na základe týchto ale i ďalších príkladov z histórie môžeme konštatovať, že množstvo hoaxov a falošných správ, ktoré sa dnes šíria, majú svoj základ v minulosti a dochádza len k ich znovuobjaveniu a prispôsobeniu k aktuálnej situácii.

1.3 VLASTNOSTI A TYPY FALOŠNÝCH SPRÁV

Falošné správy vieme rozoznať od iných správ na základe ich typických vlastností. Tieto vlastnosti môžeme pomenovať aj ako črty, ktoré vieme rozčleniť nasledovne:

a) obsahové črty

b) črty v sociálnom kontexte

Obsahové črty sú vlastne informácie, ktoré sa nachádzajú v samotnom obsahu správ, pričom tento obsah vieme rozčleniť na jednotlivé časti :

- titulok na upútanie pozornosti konzumenta správ
- telo správy, ktoré obsahuje samotnú správu
- grafická príloha, ktorú tvoria obrázky prípadne videá
- zdroj, ktorý identifikuje autora správy

Pre titulok falošných správ je typické, že obsahuje viac informácií ako titulok obyčajnej správy a má pôsobiť až prehnane pútavo alebo senzačne (Shu, Sliva, Wang, Tang, 2017).

Črty môžeme rozdeliť aj podľa toho, či ich identifikujeme na úrovní slov, viet alebo celej správy. Medzi črty na úrovní slov patrí priemerná dĺžka slova, počet znakov v slove, samotný počet slov alebo výskyt slov podľa ich dĺžky. Na základe toho akú emóciu vyvolávajú tieto správy, môžeme im tak podľa špecializovaných slovníkov priradiť konkrétny sentiment. K črtám na úrovni viet patrí použitie fráz alebo slangu,

počet slovných druhov, použitie interpunkcie alebo početnosti tzv. stopslov. Stopslová nemajú významovú hodnotu, keďže ide o predložky, zámená , spojky a podobne. Obrázky a videá falošnej správy majú za úlohu emočne podporiť cieľ, ktorý sa snaží správa dosiahnuť.

Črty v sociálnom kontexte sú informácie, ktoré sú doplnkové k obsahovým. Ide o črty užívateľov, ktorí zdieľajú správy. Zvyčajne ide o užívateľov, ktorých účty sú v priemere mladšie, lebo ide buď o tzv. trollov s falošnými účtami alebo sú to účty generované automaticky. Medzi ďalšie črty patrí previazanosť správ a zdrojov s odkazmi. Zistilo sa, že falošné účty a správy navzájom na seba odkazujú a tým sa snažia zvýšiť svoju sledovanosť. Je všeobecne známe, že falošné správy sa šíria niekoľkonásobne rýchlejšie ako obyčajné správy. Je to kvôli tomu, že nepravdivé správy sa účelovo tvoria tak, aby boli pútavejšie už v titulkoch, a tým sa snažia dosiahnuť svoj cieľ (Kumar, Shah, 2018).

2 ANALÝZA SÚČASNÉHO STAVU

Hromadné využívanie sociálnych médií vyvolalo v našej spoločnosti aj nebývalý nárast šírenia falošných správ, hoaxov a iných poloprávd v periodikách, ako aj v spoločnosti. Je to ešte zreteľnejšie, teraz v čase krízy, akou je pandémia COVID-19. Z toho je zrejme nutné poukázať na dôležitosť ich rozlišovania a odhaľovania.

Dôležitou súčasťou v boji proti falošným správam sú stránky, ktoré sa zaoberajú manuálnym overovaním ich faktov (napr. Fact Check, Hoax Slayer). Aj napriek tomu, že sa o problematike falošných správ a dezinformácií v súčasností veľa hovorí a píše, touto problematikou sa na Slovensku venuje len obmedzené množstvo odborníkov, výskumníkov a akademikov. Ako vyplýva z poslednej štúdie GLOBSEC Trends 2018 Central Europe: One Region, Different Perspectives, Slováci sú ako národ pomerne náchylní k tomu, aby verili konšpiračným teóriám a hoaxom. Podľa tejto štúdie až 53% Slovákov verí výroku, že tajné spoločenstvá ovládajú svetovú politiku a 52% si myslí, že Židia majú príliš veľkú moc a tajne ovládajú vlády a medzinárodné organizácie. Ďalšia zaujímavá skutočnosť je, že aj keď 68% ľudí narazilo na sociálnych sieťach na dezinformáciu, iba 9% z nich ju nahlásilo. Na základe toho vzniká hneď niekoľko otázok. Prečo Slováci častejšie podliehajú falošným správam? Prečo ich aj napriek ich rozlíšeniu nenahlásia? Prečo ich dokážu akceptovať?

V tejto súvislostí je tiež dôležité poukázať na to, že na Slovensku máme veľmi málo zozbieraných dát a relevantných štúdií, ktoré by sa zaoberali strategickou komunikáciou, informačnými operáciami alebo stratégiou hybridnej vojny. Neexistuje u nás dostupný nástroj, ktorý by automaticky určil, či článok publikovaný na spravodajskom serveri šíri dezinformácie alebo nie. Neexistujú u nás ani inštitúcie venujúce sa tejto problematike na centralizovanej úrovni. Žuborová a Borárosová (2019) ďalej uvádza že: „*Boj proti dezinformáciám, hoaxom a falošným správam je v slovenskom prostredí roztrieštený, nekoncepčný a nekoordinovaný.“* V súčasnosti existujú iba tri verejné inštitúcie, v porovnaní z nemalým počtom známych dezinformačných webov, ktorých je na Slovensku už teraz uvedených vyše 160 a ktoré neustále pribúdajú. Práve Ministerstvo vnútra, Ministerstvo zdravotníctva a Ministerstvo obrany pomocou svojich stránok na Facebooku bojujú proti dezinformáciám. Aktuálnou informáciou v danej oblasti je, že v novembri 2020 bol predložený materiál *Koordinovaný mechanizmus odolnosti Slovenskej republiky voči informačným operáciám*, ktorý má po prvýkrát vytvoriť normatívny rámec pre v boji proti dezinformáciám pre orgány štátnej správy.

Tento materiál nadväzuje na strategický materiál, prijatý Európskou komisiou v decembri 2018, *Akčný plán boja proti dezinformáciám*.

Z horeuvedeného vyplýva, že v najbližšom období vzrastie potreba na Slovensku uviesť a aplikovať do praxe sofistikované automatizované nástroje, ktoré by uľahčili prácu tým, ktorí sa budú profesionálne venovať odhaľovaniu a potieraniu falošných správ a dezinformácií (Žúborová, Borárosová, 2019).

2.1 METÓDY DETEKCIE

V nasledujúcej časti stručne opíšeme niektoré z používaných metód, ktorých autori sa zaoberali detekciou falošných správ z rôznych uhlov pohľadu. V princípe sa na riešenie tohto problému používajú metódy manuálne a automatizované.

Ako sme už spomínali manuálnym metódam patria tzv. fact-checking. Na Slovensku ho realizuje renomovaná tlačová agentúra AFP z Francúzka , ktorú si najala najväčšia sociálna sieť Facebook. Tejto metóde sa najintenzívnejšie venujú zo strany štátnych inštitúcií facebooková stránka Ministerstva vnútra – Hoaxy a podvody – polícia SR. Veľmi aktívne pôsobia u nás organizácie tretieho sektora ako Konšpirátori.sk, Lovci šarlatánov, Demagog.sk, Infosecurity.sk alebo Antipropaganda. Pre našu prácu bude dôležitá aktivita občianskeho združenia Konšpirátori.sk, ktoré sa systematicky venuje bodovému ohodnocovaniu webstránok.

Vzhľadom na extrémne pribúdajúce množstvo falošných správ nie je manuálny spôsob ich odhaľovania už postačujúci. Preto v súčasnosti vstupuje do popredia dôležitosť rozvoja automatizovaných prístupov a metód. Ponúkame prehľad metód od viacerých autorov, ktorí sa problémom detekcie zaoberali:

- Hĺbková analýza syntaxe

Hlbokú syntax je možné analyzovať pomocou Pravdepodobnostnej bezkontextovej gramatiky (Probabilistic context-free grammar). Podstatné mená, slovesá a podobne sa prepisujú do syntaktických lingvistických častí. Štruktúry syntaxe sú popísané zmenou viet na syntaktické stromy. Pravdepodobnosti sú potom priradené k syntaktickému stromu (Feng, Banerjee, Choi, 2012).

- Klastrová analýza

Tento postup zoskupuje podobné spravodajské správy na základe normalizovanej frekvencie vzťahov a po výpočte centier klastrov skutočných a falošných správ, je tento model schopný zistiť klamnú hodnotu nového článku. Princíp je založený na

súradnicových vzdialenostiach, kde sa počítajú jeho euklidovské vzdialenosti od klastrov skutočných a falošných správ. Nevýhodou tohto prístupu však môže byť ak sa použije na falošné spravodajské články, ktoré sú relatívne nové a teda podobné súbory spravodajských príbehov ešte nemusia byť dostupné (Conroy, Rubin, Chen, 2016).

- Metódy založené na prediktívnom modelovaní

Zistenie falošných správ sa dá dosiahnuť aj metódami založenými na prediktívnom modelovaní. Jedným z typov by bol model logistickej regresie. V tomto modeli kladné koeficienty zvyšujú pravdepodobnosť pravdy, zatiaľ čo negatívne zvyšujú pravdepodobnosť podvodu (Shivam, Pradeep, 2018).

- Kontrola faktov

Kontrola faktov je forma štúdie falošných správ, ktorá sa zameriava na hodnotenie pravdivosti správ. Existujú dva typy kontroly faktov, a to manuálna a automatická.

- Detekcia falošných správ založená na propagácii

Detekcia na základe šírenia analyzuje šírenie falošných správ. Štruktúra podobná stromu sa často používa na predstavenie kaskády falošných správ. Ukazuje šírenie falošných správ na sociálnych sieťach používateľmi. Koreňový uzol predstavuje používateľ, ktorý zverejňuje falošné správy. Zvyšok uzlov predstavuje používateľov, ktorí správy následne šíria ich preposielaním alebo zverejnením. Za predpokladu, že kaskády falošných správ sa líšia od kaskád skutočných správ, bolo na detekciu falošných správ použité výpočtové rozdiely vo vzdialenosti medzi dvoma kaskádami.

- Analýza falošných správ založená na dôveryhodnosti

Tento prístup sa zameriava na falošné správy na základe spravodajských a sociálnych informácií. Napríklad intuitívne je pravdepodobné, že spravodajský článok uverejnený na nespoľahlivých webových stránkach a preposlaný nespoľahlivými používateľmi bude falošná správa, ako správa zverejnená autoritatívnymi a dôveryhodnými používateľmi. Inými slovami, tento prístup sa zameriava na zdroj spravodajského obsahu. Perspektíva dôveryhodnosti štúdia falošných správ sa ako taká obvykle prekrýva so štúdiou falošných správ založenou na propagácii (Zhou, Zafarani, 2018).

- Analýza účtov

Pri tejto metóde bol vyvinutý detekčný model, ktorý rozlišoval účty botov. Kategorizované boli 3 skupiny - ľudia, roboti a kyborgovia. Bol vybudovaný systém so

4 funkciami analýzy, a to entropická miera, detekcia spamu, vlastnosti a rozhodovanie účtu. Táto metóda úspešne identifikovala „ľudskú“ triedu s presnosťou 96%.

- Doplnky prehliadača

Doplnky prehliadača dokážu na webových stránkach sociálnych médií zistiť klamlivý obsah, ako je návnada na klikanie, zaujatosť a konšpiračná teória. Jedným z príkladov je „Fake News Detector“, ktorý využíva techniku strojového učenia na zhromažďovanie základných údajov. Zbierané údaje sa navyše používajú na vylepšenie a umožnenie programu učiť sa. Ďalším príkladom vyvinutého doplnku prehliadača bol ten, ktorý vytvorili štyria študenti univerzity počas hackathonu na Princetonskej univerzite. Tento systém vykonáva analýzu informačného kanála používateľa v reálnom čase a analyzuje kľúčové slová, obrázky a zdroje a upozorňuje používateľa na zverejnenie alebo zdieľanie potenciálne nepravdivého obsahu (Figueira, Guimaraes, Torgo, 2018).

2.2 HĹBKOVÁ ANALÝZA SYNTAXE

Vzhľadom na cieľ našej práce, ktorým je vytvoriť návrh metódy identifikácie falošných správ v slovenčine, zameriame sa na hĺbkovú analýzu syntaxe. Pre lepšie porozumenie jazyka s ktorým budeme pracovať, krátko opíšeme kategórie jazykov podľa tvorenia slov a slovných tvarov tak, ako ho v odbornej literatúre delí Oravec J. a kol. (1988).Tento autor vyčleňuje tieto kategórie:

- *Aglutinačný typ jazyka* - jednotlivé gramatické kategórie sa tvoria osobitnou príponou, takže v gramatickom tvare sa k základu prilepí (aglutinuje) niekoľko prípon. Aglutinačný typ jazyka sa vyznačuje veľkým počtom prípon, ale aj ich nemennosťou v dôsledku čoho sa vytvárajú mnohopočetné odvodeniny podstatných mien a slovies. K ďalším znakom týchto jazykov patrí jednotné časovanie a menej časté používanie vedľajších viet Patria tu napríklad maďarčina, turečtina, fínčina ale aj japončina.
- *Flektívny typ jazyka* - typ jazyka, v ktorom sa podstatné mená skloňujú podľa gramatických kategórií rodu, čísla a pádu a slovesá sa časujú podľa osoby, čísla, rodu, spôsobu a času. K tomuto typu jazyka patrí väčšina slovanských ako bulharčina, poľština a slovenčina, ale aj latinčina, gréčtina.
- *Introflektívny typ jazyka* – ide taktiež o ohybný jazykový typ, s tým rozdielom, že gramatické tvary sú tvorené formálnymi zmenami hlások uprostred základu slova. Tento typ jazyka prevláda hlavne v arabčine, hebrejčine a nemčine.

- *Analytický (izolačný) typ jazyka* - tvary sa tvoria pomocou osobitných slov alebo častíc. Slovosled určuje význam vety preto zohráva významnú úlohu. Typickým znakom analytických jazykov je nemennosť plnovýznamových slov. Tvary sa pri menách tvoria pomocou predložiek, pri slovesách rôznymi pomocnými slovesami. Do tohto typu zaraďujeme západoeurópske jazyky ako angličtina a francúzština.
- *Polysyntetický typ jazyka* – patrí sem čínština a iné východoázijské jazyky. Slová sa neskloňujú ani nečasujú. Typické pre tento typ je tvorenie nových slov skladaním. V tomto type jazyka sa strácajú rozdiely medzi plnovýznamovými a neplnovýznamovými slovami. Dôraz sa kladie na intonáciu.

Slovenčina je flektívny jazyk s prvkami iných typov jazyka. Flektívnosť sa tu prejavuje rozdielmi v sústave tvarov, rozdielmi v skloňovaní podstatných a prídavných mien, číslovieк, zámen. Veľký počet vzorov slovenských slovies je taktiež výraznou flektívnou črtou. Slovenčina sa vyznačuje tvorením veľkého množstva vedľajších viet. Morfologická stavba slovenčiny je preto prevažne flektívna, ale dopĺňa sa aglutinačnými a analytickými prvkami pri ohybných slovných druhoch a polysyntetickými prvkami pri neohybných slovných druhoch (Oravec, 1988).

2.3 ANALÝZA DOSTUPNÝCH PRÁC S TÉMATIKOU DETEKCIE FALOŠNÝCH SPRÁV

V súčasnosti existuje niekoľko štúdii i prác, ktoré sa zaoberajú detekciou „fake news“ v rôznych typov jazykoch. Ponúkame pohľad na niektoré z nich.

Ruchansky a kol. (Ruchansky, Seo and Liu, 2017) pozorovali tri dôležité vlastnosti falošných správ: obsah správy, zdroj správy a reakcia čitateľa. V predchádzajúcich prácach sa autori zameriavali iba na jednu konkrétnu vlastnosť, ale v kalifornskej štúdii sa však pokúsili zamerať na všetky súčasne. Vytvorili model s názvom CSI (Capture, Scoring, Integration), ktorý sa skladá z troch modulov.

Prvý modul analyzuje obsah článku a reakciu používateľa. Na zachytenie závislostí používa neurónovú sieť. Druhý modul sa učí charakteristiky zdrojov na základe správania používateľov. Posledný modul potom kombinuje výsledky prvých dvoch modulov a urobí konečné hodnotenie toho, či je článok zavádzajúci. Vytvorený model sa trénoval na veľkom počte existujúcich správ, aby mohol správne rozlišovať medzi pravdivými a nepravdivými správami. Na tento účel sa použili dve voľne dostupné databázy, a to Twitter a Weibo, ktoré obsahujú veľké množstvo správ a diskusných

príspevkov z reálneho sveta. Výsledky boli sľubné. V datasete Twitter dosiahli mieru presnosti detekcie 89,2%, v prípade súboru údajov Weibo dokonca až 95,3% (Ruchansky, Seo, Liu, 2017).

Przybyła (2020), sa pokúsil zamerať sa na preskúmanie automatizovaných metód, ktoré dokážu odhaliť online dokumenty s nízkou dôveryhodnosťou (najmä falošné správy), na základe štýlu, v ktorom sú písané. Poukazujú, že textové klasifikátory, napriek zdanlivo dobrému výkonu, keď sa hodnotia zjednodušene, v skutočnosti ukazujú prebytok zdrojov dokumentov v tréningových dátach. Aby dosiahli skutočne ukážkovú predpoveď, zhromaždili 103 219 dokumentov z 223 online zdrojov označených mediálnymi odborníkmi. Ďalej navrhli realistické scenáre hodnotenia a navrhli klasifikátor: neurónovú sieť. Hodnotenie ukazuje, že navrhovaný klasifikátor zachováva vysokú presnosť v prípade dokumentov o predtým nevidených témach (napr. Nové udalosti) a z doteraz nevidených zdrojov (napr. Novovznikajúce spravodajské weby). Analýza modelu naznačuje, že sa skutočne zameriava na znamenitý slovník, o ktorom sa vie, že je typický pre falošné správy.

Kapusta a Obonya (2020), preskúmali rôzne prístupy k detekcii falošných správ, ktoré sú založené na morfologickej analýze. Toto je jedna zo základných zložiek spracovania prirodzeného jazyka. Cieľom bolo zistiť, či je možné na základe morfologickej analýzy vylepšiť metódy prípravy súboru dát. Bol zhromaždený vlastný a jedinečný súbor údajov, ktorý pozostával z článkov od overených vydavateľov a článkov zo spravodajských portálov, ktoré sú známe ako vydavatelia falošných a zavádzajúcich správ. Články boli v slovenskom jazyku, ktorý patrí k flektívnym typom jazykov. V tomto článku preskúmali rôzne prístupy k príprave množiny údajov založené na morfologickej analýze. Pripravené datasety boli vstupnými údajmi pre vytvorenie klasifikátora falošných a skutočných správ. Na klasifikáciu boli vybrané rozhodovacie stromy, ktoré boli vybrané zámerne, kvôli ľahkej interpretácii ich výsledkov. Morfologickou skupinovou analýzou sa našla vhodná technika predspracovania súborov dát. Túto techniku je možné použiť na zlepšenie klasifikácie falošných správ.

Záver práce nás viedol k poznatku, že klasifikácia rozhodovacieho stromu je jednou z najzákladnejších a prakticky najjednoduchších metód strojového učenia. Ak by sme pre vytvorené súbory údajov použili iné metódy, ako napríklad neurónové siete, presnosť klasifikácie by sa zvýšila. Tieto cenné informácie môžeme neskôr použiť pri ďalšej implementácií programu a dosiahnuť tak spresnenie výsledkov práce.

Kapusta a kol. (Kapusta, Munk and Benko, 2020) porovnali základné textové charakteristiky falošných a skutočných typov spravodajských článkov. Analyzovali dva súbory údajov, ktoré obsahovali celkom 28 870 článkov. Výsledky boli overené pomocou tretieho súboru údajov, ktorý tvorilo 402 článkov. Najdôležitejším zistením bol štatisticky významný rozdiel v spravodajskom sentimente, kde sa ukázalo, že falošné spravodajské články majú negatívnejší sentiment. Zaujímavým výsledkom bol tiež rozdiel priemerných počtom slov na vetu. Výsledok bol prekvapením. Predpokladalo sa, že falošné spravodajské články budú jednoduchšie, presnejšie, teda s nižším počtom slov na vetu. Výsledky však ukázali, že falošné správy sa pravdepodobne snažia čitateľa uviesť do omylu komplikovanejším a popisnejším štýlom. Nájdenie štatisticky významných rozdielov v jednotlivých textových charakteristikách je dôležitou informáciou pre použitie budúceho klasifikátora falošných správ.

Uvedomujeme si, že samotná téma detekcie falošných správ je rozsiahla a poskytuje veľa možností riešenia. Je obľúbenou výskumnou témou v oblasti dátovej analýzy, ale väčšina prác a štúdií, ktoré sú dostupné na portáloch a weboch sú spracované na analytický typ jazyka (angličtina). Preto sme sa v procese ďalšieho ujasňovania si tejto témy rozhodli riešiť identifikáciu falošných správ v slovenských textoch. Pri tom bude potrebné vyriešiť niekoľko problémov typických pre flektívne typy jazykov. Rozhodli sme sa zamerať na súbor článkov s aktuálnou témou koronavírus - COVID-19. V aplikačnej časti práce sa budeme venovať implementácií programu, ktorý bude mať čo najvyššiu možnú úspešnosť.

3 METODIKA VÝSKUMU

Prvoradou úlohou práce je navrhnúť optimálnu automatizovanú metódu pre identifikáciu falošných správ, ktorá bude môcť byť následne použitá v praxi. Po použití metód spracovania prirodzeného jazyka, ktorými vyznačíme základné charakteristiky jednotlivých správ sa budeme venovať výberu vhodných metód strojového učenia pre klasifikáciu falošných správ. Špecifickou vlastnosťou práce bude použitie vyhovujúcich metód a nástrojov na texty v slovenskom jazyku. Pri realizácii bude nevyhnutné zohľadniť to, že slovenčina patrí medzi flektívne typy jazykov a nájsť špecifické riešenia. So zámerom dosiahnuť cieľ práce sme si stanovili nasledovné úlohy:

1. Preskúmať aktuálny stav predošlých prác s témou detekcie falošných správ.
2. Vytvoriť dataset článkov s témou COVID-19.
3. Upraviť dataset na potrebný formát k následnému použitiu v jazyku *Python*.
4. Vybrať vhodné metódy na predikciu falošných správ.
5. Implementovať dve vybrané metódy v programovacom jazyku *Python*.
6. Porovnať úspešnosť použitých metód a interpretovať výsledky.

Pri analýze a skúmaní súčasného stavu chceme zistiť akým spôsobom navrhovali riešiť detekciu falošných správ autori predchádzajúcich prác. Tiež nás zaujíma v akom rozsahu sa detekcia falošných správ rieši v zahraničných, ale aj slovenských publikáciách a prácach. Zároveň budeme sledovať či na Slovensku existujú dostupné nástroje zaoberajúce sa touto problematikou.

Príprava dát je jednou z najnáročnejších fáz procesu, pretože kvalita vstupných dát výrazne vplýva na kvalitu dosiahnutých výsledkov. Preto sa v tomto kroku zameriame na zber a prípravu vhodných textov. Výsledkom čoho bude dataset slovenských článkov s indikátorom na pravdivosť článku. Zameriavať sa budeme výhradne na články spojené s aktuálnou témou koronavírus COVID-19. Pri rozhodovaní o tom, či sa jedná o pravdivú alebo falošnú správu nám pomôže hodnotenie webu od občianskeho združenia *Konšpirátori.sk*.

Úprava datasetu bude spočívať v čistení dát od nežiadúcich slov, slovných spojení ako aj textov s českými citáciami. Keďže v našej práci budeme pracovať so slovenským jazykom, české texty a výrazy by mohli vo výsledkoch spôsobiť veľké štatistické odchýlky. Pri úprave datasetu bude ďalším dôležitým krokom aj jeho

konvertovanie do *CSV* formátu z dôvodu jeho následného použitia v programovacom jazyku *Python*.

V ďalšej fáze predspracovania textových dát sa budeme venovať úprave interpunkcie, teda odstráneniu čiarok a ich nahradeniu medzerami. Hlavným dôvodom tohto kroku je fakt, že *CSV* formát považuje čiarku za oddeľovač (delimiter), a tým by sa dataset nesprávne načítal. V datasete budú ponechané všetky ďalšie interpunkčné znamienka ako aj veľké a malé písmená, tak ako to bude prevzaté z originálneho zdroja.

Po dokončení krokov, ktorých úlohou je predspracovanie textov, sa zameriame na výber vhodného klasifikátora. Pre našu prácu sa nám ponúkajú dve možnosti najpoužívanejších algoritmov, ktoré sa využívajú na klasifikáciu a predikciu. Ide o rozhodovacie stromy a neurónové siete. Hlavné rozdiely v týchto dvoch možnostiach sú presnosť rozhodovania a prehľadnosť zobrazenia výsledkov. Pri neurónových sieťach by sme mali dosiahnuť väčšiu presnosť, zatiaľ čo pri rozhodovacích stromoch je výhodou jednoduché zobrazenie pravidiel, na základe ktorých by bola určená pravdivosť článku.

Ďalšou úlohou, ktorou sa budeme v našej práci zaoberať je implementácia programu v jazyku *Python* s použitím nami vybraných klasifikátorov. Tento program bude fungovať na základe modelu, ktorý sa natrénuje na čo najväčšom množstve dát. Tieto trénovacie dáta sa použijú na učenie algoritmu a následne tak dokážu predpovedať výsledky nových prípadov. Výsledkom bude aplikácia, ktorá rozlišuje falošné a pravdivé správy. Táto aplikácia bude používať dva rôzne klasifikátory, a preto zobrazí dva od seba nezávislé výsledky.

V poslednej nami stanovenej úlohe sa zameriame na získané výsledky programu pri použití oboch klasifikátorov. Porovnáme tieto dva výsledky a ak výsledný rozdiel nebude štatisticky významný, ostaneme pri používaní takého klasifikátora, v ktorom sa budú výsledky zobrazovať prehľadnejšie a jednoduchšie.

Realizáciou týchto úloh, chceme vytvoriť aplikáciu, ktorá bude pripravená rozlišovať jednotlivé články a uľahčí tak užívateľovi rozlíšiť a odhaliť pravdivé správy od tých falošných.

4 IMPLEMENTÁCIA NAVRHOVANÉHO PROGRAMU

Implementácia prebiehala v prostredí Google Colab, ktoré umožňuje vytvárať programy v programovacom jazyku Python. Samotná implementácia pozostávala z piatich základných častí:

- Zber dát

V tejto časti bolo potrebné nájsť vhodný materiál a vytvoriť tak dataset reálnych článkov, ktorý je zostavený z názvu, obsahu, zdroja a identifikátora na pravdivosť článku.

- Predspracovanie dát

Fáza predspracovania je kľúčová pre ďalšiu úspešnú implementáciu programu. Zamerali sme sa na prípravu a úpravu vstupných dát do takého formátu, ktorý je potrebný pre ďalšie použitie nami vybraných klasifikátorov. Vstupné dáta, teda články sme rozdelili v pomere 80% trénovacia množina a 20% testovacia množina. Dôležitým aspektom v tejto fáze bolo zachovanie autenticity textov, tak aby sme neovplyvnili výsledky, teda aby náš zásah a ich úprava boli minimálne.

- Rozhodovacie stromy

Na základe predošlých skúsenosti s použitím rozhodovacích stromov sme sa rozhodli implementovať tento klasifikátor detekcie falošných správ. Následne sme zobrazili a opísali výsledky, ktoré sa nám podarilo dosiahnuť.

- Neurónové siete

Ďalším klasifikátorom rozlišovania pravdivých a falošných správ, ktorý sme sa rozhodli použiť na pripravenom datasete je neurónová sieť. Aj v tomto prípade sme po aplikácií zobrazili dosiahnuté výsledky.

- Vyhodnotenie výsledkov

V poslednej fáze sme porovnali percentuálnu úspešnosť oboch klasifikátorov a vyhodnotili výsledky. K tomu patrí aj zdôvodnenie výsledkov na základe teoretických predpokladov.

4.1 ZBER DÁT

Keďže žiaden vhodný existujúci dataset nevyhovoval našim požiadavkám, rozhodli sme sa vyskladať si svoj vlastný. Na extrahovanie článkov dostupných na internete je možné použiť rôzne nástroje. My sme sa rozhodli pre manuálne vyskladanie článkov spojených s aktuálnou témou koronavírus COVID-19. V našom dátovom súbore sú zastúpené články s pravdivými i falošnými správami. Pri rozhodovaní, či sa jedná

o pravdivú alebo falošnú správu nám pomohlo hodnotenie webového zdroja od občianskeho združenia *Konšpirátori.sk*, ktorý sa systematicky a dlhodobo zaoberá pravdivosťou obsahu článkov na internete. Na vytvorenie nášho datasetu sme sa rozhodli využiť formát súboru CSV (Comma Separated Values), pre ktorý je typické, že údaje sú navzájom od seba oddelené čiarkou. Podiel falošných správ k pravdivým sme stanovili v pomere 2:1, teda dataset je zostavený zo sto „fake news“ a päťdesiat „real news“. Dátová množina je teda zložená z pravdivých a falošných správ, pričom pravdivým správam je priradená hodnota 0 a falošné správy sú označené hodnotami medzi 40 a 100. Tieto hodnoty boli prevzaté zo stránky *Konšpirátori.sk* a následne boli upravené tak, aby zodpovedali nami požadovanému formátu. Najdôležitejšou časťou datasetu je samotný obsah článku, ktorý bude hlavným predmetom morfologických analýz. Rovnako tak bol ku každému článku doplnený titulok a zdroj odkiaľ bol článok prevzatý.

4.2 PREDSPRACOVANIE DÁT

Aby sme dataset mohli ďalej spracovávať bolo potrebné ho upraviť tak, aby obsahoval čo najmenej prvkov, ktoré by nám mohli skresliť záverečné vyhodnotenia. V ďalšom kroku bolo preto nutné upraviť interpunkciu textov tak, aby neobsahovali čiarky práve z dôvodu použitého formátu CSV, ktorý používa čiarku ako oddeľovač.

Následne sme importovali potrebné knižnice programovacieho jazyka Python a načítali sme nami vytvorený dataset článkov do používateľského prostredia Google Colab. Tieto knižnice pridávajú podporné funkcie pre prácu s veľkými poliami a maticami. Takisto obsahujú rozsiahlu zbierku matematických funkcií na vysokej úrovni, ktoré sú vhodné pre prácu s rozsiahlymi dátovými poliami. Okrem toho ponúkajú dátové štruktúry a operácie na manipuláciu a analýzu číselných tabuliek.

```
import numpy as np
import pandas as pd
import nltk
import codecs
```

```
[ ] clanky = pd.read_csv("CovidDataset_tags_lemas.csv")
```

Obrázok 1 Import knižníc a načítanie datasetu

Základnou lingvistickou informáciou vo flektívnych jazykoch je morfologická anotácia, ktorá zahŕňa slovnodruhové a tvarové charakteristiky slov v kontexte. Priradenie krátkej charakteristiky slova vo vete sa hovorí značkovanie alebo teda tagovanie. Prideľovanie tagov pre každý článok prebiehal pomocou knižnice „TreeTaggerWrapper“. Po nastavení parametra „TAGLANG“ na jazyk slovenčina uvádzame ukážku ako tento nástroj fungoval na vete „Pomohol v tomto boji“ (viď obrázok 2). Pre všetky textové jednotky (takzvané tokeny), teda reťazce znakov, ktoré sa štandardne nachádzajú medzi dvoma medzerami, sa priradili atribúty tag a lema, ktoré sme si následne uložili do nových stĺpcov tak, aby sme s nimi mohli ďalej pracovať. Lema predstavuje základný, „slovníkový“ tvar tokenu, zahŕňajúci všetky tvary slov z ohybných slovných druhov a prísloviek.

```
import treetaggerwrapper
tagger = treetaggerwrapper.TreeTagger(TAGLANG='sk')

tags = tagger.tag_text('Pomohol v tomto boji.')

print(tags)

['Pomohol\tVLdscm+\tpomôcť', 'v\tEu6\tv', 'tomto\tPFis6\ttento', 'boji\tSSis6\tboj', '.\tSENT\t.']
```

Obrázok 2 Ukážka knižnice TreeTagWrapper

Pre ilustráciu zobrazíme ako vyzerajú vygenerované tagy v našom datasete. Príkladom je nasledujúci obrázok (obrázok 3), v ktorom sme zobrazili tagy prvého článku nášho datasetu. Ide o článok, ktorý začína slovami „Kým vo svete zúri koronavírus...“. Na ňom demonštrujeme ako funguje priraďovanie tagov.

Každej textovej jednotke – slovu, je priradený práve jeden tag. Takže slovu „Kým“ prináleží tag „O“, čo je značka (tag) pre konjunkciu, teda spojku. Podobne slovu „vo“ sa priradí tag v tvare „Ev6“. Prvý znak v tagu „E“ znamená, že toto slovo je prepozícia, inak povedané predložka. Ďalšie znaky v tomto tagu presnejšie popisujú toto slovo a to konkrétne „v“ znamená, že predložka je vokalizovaná a číslo „6“ vyjadruje spojenie s pádom lokál. Dôležité je pripomenúť, že počas našej implementácie budeme využívať iba prvé písmeno tagu, to znamená, že budeme rozlišovať tagy na základe slovného druhu alebo slovnej triedy. Preto pri ďalšom opise tagov budeme vysvetľovať iba prvé znaky, ktoré obsahuje. Ďalšiemu slovu „svete“ sa vygeneroval tag „SSiv6“,

pričom prvé písmeno „S“ znamená substantívum, tiež známe ako podstatné meno. Slovu „zúri“ prináleží tag „V“ teda verbum, čiže ide o sloveso.

```
clanky['tags'][0]
```

```
'O Ev6 SSis6 VKesc+ SSis1 NAmp1 SSmp1 VKepc+ Eu7 PFns7 PAip4 SSip4 VBesc+ VIe+ Eu4 PUis4
SSis4 SSfs4 O SSfs4 SENT O O SSis1 T VKesc+ Eu2 PFis2 SSis2 O Eu2 AAns2x SSns2 VKesc+ VIe
+ Eu3 AAfs3x SSfs3 SSns2 SSmp2 O PUis2 AAis2x SSis2 SENT PD VKesc+ NUns1 AAip2x SSip2 SSi
s2 Eu7 PAmp7 VKepb+ VIe+ SENT SSfs1 O PFmp1 VKepa+ O AAfs1y SSfs1 Eu4 SSis4 VKesc+ Eu2 AA
mp2y SSmp2 VBesc+ AAns1x VIe+ PFns4 PFns4 Eu4 SSfs4:r Dx VKesc+ SENT AAis7x SSis7 VKesc+
O Eu6 SSfs6:r VKesc+ NUns4 AAmp2x SSmp2 SENT T Eu6 SSns6 Eu7 SSfp7 NAis2 SSis2 Eu2 PPhp2
SENT VKepa+ R Dx VIe+ O AAmp1x SSmp1:r VKepc+ Eu6 SSis6 AAms1y O T VBepc+ Dx Gtmp1x Ev6 A
Afp6y AAfp6x SSfp6 O AAmp1x SSmp1:r Z SENT SSis1 VKdsc+ AAfs4x SSfs4 SSns2 Ev3 PAfs3 Eu6
SSfs6:r VKesc+ O R VKdsc+ NUns4 SSfp2 Eu7 AAmp7x O Eu7 AAmp7x SENT O PFns1 T VBesc- AAfs1
x SSfs1 PAfs4 Y R VLdscn+ NUns1 SSmp2 T PD PFns1 VBesc+ SSns1 SSis2 PFns1 VBesc+ VIe+ SSi
p4 Eu6 PAfp6 Y VKepa+ VLepah+ VIe+ SENT SSfs1 Z W SSms1:r VLdscm+ SSmp3 NFfs4 SSfs4 Gkfs4
x R AAfs2x SSfs2 SENT VKesc+ Dx AAfs1x Eu6 SSns6 AAip2x SSip2 SENT O T VKesa+ SSms7 W Eu6
PFfs6 SSfs6 PFns1 Dx VKdsc+ SENT OY VLescf+ W Gtis1x SSis1 AAip4x AAip4x SSip4 O VLescf+
T Dx Dx T Y VLdscf+ VId+ SSfp4 O VId+ SSip4 SENT Eu2 PFns2 VKepa+ R VLdpcm+ SSns2 SSfs2 O
PFns2 O PFfp1 SSfp1 R VLepcf+ SSfp1 Eu6 PFns6 SENT VKesc+ PFns1 O SSfs4 SSmp2 Eu6 W O SSf
s4 O VKesc+ PPhp3 PFns1 T O O VKesc+ Eu4 SSis4 Gtis4x T T AAis4x SSis4 VKdsc- VIe+ T PD P
Php3 VKepc+ SENT SSfs1 O W VLdscf- SSfs4 Eu7 SSis7 VBesc+ Eu6 Gkip6x SSip6 VIe+ AAfs4x SS
fs4 Eu6 SSip6 O SSip6 Eu3 W SENT SSfs1 Z SSfs1 O SSis1 PAmp1 SSmp1 Eu6 AAns6x SSns6 VKepc
+ O Eu3 SSis3 R NUns1 SSmp2 VKdsc+ SSip4 PAip4 PPhp3 SSfs1 VKesc+ SENT O Eu7 PFis7 SSis7
VKesa+ VKesa+ O AAmp1x SSmp1 VBepc+ Eu4 PFis4 SSis4 VIe+ Eu2 SSis2 AAfp2x SSfp2 SENT T O
VKepc+ NAfp1 AAfp1x SSfp1 Dx Eu6 SSfs6 PFfs1 SSfs1 PFmp3 VKesc+ NAfp4 AAfp4x SSfp4 O SSfp
4 SENT Dx PAfp3 SSfp3 O SSmp3 VKdsc+ SSfs1 O SSmp1 VKepc+ SSns4 PAns4 PUns4 VKdsc+ SENT D
x T VKdpc+ Dx AAip4x SSip4 SSis2 O VKdsc+ SSfs1 O SSfs1 SENT SSns2 AAfs2x SSfs2 O AAip2x
SSip2 PD VKepc+ SSfp1 SENT SSfs7 VBesc+ O VKesc+ AAns1x SSns1 Gtns1x VId+ R SSfp4 O VId+
SSfs4 SENT VKesa+ R O AAis1x SSis1 Dx VKdsc- AAis4x SSis4 SSis2 SENT Eu4 PFns4 R VKdsc+ A
Ais1x SSis1 PAms1 VKesc+ VId+ Dx SENT O R AAfs1x SSfs1 Dx VKdsc+ SSip4 AAns2x SSns2 Eu2 P
Fns2 VBepc+ VIe+ SENT PFns1 PFns1 Dx Gtns1x SSis7 VKesc+ SSmp3 VId+ Dy VId+ O SSfs1 Eu6 A
```

Obrázok 3 Ukážka vygenerovaných tagov

Podobne sa ďalej priradia tagy všetkým slovám v texte. Podrobnosti priraďovania tagov nájdeme na stránke Slovenského národného korpusu https://korpus.sk/morpho.html. V nasledujúcom obrázku (obrázok 4) ako príklad uvádzame všetky možné tagy pre slovný druh substantívum – podstatné meno.

Pozícia	Znak	Hodnota	Príklad
1. slovný druh	S	substantívum	slovo, ryba, ústav, muž
2. paradigma	S	substantívna	chlap, žena, srdce
	A	adjektívna	hlavný, vedúci, Mastný, starká, Slaná, vstupné
	F	zmiešaná	kuli, gazdiná
	U	neúplná	kanoe, kupé
3. rod	m	mužský životný	hrdina, hlavný, Mastný
	i	mužský neživotný	strom, rýľ
	f	ženský	ulica, pani, vedúca, Slaná, hradská
	n	stredný	mesto, vysvedčenie, dievča, mláďa
4. číslo	s	jednotné	slovo, ryba, ústav, muž
	p	množné	slová, ryby, ústavy, muži/mužovia
5. pád	1	nominatív	pán, vedúci, matka, Slaná, more, mláďa
	2	genitív	pána, vedúceho, matky, Slanej, mora, mláďaťa
	3	datív	pánovi, vedúcemu, matke, Slanej, moru, mláďaťu
	4	akuzatív	pána, vedúceho, matku, Slanú, more, mláďa
	5	vokatív	pane, mami, Táni, oci
	6	lokál	pánovi, mame, Slanej, mori, mláďati
	7	inštrumentál	pánom, vedúcim, matkou, Slanou, morom, mláďaťom

Obrázok 4 Tagy pre podstatné mená

Po úspešnom priradení tagov všetkým článkom, ich bolo potrebné ďalej upraviť. V ďalšom kroku sme sa zamerali na vyčistenie tagov, teda ponechali sme iba prvý znak a tým sme si uchovali informáciu o tom, aký je to slovný druh. Takže napríklad z tagu „SSiv6“ sme vytvorili „S“. Keďže upravený tag „S“ prináleží aj podstatnému menu, bola potrebná ďalšia úprava, a teda vymazanie tagu „SENT“, ktorý by po úprave bol rovnako „S“. Rozhodli sme sa tento tag úplne vynechať, keďže označuje iba bodku na konci vety, a preto nie je dôležitý pre našu ďalšiu morfologickú analýzu.

```
vysledok = ""
for i in range(144):
  tagy = clanky['tags'][i].split(" ")
  for tag in tagy:
    if(tag=='SENT'):
      print('.')
    else:
      if(len(tag)>=1):
        vysledok += tag[0] + ' '
  clanky['tags'][i] = vysledok
  vysledok = ""
```

Obrázok 5 Implementácia úpravy tagov

Pre potreby ďalšej implementácie sme si nechali vypísať a uložiť všetky tagy, ktoré sa vyskytujú v našom datasete. Zistili sme, že máme devätnásť rôznych tagov, ktoré opisujú slovné druhy a triedy.

```
[6]  vsetky = ''
     for index, row in clanky.iterrows():
        vsetky += row['tags'] + " "
     print(vsetky)

     O E S V S N S V E P P S V V E P S S O S O O S T V E P S O E A S V V E A S S S O P A S P V N A S S E

[7]  prve_tagy_vsetky = vsetky.split(" ")

[8]  prve_tagy = set(prve_tagy_vsetky)

[9]  print(prve_tagy)

     {'', 'Y', 'Q', 'S', '#', 'O', '0', 'Z', 'G', 'A', 'W', '%', 'P', 'E', 'T', 'D', 'J', 'R', 'V', 'N'}
```

Obrázok 6 Vytvorenie množiny tagov

Aby sme mohli skontrolovať ako aktuálne vyzerá náš dataset, rozhodli sme sa v ďalšom kroku zobraziť prvé štyri články. Najpodstatnejšie je overiť si, či naše tagy boli vytvorené správne. Vidíme, že každému článku bol vytvorený nový stĺpec „tags“, kde sme uložili iba prvý znak tagu. Rovnako tak sa vytvoril stĺpec „lemas“, ktorý obsahuje základný tvar slova.

```
clanky.head()
```

	Unnamed: 0	fake_index	publisher	title	content	tags	lemas
0	0	98	protiprudu.org	Aké následky bude mať koronavírus?	Kým vo svete zúri koronavírus mnohí ľudia pre...	O E S V S N S V E P P S V V E P S S O S O O S ...	kým v svet zúriť koronavírus mnohý človek|ľudi...
1	1	97	biblik.sk	VAKCÍNA PROTI COVID MÁ TOĽKO NEŽIADÚCICH ÚČINK...	"Český vakcinológ Marek Petráš tvrdí že klini...	Z A S S S V O A S V D N G S E S T S T E S % V ...	" český vakcinológ marek petráš tvrdiť že klin...
2	2	97	biblik.sk	Dr. RENÉ BALÁK: OTÁZKY O VAKCÍNE COVID-19	"Už mnoho rokov sa vo vakcínach používajú nano...	Z T N S R E S V A S Z S Q S S Z Z Z W S S Z S ...	" už mnoho rok sa v vakcína používať nanotechn...
3	6	97	biblik.sk	LEKÁR ODMIETOL VYŠETRIŤ PACIENTA LEBO NEMAL RÚŠKO	Dnes 20.11.2020 okolo 13:00 v Trenčianskej nem...	D 0 E 0 Z 0 E A S S W S V P V V W O V P S V T ...	dnes @card@ okolo 13 : 0|00 v Trenčianskej nem...

Obrázok 7 Zobrazenie aktuálneho stavu datasetu

V ďalšom kroku sme si definovali funkciu, ktorá vypočíta relatívnu početnosť tagov v jednom článku. Tá vznikne podielom absolútnej početnosti s počtom všetkých tagov v článku. Absolútna početnosť je číslo, ktoré udáva, koľkokrát sa v jednom článku vyskytuje konkrétny tag. V našom programe vytvoríme devätnásť stĺpcov pre každý tag, ktorý sa môže v našom datasete vyskytnúť a do neho uložíme hodnoty relatívnej početnosti. Takto získame vstupné údaje do rozhodovacích stromov a neurónových sietí, s ktorými budeme ďalej pracovať.

```
def rataj_tagy(tagy,co):
    vysledok = 0
    if type(tagy) == str:
        #medzere musia tu byť dve
        pocet_vsetkych = tagy.count(' ')
        if pocet_vsetkych == 0:
            pocet_vsetkych = 1
        vysledok = tagy.count(co)/pocet_vsetkych
    print('.')
    return vysledok
```

```
[ ] for jeden_tag in prve_tagy:
        if jeden_tag!= "":
            hladaj = jeden_tag+' '
            clanky[jeden_tag] = clanky.apply(lambda row: rataj_tagy(row['tags'],hladaj), axis=1)
```

Obrázok 8 Implementácia relatívnej početnosti tagov

Kvôli prehľadnosti datasetu sme sa rozhodli ponechať iba niektoré jeho stĺpce. Konkrétne ide o všetky tagy a ich relatívnu početnosť a samozrejme indikátor o tom, či je článok označený ako falošný alebo pravdivý. Do tejto fázy predspracovania dát sme mali hodnoty v stĺpci „fake_index“ uložené ako hodnoty pridelené občianskym združením Konšpirátori.sk. Tieto hodnoty sme v ďalšom kroku nahradili jednoduchším zápisom „1“ pre falošnú správu a „0“ pre pravdivú. Nasledujúci obrázok (obrázok 9) zobrazuje aktuálny stav datasetu.

```
[ ] clanky.to_csv('vystup_v1.csv')
```

```
clanky_vstup = pd.read_csv("vystup_v2.csv", sep = ";")
```

+ Code + Text

```
[ ] clanky_vstup.tail()
```

	row	fake_index	R	J	A	O	E	S	P	V
139	145	0	0.027855	0.0	0.144847	0.052925	0.114206	0.289694	0.050139	0.167131
140	146	0	0.004878	0.0	0.092683	0.039024	0.131707	0.385366	0.043902	0.117073
141	147	0	0.019417	0.0	0.095700	0.063800	0.130374	0.273232	0.072122	0.185853
142	148	0	0.013699	0.0	0.091324	0.041096	0.086758	0.337900	0.063927	0.173516
143	149	0	0.017544	0.0	0.098246	0.049123	0.140351	0.329825	0.042105	0.133333

Obrázok 9 Zobrazenie relatívnej početnosti tagov

Pre detailnejší opis datasetu sme použili príkaz „describe()“ dostupný z knižnice „Pandas“, ktorý nám vypočíta a zobrazí viacero hodnôt. Pre každý stĺpec zobrazí:

- count: počet prvkov
- mean: priemer hodnôt
- std: štandardná odchýlka údajov
- min a max: minimálne a maximálne hodnoty
- 50%: stredná hodnota (medián)

```
clanky_vstup.describe()
```

	row	fake_index	R	J	A	O	E	
count	144.000000	144.000000	144.000000	144.000000	144.000000	144.000000	144.000000	144.000(
mean	76.479167	0.652778	0.024672	0.000154	0.099689	0.072848	0.103644	0.299{
std	42.734007	0.477749	0.009919	0.000566	0.025962	0.020829	0.022434	0.045:
min	0.000000	0.000000	0.000000	0.000000	0.039106	0.015504	0.050279	0.177{
25%	39.750000	0.000000	0.017899	0.000000	0.082153	0.061408	0.087998	0.266(
50%	76.500000	1.000000	0.023930	0.000000	0.095917	0.073611	0.101912	0.300{
75%	113.250000	1.000000	0.030011	0.000000	0.114143	0.086973	0.117381	0.330(
max	149.000000	1.000000	0.054974	0.003390	0.196078	0.131579	0.170732	0.406(

Obrázok 10 Súhrn štatistík datasetu

Pre ďalšiu prácu budeme potrebovať premeniť typ uloženia našich všetkých tagov zo štruktúry množiny do štruktúry zoznamu. Pri predchádzajúcom uložení tagov sa nám v množine vyskytol prázdny znak, ktorý samozrejme z tohto zoznamu vynecháme, keďže nereprezentuje žiaden slovný druh ani triedu.

```
tagy = []
for tag in prve_tagy:
    if tag!="":
        tagy.append(tag)
```

```
print(tagy)
```

```
['Q', '#', 'N', 'S', 'O', 'T', 'E', 'R', 'W', '%', 'Y', 'P', 'A', 'V', '0', 'G', 'D', 'J', 'Z']
```

Obrázok 11 Vytvorenie zoznamu tagov

K záverečnej časti predspracovania údajov patrí importovanie knižnice „train_test_split“ , ktoré zabezpečí rýchle a efektívne rozdelenie vstupného súboru na trénovaciu a testovaciu množinu. Pomocou tejto knižnice sme rozdelili dataset na stĺpce,

ktoré slúžia ako funkcia, tu je zahrnutých všetkých devätnásť stĺpcov s tagmi a cieľovú premennú, v našom prípade stĺpec „fake_index“, ktorý slúži na výpočet predpovede.

Ako sme už skôr spomínali, je potrebné rozdeliť vstupné údaje na trénovaciu a testovaciu množinu v pomere 80% trénovacích a 20% testovacích dát. V tomto príkaze sme použili parameter „random_state“, ktorý zabezpečuje rovnaké rozdelenie pri viacerých zbehnutiach programu.

```
#split dataset in features and target variable
X = clanky_vstup[tagy] # Features
y = clanky_vstup.fake_index # Target variable
```

```
# 80% training and 20% test
from sklearn.model_selection import train_test_split # Import train_test_split function
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Obrázok 12 Rozdelenie dát na trénovacie a testovacie

V tomto momente je fáza predspracovania dát ukončená a môžeme pokračovať implementovaním rozhodovacích stromov a neurónových sietí.

4.3 ROZHODOVACIE STROMY

K známym klasifikačným modelom patria algoritmy pre vytváranie rozhodovacích stromov. Rozhodovací strom môžeme definovať ako strom (stromový graf), kde každý nelistový uzol stromu predstavuje test hodnoty atribútu a vetvy vedúcej z tohto uzla, možné výsledky testu. Listové uzly stromu sú ohodnotené identifikátory tried (výsledky klasifikácie). Klasifikácia pomocou stromu prebieha cestou záznamu od koreňa stromu k jeho listu. V každom kroku je záznam testovaný podľa podmienok v aktuálnom uzle rozhodovacieho stromu a ďalej pokračuje po vetve podľa konkrétneho výsledku testu. Keď záznam dôjde až do listového uzla, je klasifikovaný triedou, ktorá je identifikovaná hodnotou príslušného listu rozhodovacieho stromu.

Intuitívne vizuálne zobrazenie stromom napomáha jasnejšiemu pochopenie výsledkov a vzťahov i laickým používateľom a v praxi tak uľahčuje ich rozhodovanie. Stromové grafy tak dovoľujú vizuálne preskúmať výsledky a posúdiť vhodnosť modelu. Rozhodovací strom možno pomerne ľahko previesť na rozhodovacie pravidlá. Každej ceste od koreňa k listu stromu zodpovedá jedno pravidlo. Nelistové uzly sú predpoklady, listový uzol potom záverom pravidla.

Atribút vhodný pre vetvenie stromu vyberáme na základe jeho charakteristík prevzatých z teórie informácie a pravdepodobnosti: entropia, informačného zisku, pomerného informačného zisku, Chi-square testu, Giniho indexu a ďalších (Křupka, Kašparová, Jirava, 2010).

Pre implementáciu rozhodovacích stromov v používateľskom rozhraní Google Colab bolo potrebné importovanie niektorých Python knižníc, a to „DecisionTreeClassifier“ pre použitie klasifikátora rozhodovacie stromy a „metrics“ pre následné vypočítanie úspešnosti modelu.

Po úspešnom naimportovaní knižníc, sme potom vytvorili objekt klasifikátora rozhodovacích stromov a príkazom „fit()“ sme ho natrénovali pomocou dát, na to určených. Nakoniec sme mohli predikovať výsledky na testovacích dátach a následne ich porovnať so skutočnými dátami v datasete. Týmto sme získali úspešnosť predikcie modelu. V našom prípade je to 75,86 % úspešnosť, čo je v prvotnom nastavení uspokojivý výsledok. Zo štatistického hľadiska pri náhodnom výbere výsledkov, by úspešnosť mala byť 50 % a teda môžeme tvrdiť, že náš model rozhodovacích stromov je výrazne lepší ako náhodný výber výsledkov.

```
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
```

```
# Create Decision Tree classifer object
#clf = DecisionTreeClassifier(criterion="gini", max_depth=20)
clf = DecisionTreeClassifier(criterion="entropy")

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)
```

```
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.7586206896551724
```

Obrázok 13 Implementácia rozhodovacích stromov

Následne sme si otestovali rozhodovací strom s rôznou hĺbkou a sledovali sme ako sa mení úspešnosť detekcie falošných a pravdivých správ. Rozhodli sme sa použiť maximálnu hĺbku stromu od 2 do 19. Pri týchto nastaveniach sme mohli pozorovať, ako

sa už pri maximálnej hĺbke 7 úspešnosť detekcie stabilizuje okolo hodnoty 80 %. Pri vyššej hodnote hĺbky sa úspešnosť už výrazne nemení. Nasvedčuje tomu výpis počtu uzlov, ktorý je od hodnoty hĺbky 7 stabilizovaný okolo hodnoty 30.

```
for i in range(2,20):
    #clf = DecisionTreeClassifier(criterion="gini",max_depth=i)
    clf = DecisionTreeClassifier(max_depth=i)
    clf = clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)
    pocet_uzlov = clf.tree_.node_count
    pocet_listov = len([x for x in clf.tree_.feature if x != _tree.TREE_UNDEFINED]) + 1
    print("Max Deep: ",i,", Node count:",pocet_uzlov,", Leaf count:",pocet_listov,
          ", Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Max Deep:  2 , Node count: 7 , Leaf count: 4 , Accuracy: 0.896551724137931
Max Deep:  3 , Node count: 11 , Leaf count: 6 , Accuracy: 0.896551724137931
Max Deep:  4 , Node count: 15 , Leaf count: 8 , Accuracy: 0.8275862068965517
Max Deep:  5 , Node count: 21 , Leaf count: 11 , Accuracy: 0.7586206896551724
Max Deep:  6 , Node count: 27 , Leaf count: 14 , Accuracy: 0.7931034482758621
Max Deep:  7 , Node count: 31 , Leaf count: 16 , Accuracy: 0.7586206896551724
Max Deep:  8 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
Max Deep:  9 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
Max Deep:  10 , Node count: 31 , Leaf count: 16 , Accuracy: 0.8275862068965517
Max Deep:  11 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
Max Deep:  12 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7586206896551724
Max Deep:  13 , Node count: 29 , Leaf count: 15 , Accuracy: 0.8275862068965517
Max Deep:  14 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
Max Deep:  15 , Node count: 31 , Leaf count: 16 , Accuracy: 0.7931034482758621
Max Deep:  16 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
Max Deep:  17 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
Max Deep:  18 , Node count: 31 , Leaf count: 16 , Accuracy: 0.7931034482758621
Max Deep:  19 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
```

Obrázok 14 Testovanie rôznej maximálnej dĺžky rozhodovacích stromov

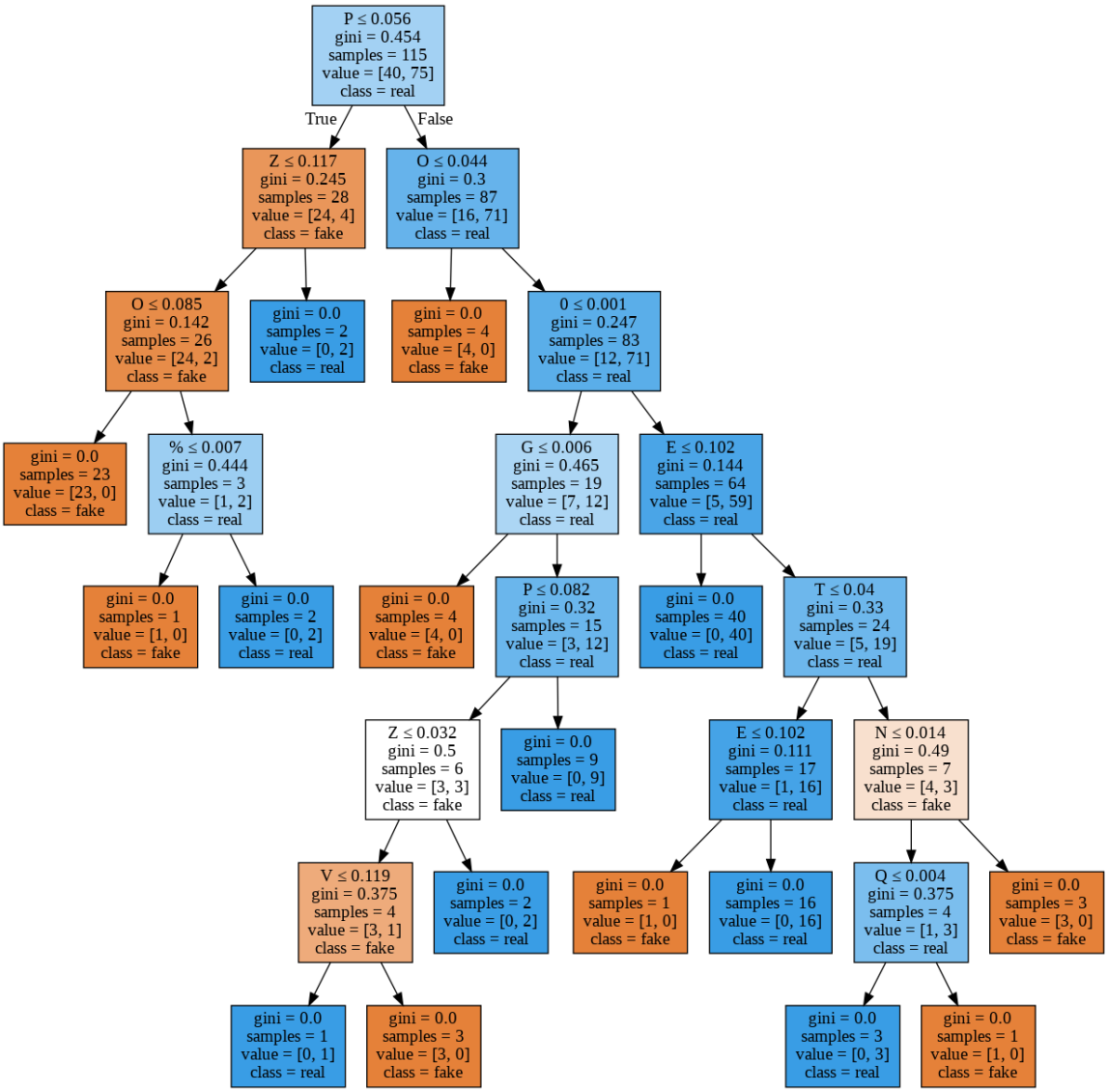
Na zobrazenie rozhodovacieho stromu sme použili knižnicu „sklearn.tree“. Konkrétne funkciu „export_graphviz“, ktorá generuje rozhodovací strom s formátom Graphviz. Čítanie tohto súboru sme zrealizovali pomocou knižnice „PyDotPlus“, pomocou ktorej sme vygenerovali grafické znázornenie stromu. Táto knižnica je vylepšená verzia starého projektu „PyDot“, ktorý poskytuje rozhranie Python pre jazyk Dot Graphviz.

```
dot_data = StringIO()
export_graphviz(clf, out_file=dot_data,
                filled=True, rounded=False,
                special_characters=True,feature_names = tagy,class_names=['fake','real'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('skuska.png')
Image(graph.create_png())
```

Obrázok 15 Implementácia grafu rozhodovacieho stromu

Na obrázku (obrázok 16) ponúkame pohľad na nami vygenerovaný rozhodovací strom. Každá stromová štruktúra sa skladá z prepojených uzlov. Ako je možné vidieť na zobrazení stromu, prvý štvorec zhora je uzlom, ktorý je prepojený s dvoma ďalšími uzlami nižšie. Tieto uzly nazývame potomok uzla. Uzol, ktorý nemá potomkov sa nazýva list, a aj túto hodnotu sme mohli pozorovať v vyššie uvedenom obrázku (obrázok 14) .

V nasledujúcom obrázku (obrázok 16) sa zobrazuje posledný rozhodovací strom, ktorý má nastavenú maximálnou hĺbku 19. Môžeme si všimnúť, že reálna hĺbka je 7 a ďalej sa už hĺbka nezvyšuje. Tak ako sme už v predchádzajúcej časti uvádzali, počet uzlov pri maximálnej hĺbke 7 je 29 a ďalej sa už výrazne nemení.



Obrázok 16 Zobrazenie grafu rozhodovacieho stromu

V ďalšom kroku sme premenili stromovú štruktúru do pravidiel vo forme podmienok a na to sme vytvorili vlastnú funkciu „tree_to_code“. Platí, že počet pravidiel sa rovná počtu listov v rozhodovacom strome. V našom prípade je to 16 pravidiel. Pravidlo rozhodovacieho stromu vzniká tak, že pre jeden list spíšeme všetky podmienky od koreňa po list. Koreň je uzol s hĺbkou nula.

Napríklad pravidlo číslo jedna vzniklo posúvaním sa od koreňa do ľavého potomka (viď obrázok 16) tri krát. Posúvanie sa do ľavého potomka znamená, že podmienky v každom uzle sú označené ako „True“ a tak vznikne pravidlo číslo jedna.

```
tree_to_code(clf, tagy, y)
```

```
1 ) P <= 0.05590413138270378  & Z <= 0.11713586747646332  & O <= 0.08530711755156517  [[23.  0.]]
2 ) P <= 0.05590413138270378  & Z <= 0.11713586747646332  & O > 0.08530711755156517 & % <= 0.006870228797197342  [[1. 0.]]
3 ) P <= 0.05590413138270378  & Z <= 0.11713586747646332  & O > 0.08530711755156517 & % > 0.006870228797197342 [[0. 2.]]
4 ) P <= 0.05590413138270378  & Z > 0.11713586747646332 [[0. 2.]]
5 ) P > 0.05590413138270378 & O <= 0.044200627133250237  [[4. 0.]]
6 ) P > 0.05590413138270378 & O > 0.044200627133250237 & 0 <= 0.0014839951181784272  & G <= 0.006029285024851561  [[4. 0.]
7 ) P > 0.05590413138270378 & O > 0.044200627133250237 & 0 <= 0.0014839951181784272  & G > 0.006029285024851561 & P <= 0.0
8 ) P > 0.05590413138270378 & O > 0.044200627133250237 & 0 <= 0.0014839951181784272  & G > 0.006029285024851561 & P <= 0.0
9 ) P > 0.05590413138270378 & O > 0.044200627133250237 & 0 <= 0.0014839951181784272  & G > 0.006029285024851561 & P <= 0.0
10 ) P > 0.05590413138270378 & O > 0.044200627133250237 & 0 <= 0.0014839951181784272  & G > 0.006029285024851561 & P > 0.0
11 ) P > 0.05590413138270378 & O > 0.044200627133250237 & 0 > 0.0014839951181784272 & E <= 0.10154327377676964  [[ 0. 40.]
12 ) P > 0.05590413138270378 & O > 0.044200627133250237 & 0 > 0.0014839951181784272 & E > 0.10154327377676964 & T <= 0.040
13 ) P > 0.05590413138270378 & O > 0.044200627133250237 & 0 > 0.0014839951181784272 & E > 0.10154327377676964 & T <= 0.040
14 ) P > 0.05590413138270378 & O > 0.044200627133250237 & 0 > 0.0014839951181784272 & E > 0.10154327377676964 & T > 0.0404
15 ) P > 0.05590413138270378 & O > 0.044200627133250237 & 0 > 0.0014839951181784272 & E > 0.10154327377676964 & T > 0.0404
16 ) P > 0.05590413138270378 & O > 0.044200627133250237 & 0 > 0.0014839951181784272 & E > 0.10154327377676964 & T > 0.0404
```

Obrázok 17 Podmienky rozhodovacieho stromu

V nasledujúcej časti porovnávame úspešnosť detekcie za použitia dvoch rozdielnych kritérií „Gini“ a „Entropy“. Gini sa počíta pomocou nasledujúceho vzorca:

$$GiniIndex = 1 - \sum_j p_j^2$$

Kde p_j je pravdepodobnosť triedy j.

Gini meria frekvenciu, s akou bude akýkoľvek náhodne označený prvok súboru údajov ohodnotený nesprávne. Minimálna hodnota Giniho indexu je 0. To nastane vtedy, keď je uzol čistý, čo znamená, že všetky obsiahnuté prvky v uzle sú jednej jedinečnej triedy. Preto sa tento uzol znova nerozdelí. Optimálne rozdelenie si teda vyberá prvky s menším indexom Gini. Maximálnu hodnotu získa vtedy, keď je pravdepodobnosť týchto dvoch tried rovnaká:

$$Gini_{min} = 1 - (1^2) = 0$$
$$Gini_{max} = 1 - (0{,}5^2 + 0{,}5^2) = 0{,}5$$

Entropia sa počíta pomocou nasledujúceho vzorca:

$$Entropy = -\sum_j p_j \ . log_2 \, . p_j$$

Kde, rovnako ako predtým, p_j je pravdepodobnosť triedy j.

Entropia je miera informácie, ktorá naznačuje poruchu vlastností s cieľom. Podobne ako v prípade Giniho indexu, optimálne rozdelenie vyberá prvok s menšou Entropiou. Maximálnu hodnotu získa, keď je pravdepodobnosť týchto dvoch tried rovnaká a uzol je čistý, keď má Entropia minimálnu hodnotu, ktorá je 0:

$$Entropy_{min} = -\,1 \, . log_2 \, (\, 1 \,) = 0$$
$$Entropy_{max} = -\,0{,}5 \, . log_2 \, (\, 0{,}5 \,) - 0{,}5 \, . log_2 \, (\, 0{,}5 \,) = 1$$

Z teoretického hľadiska by kritérium Gini malo byť značne rýchlejšie, pretože je výpočtovo menej náročné a na druhej strane by získané výsledky pomocou kritéria Entropia mali byť o niečo lepšie. Pretože sú výsledky podobné, zdá sa, že v našom prípade sa neoplatí investovať čas do tréningu pri použití kritéria Entropia.

```
for i in range(2,20):
    clf = DecisionTreeClassifier(criterion="gini",max_depth=i)
    clf = clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)
    pocet_uzlov = clf.tree_.node_count
    pocet_listov = len([x for x in clf.tree_.feature if x != _tree.TREE_UNDEFINED]) + 1
    print("Max Deep: ",i,", Node count:",pocet_uzlov,", Leaf count:",pocet_listov,
          ", Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Max Deep:  2 , Node count: 7 , Leaf count: 4 , Accuracy: 0.896551724137931
Max Deep:  3 , Node count: 11 , Leaf count: 6 , Accuracy: 0.896551724137931
Max Deep:  4 , Node count: 15 , Leaf count: 8 , Accuracy: 0.8275862068965517
Max Deep:  5 , Node count: 19 , Leaf count: 10 , Accuracy: 0.7586206896551724
Max Deep:  6 , Node count: 25 , Leaf count: 13 , Accuracy: 0.7586206896551724
Max Deep:  7 , Node count: 31 , Leaf count: 16 , Accuracy: 0.7586206896551724
Max Deep:  8 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
Max Deep:  9 , Node count: 29 , Leaf count: 15 , Accuracy: 0.8275862068965517
Max Deep:  10 , Node count: 31 , Leaf count: 16 , Accuracy: 0.8275862068965517
Max Deep:  11 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
Max Deep:  12 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7586206896551724
Max Deep:  13 , Node count: 31 , Leaf count: 16 , Accuracy: 0.8275862068965517
Max Deep:  14 , Node count: 31 , Leaf count: 16 , Accuracy: 0.7931034482758621
Max Deep:  15 , Node count: 29 , Leaf count: 15 , Accuracy: 0.8275862068965517
Max Deep:  16 , Node count: 29 , Leaf count: 15 , Accuracy: 0.8275862068965517
Max Deep:  17 , Node count: 31 , Leaf count: 16 , Accuracy: 0.8275862068965517
Max Deep:  18 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7586206896551724
Max Deep:  19 , Node count: 31 , Leaf count: 16 , Accuracy: 0.8275862068965517
```

Obrázok 18 Rozhodovacie stromy s použitím kritéria Gini

```
for i in range(2,20):
    clf = DecisionTreeClassifier(criterion="entropy",max_depth=i)
    clf = clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)
    pocet_uzlov = clf.tree_.node_count
    pocet_listov = len([x for x in clf.tree_.feature if x != _tree.TREE_UNDEFINED]) + 1
    print("Max Deep: ",i,", Node count:",pocet_uzlov,", Leaf count:",pocet_listov,
          ", Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Max Deep:  2 , Node count: 7 , Leaf count: 4 , Accuracy: 0.896551724137931
Max Deep:  3 , Node count: 13 , Leaf count: 7 , Accuracy: 0.8275862068965517
Max Deep:  4 , Node count: 19 , Leaf count: 10 , Accuracy: 0.896551724137931
Max Deep:  5 , Node count: 21 , Leaf count: 11 , Accuracy: 0.8275862068965517
Max Deep:  6 , Node count: 25 , Leaf count: 13 , Accuracy: 0.7586206896551724
Max Deep:  7 , Node count: 27 , Leaf count: 14 , Accuracy: 0.6896551724137931
Max Deep:  8 , Node count: 31 , Leaf count: 16 , Accuracy: 0.7931034482758621
Max Deep:  9 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
Max Deep:  10 , Node count: 31 , Leaf count: 16 , Accuracy: 0.7241379310344828
Max Deep:  11 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
Max Deep:  12 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7586206896551724
Max Deep:  13 , Node count: 31 , Leaf count: 16 , Accuracy: 0.8275862068965517
Max Deep:  14 , Node count: 31 , Leaf count: 16 , Accuracy: 0.7586206896551724
Max Deep:  15 , Node count: 31 , Leaf count: 16 , Accuracy: 0.7931034482758621
Max Deep:  16 , Node count: 29 , Leaf count: 15 , Accuracy: 0.8275862068965517
Max Deep:  17 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7586206896551724
Max Deep:  18 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
Max Deep:  19 , Node count: 29 , Leaf count: 15 , Accuracy: 0.7931034482758621
```

Obrázok 19 Rozhodovacie stromy s použitím kritéria Entropy

V ďalšej časti sa môžeme pozrieť na dôležitosť alebo inak povedané, váhu jednotlivých slovných druhov a slovných tried. Najdôležitejším faktorom pri rozhodovaní, či je správa falošná alebo pravdivá, boli zámená (P = pronominá). Ďalšími rozhodovacími faktormi boli napríklad príčastia (G = particípium), spojky (O = konjunkcia) a predložky (E = prepozícia). V nami vygenerovanom modeli bolo spolu 8 slovných druhov a tried, ktoré mali nulovú váhu a 11 takých, ktoré boli pre detekciu správ dôležité. Medzi tie, ktoré neboli pre rozhodovanie podstatné patria napríklad podstatné mená (substantívum) alebo príslovky (adverbium).

```
[ ]  importances = pd.DataFrame({'feature':X_train.columns,'importance':np.round(clf.feature_importances_,8)})
     importances = importances.sort_values('importance',ascending=False)
```

```
[ ]  importances
```

	feature	importance
9	P	0.402535
0	G	0.166883
10	O	0.107042
5	E	0.061035
3	T	0.049943
2	0	0.047328
16	Z	0.045214
13	N	0.036964
7	A	0.028750
1	V	0.028750
18	J	0.025556
6	R	0.000000
8	W	0.000000
4	S	0.000000
11	Q	0.000000
12	#	0.000000
14	Y	0.000000
15	%	0.000000
17	D	0.000000

Obrázok 20 Dôležitosť tagov

Tieto váhy nám ukazujú, že nezáleží na počte podstatných mien alebo prísloviek použitých v článku, ale falošné správy sa skôr vyznačujú vysokým počtom zámen.

4.4 NEURÓNOVÉ SIETE

Princíp softvérovej neurónovej siete sa inšpiruje biologickou neurónovou sieťou, kde základným stavebným prvkom je nervová bunka - neurón. Jednotlivé neuróny sú vzájomne prepojené a ohodnotené váhami. Takéto prepojenie a schopnosť tieto váhy učiť sa na základe trénovacích vzorov v dátach, dáva neurónovej sieti nové široké možnosti v oblasti analýzy dát. Hlavnou prednosťou neurónovej siete je schopnosť zapamätať si kombináciu, ktorá viedla k požadovanému výstupu, a pri nových vstupoch sa potom obracať na svoju pamäť a na základe skúseností odhadovať nový výsledok. V tomto prípade hovoríme o zovšeobecňovaní (generalizácií), ktorá je ďalšou veľkou prednosťou algoritmu neurónových sietí. Zjednodušene povedané, ide o primeranú zručnosť správne zareagovať aj na vstupy, ktoré neboli súčasťou trénovacích dát, a vyvodiť z nich všeobecné závery o dátach.

Ako už bolo naznačené, neurónové siete sa skladajú z neurónov prepojených väzbami. Model neurónu sa skladá z troch častí: vstupné, výstupné a funkčné. Na základe váh môžu byť jednotlivé vstupy potlačené, alebo naopak zvýhodnené. Jeden neurón sám o sebe nie je schopný vykonávať o nič viac zložitejšiu funkciu ako klasická regresná analýza. Sila neurónových sietí sa však prejaví až pri prepojení neurónov medzi sebou do väčších štruktúr. Neuróny sú usporiadané do vrstiev. Nezabudnime, že algoritmus sa učí sám a všetky váhy si sám volí. Ak si predstavíme, ako takouto sieťou prúdia dáta, je zjavné, že práve vďaka zložitosti spojenia dokáže neurónová sieť nájsť aj zložitejšie a nelineárne vzťahy. Na druhej strane je ale pravda, že zo získanej neurónovej siete nikdy nebudeme schopní získať interpretáciu, prečo to dopadlo, ako to dopadlo. Nie sme schopní jednoducho interpretovať výsledky, či získať jednoduchý predpis medzi závislými a nezávislými premennými (Uldrich, Jurczyk, 2014).

Pre implementáciu neurónových sietí ako klasifikátora našej detekcie pravdivých a falošných článkov sme použili Python knižnicu „keras“. Ponúka konzistentné a jednoduché rozhrania API, minimalizuje počet akcií používateľov potrebných pre bežné prípady použitia a poskytuje jasné a použiteľné chybové správy. Má tiež rozsiahlu dokumentáciu a príručky pre vývojárov.

V nasledujúcom obrázku (obrázok 21) môžeme vidieť zadefinovanie neurónovej siete s troma vrstvami. Hodnota „input_dim“ je v prvej vrstve 19, pretože máme 19 slovných druhov a tried v našom datasete. V ďalších vrstvách už nie je potrebné zadávať túto hodnotu, keďže neurónová sieť si ich vygeneruje. Rovnako zadávame hodnotu

„units“, ktorá je v našom prípade 11 v prvých dvoch vrstvách s aktivačnou funkciou „relu“ a v poslednej je hodnota 1 s aktivačnou funkciou „sigmoid“. Ďalšia použitá premenná „batch_size“ nám rozdelí vstupné hodnoty do skupín (šarží). V našom prípade sme mali 115 článkov, z ktorých sa 70% (80 článkov) použilo ako trénovacie hodnoty počas učenia neurónovej siete. Keďže hodnotu „batch_size“ sme nastavili na 5, našich 80 článkov sa rozdelilo do 16 skupín po 5 článkov.

V našich pokusoch sme vyskúšali rôzne kombinácie týchto vstupných hodnôt, no najlepšie výsledky sme dosiahli práve použitím hore uvedených hodnôt.

```
[247] model = Sequential()
      model.add(Dense(11,activation='relu',input_dim = 19))
      model.add(Dense(11,activation='relu'))
      model.add(Dense(1,activation='sigmoid'))

      model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
history = model.fit(X_train,y_train,epochs= 500, validation_split=0.30, batch_size=5)
```

Obrázok 21 Implementácia neurónových sietí

Ďalej sa môžeme pozrieť ako prebiehalo trénovanie neurónovej siete. Použitím hodnoty „epochs = 500“ sme dosiahli päťsto iterácií učenia modelu a v ďalšom obrázku (obrázok 22) sa môžeme pozrieť ako vyzerá úspešnosť neurónovej siete v prvých desiatich opakovaniach. Úspešnosť detekcie môžeme pozorovať na hodnote „accuracy“, ktorá sa na začiatku pohybuje okolo hodnoty 65%. Hodnota „val_accuracy“ je úspešnosť predikcie na dátach, ktoré boli určené na validáciu, teda na overenie.

```
Epoch 1/500
16/16 [==============================] - 1s 16ms/step - loss: 0.6900 - accuracy: 0.6626 - val_loss: 0.6870 - val_accuracy: 0.6286
Epoch 2/500
16/16 [==============================] - 0s 5ms/step - loss: 0.6867 - accuracy: 0.6232 - val_loss: 0.6839 - val_accuracy: 0.6286
Epoch 3/500
16/16 [==============================] - 0s 5ms/step - loss: 0.6795 - accuracy: 0.6707 - val_loss: 0.6807 - val_accuracy: 0.6286
Epoch 4/500
16/16 [==============================] - 0s 4ms/step - loss: 0.6694 - accuracy: 0.7352 - val_loss: 0.6772 - val_accuracy: 0.6286
Epoch 5/500
16/16 [==============================] - 0s 4ms/step - loss: 0.6701 - accuracy: 0.6760 - val_loss: 0.6742 - val_accuracy: 0.6286
Epoch 6/500
16/16 [==============================] - 0s 5ms/step - loss: 0.6728 - accuracy: 0.6311 - val_loss: 0.6713 - val_accuracy: 0.6286
Epoch 7/500
16/16 [==============================] - 0s 5ms/step - loss: 0.6794 - accuracy: 0.5814 - val_loss: 0.6684 - val_accuracy: 0.6286
Epoch 8/500
16/16 [==============================] - 0s 5ms/step - loss: 0.6627 - accuracy: 0.6503 - val_loss: 0.6650 - val_accuracy: 0.6286
Epoch 9/500
16/16 [==============================] - 0s 5ms/step - loss: 0.6453 - accuracy: 0.7042 - val_loss: 0.6622 - val_accuracy: 0.6286
Epoch 10/500
16/16 [==============================] - 0s 4ms/step - loss: 0.6382 - accuracy: 0.7035 - val_loss: 0.6593 - val_accuracy: 0.6286
```

Obrázok 22 *Prvých desať iterácií trénovania neurónových sietí*

V posledných desiatich opakovaniach je možné pozorovať značný pokrok a to na úspešnosti detekcie, ktorá sa pohybuje okolo hodnoty 90%. Taktiež si môžeme všimnúť, že hodnoty „loss“, teda chybovosť modelu sa značne znížila.

```
Epoch 490/500
16/16 [==============================] - 0s 5ms/step - loss: 0.1438 - accuracy: 0.9501 - val_loss: 0.3909 - val_accuracy: 0.8286
Epoch 491/500
16/16 [==============================] - 0s 5ms/step - loss: 0.1518 - accuracy: 0.9521 - val_loss: 0.3912 - val_accuracy: 0.8286
Epoch 492/500
16/16 [==============================] - 0s 5ms/step - loss: 0.1161 - accuracy: 0.9705 - val_loss: 0.3865 - val_accuracy: 0.8286
Epoch 493/500
16/16 [==============================] - 0s 5ms/step - loss: 0.1140 - accuracy: 0.9514 - val_loss: 0.3902 - val_accuracy: 0.8286
Epoch 494/500
16/16 [==============================] - 0s 5ms/step - loss: 0.1690 - accuracy: 0.9438 - val_loss: 0.3887 - val_accuracy: 0.8571
Epoch 495/500
16/16 [==============================] - 0s 5ms/step - loss: 0.1958 - accuracy: 0.9107 - val_loss: 0.3908 - val_accuracy: 0.8286
Epoch 496/500
16/16 [==============================] - 0s 5ms/step - loss: 0.1200 - accuracy: 0.9577 - val_loss: 0.3928 - val_accuracy: 0.8286
Epoch 497/500
16/16 [==============================] - 0s 5ms/step - loss: 0.1691 - accuracy: 0.9275 - val_loss: 0.4013 - val_accuracy: 0.8571
Epoch 498/500
16/16 [==============================] - 0s 6ms/step - loss: 0.1318 - accuracy: 0.9589 - val_loss: 0.3967 - val_accuracy: 0.8857
Epoch 499/500
16/16 [==============================] - 0s 5ms/step - loss: 0.1213 - accuracy: 0.9839 - val_loss: 0.3870 - val_accuracy: 0.8571
Epoch 500/500
16/16 [==============================] - 0s 5ms/step - loss: 0.0853 - accuracy: 0.9784 - val_loss: 0.3854 - val_accuracy: 0.8571
```

Obrázok 23 Posledných desať iterácií trénovania neurónových sietí

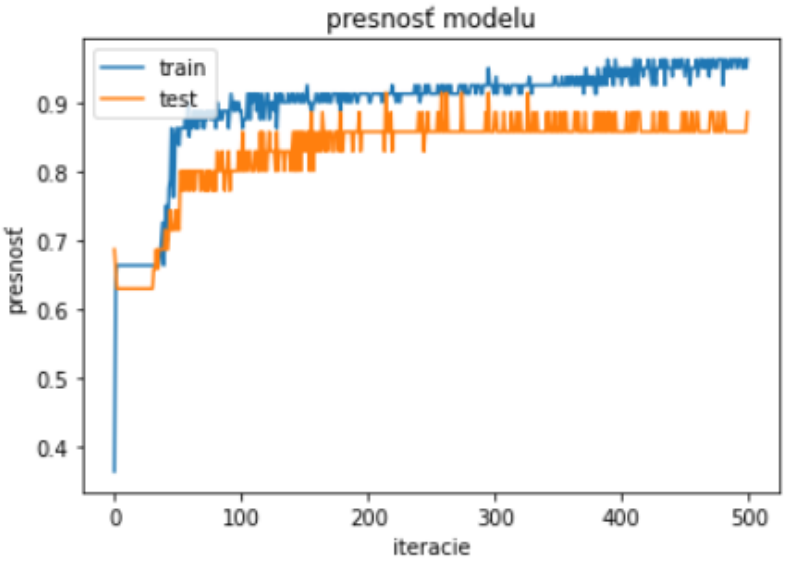
Natrénovaný model môžeme použiť na testovaciu vzorku aby sme videli akú hodnotu úspešnosti sme dosiahli. Táto hodnota sa rovná 93,10%.

```
scores = model.evaluate(X_test, y_test, batch_size=5)
```

```
6/6 [==============================] - 0s 2ms/step - loss: 0.2696 - accuracy: 0.9310
```

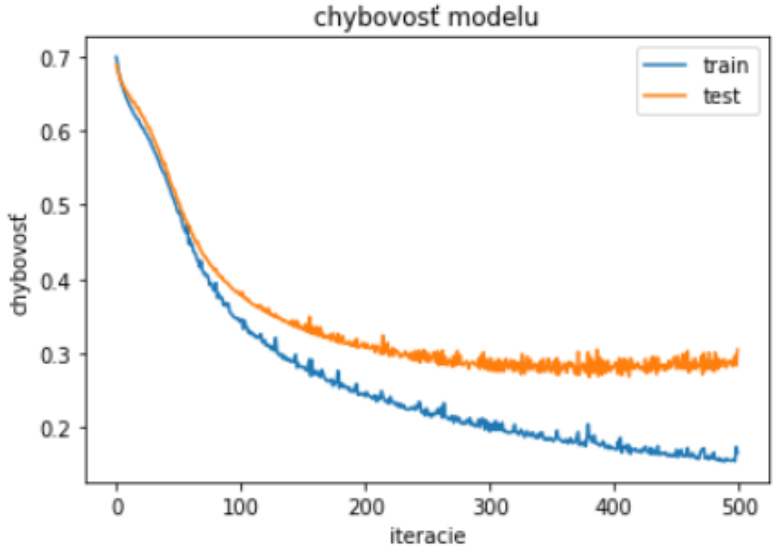
Obrázok 24 Výsledná presnosť neurónových sietí

S použitím knižnice „matplotlib.pyplot“ sme vykreslili ako sa menila presnosť modelu po každom opakovaní trénovania. Môžeme vidieť, že úspešnosť neurónovej siete sa približne po dvesto opakovaniach už výrazne nemení.



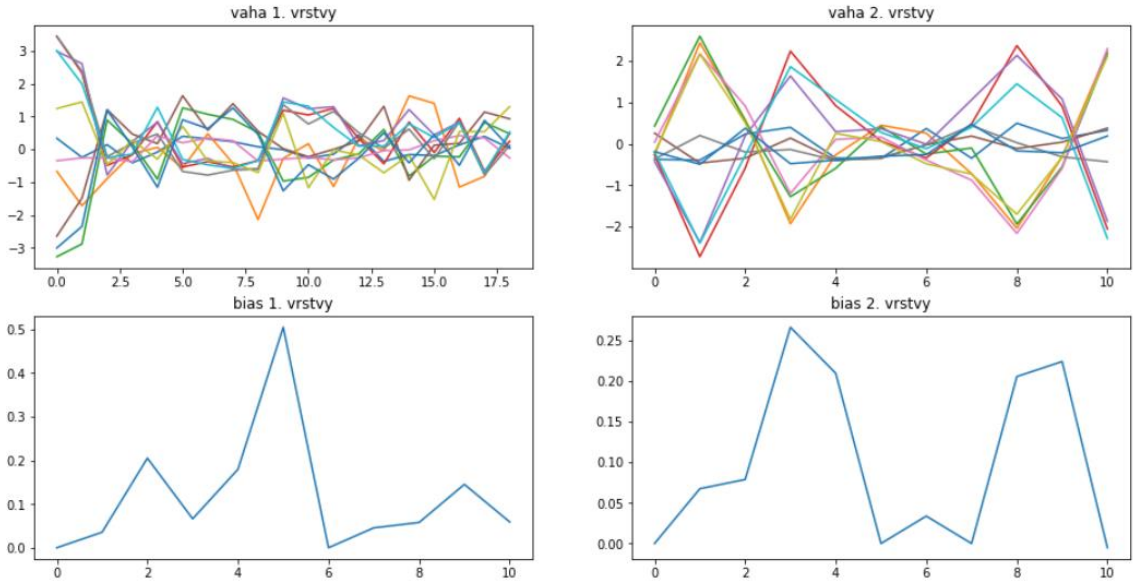
Obrázok 25 Zmena presnosti modelu počas trénovania

Rovnako tak môžeme na obrázku (obrázok 26) sledovať chybovosť modelu. Je zjavné, že čím je menšia chybovosť, tým je väčšia úspešnosť modelu. Preto môžeme tiež pozorovať, že chybovosť sa po dvesto opakovaniach takisto výrazne nemení.



Obrázok 26 Zmena chybovosti modelu počas trénovania

Na obrázku (obrázok 27) sme zobrazili výsledné váhy prvej a druhej vrstvy neurónovej siete. Vidíme, že v prvej vrstve bola dimenzia o výške devätnásť, tak ako sme to zadali, no v druhej vrstve už model vygeneroval len jedenásť dimenzií. V spodných dvoch tabuľkách máme zobrazené hodnoty „bias“ pre dané vrstvy.



Obrázok 27 Váhy prvej a druhej vrstvy neurónových sietí

Výsledné hodnoty poslednej vrstvy si môžeme ukázať priamym výpisom. Zobrazilo sa nám jedenásť váh a na konci jedna hodnota „bias“.

```
[array([[-0.25078893],
       [-2.850535  ],
       [-2.3592503 ],
       [ 1.2981306 ],
       [ 1.0650603 ],
       [-0.4807669 ],
       [-2.4185429 ],
       [-0.32150054],
       [-2.2880971 ],
       [ 1.1153152 ],
       [ 0.51244885]], dtype=float32), array([0.19321401], dtype=float32)]
```

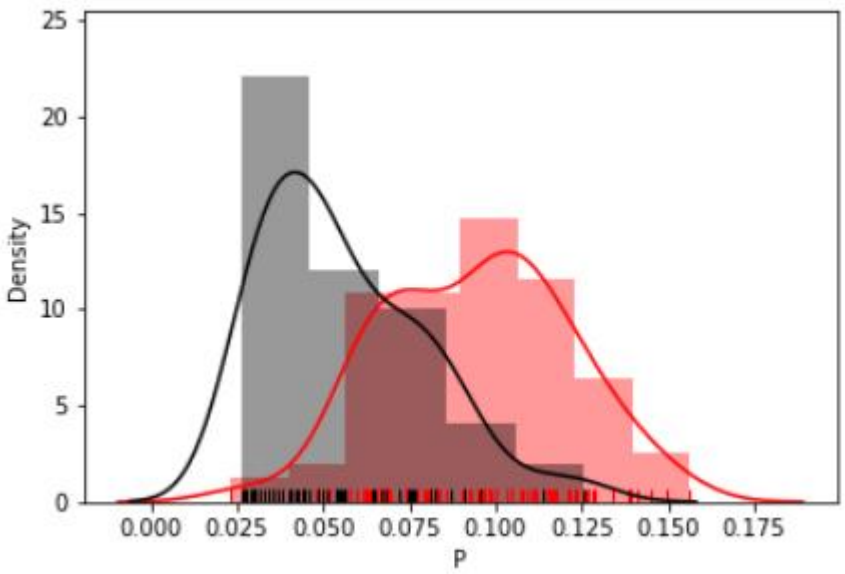
Obrázok 28 Váhy tretej vrstvy neurónových sietí

4.5 VYHODNOTENIE VÝSLEDKOV

V práci sme sa zamerali na detekciu falošných a pravdivých správ a tvrdení s témou Covid-19. Pri realizácii navrhnutej metódy sme pracovali s dátovou množinou obsahujúcou články z rôznych zdrojov.

Pri použití prvého klasifikátora - rozhodovacích stromov, ktoré patria medzi základné metódy strojového učenia, sme na základe našich výsledkov dospeli k týmto zisteniam.

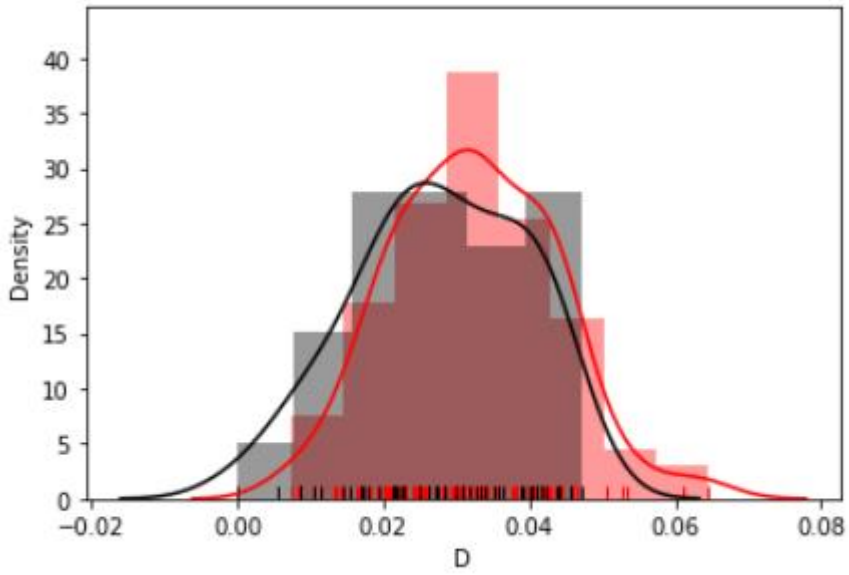
Pre falošné správy je typickejší vysoký počet zámen, čo sa napríklad dá pozorovať aj na grafe početnosti zámen. Falošné správy sme vyznačili červenou farbou a čiernou farbou sú vyznačené správy pravdivé. Na tomto grafe môžeme pozorovať, že hodnoty početnosti zámen falošných správ sa pohybujú v priemere okolo hodnoty 0,11. V pravdivých správach sa tieto hodnoty pohybujú okolo čísla 0,04 .To nám vysvetľuje, prečo v našich výsledkoch boli zámená v rozhodovacích stromoch najdôležitejším faktorom pri detekcii falošných správ. Preto pri porovnaní súboru falošných správ zo súborom pravdivých správ, sú tieto súbory od seba zreteľne rozlíšiteľné počtom zámen, tak ako to vidíme na grafe.



Obrázok 29 Relatívna početnosť zámen v pravdivých a falošných správach

Na druhej strane keď sa pozrieme na graf početnosti prísloviek rozdiel medzi hodnotami falošných a pravdivých správ nie je značne viditeľný. Fakt, že tieto hodnoty

sú si dosť podobné vysvetľuje, prečo rozhodovací strom pridelil nulovú dôležitosť početnosti prísloviek.



Obrázok 30 Relatívna početnosť prísloviek v pravdivých a falošných správach

Pri použití rozhodovacích stromov naša detekcia dosiahla úspešnosť o výške približne 80 %, čo považujeme sa veľmi dobrý výsledok. Jeden z dôvodov, prečo sa nám podarilo dosiahnuť lepšie výsledky môže byť aj fakt, že v bežnej praxi sa pri predspracovaní datasetu odstraňujú stop slová. K týmto slovám patria predložky, spojky a neplnovýznamové slová, ktoré sme ale v našej práci neodstraňovali a práve rozhodovacie stromy niektorým týmto slovám pridelili vysokú dôležitosť. Predpokladáme, že vymazaním niektorých stop slov a to najmä zámen (ty, ja, vy, ten, tento, ktorý), by sme mohli vymazať aj typické črty, ktoré sú charakteristické pre falošné správy. Tým sme potvrdili skúsenosti z lingvistických analýz štýlu písania falošných správ, ktoré uvádzajú, že pre falošné správy je typická forma písania príbehu. Ten obsahuje viac zámen, zatiaľ čo v skutočných správach je častejšie obsiahnuté vysvetlenie, ktoré používa viac podstatných mien a predložiek (Grieve, 2018).

S použitím druhého klasifikátora - neurónových sietí, sme očakávali vyššiu úspešnosť detekcie falošných správ. Získané výsledky našej implementácie to potvrdzujú. Zatiaľ čo úspešnosť za použitia rozhodovacích stromov sa pohybovala okolo 80 %, pri neurónových sieťach sa nám podarilo dosiahnuť úspešnosť výsledkov vyše 90 %. Na druhej strane je ale tiež dôležité pripomenúť, že neurónová sieť má aj svoju

nevýhodu, ktorou je značne vyššia výpočtová náročnosť. Pri našej implementácií sme to pocítili hlavne pri tom, ako sa učenie neurónovej siete opakovalo päťsto krát. Zároveň pri zvyšovaní hodnôt dimenzií sa každé opakovanie predĺžilo a to predĺžilo aj celý proces učenia. Tento fakt však nepovažujeme za veľkú prekážku, keďže výkon priemerného počítača v dnešnej dobe dokáže zvládnuť trénovanie aj zložitejšej neurónovej siete za niekoľko sekúnd, prípadne minút. Čo ale považujeme za väčšiu nevýhodu, je neschopnosť neurónových sieti interpretovať výsledky. Zatiaľ čo pri rozhodovacích stromoch sme videli hodnoty dôležitosti a podmienky, na základe ktorých sa model rozhodoval, pri neurónových sieťach jedinou interpretovateľnou časťou sú váhy jednotlivých neurónov. No samotné váhy nám pri zmenách dimenzií nevedia jednoznačne odpovedať na otázku, ktoré slovné druhy boli najdôležitejším faktorom pri detekcií falošných správ.

ZÁVER

O tom, že falošné správy známe ako fake news, sa stali vážnym problémom v súčasnosti, už nikto nepochybuje. Preto je veľmi dôležité, aby informačné technológie priniesli také automatizované riešenia ich odhaľovania a odstraňovania, ktoré by čo najviac zamedzili ich šíreniu.

Cieľom našej diplomovej práce bolo nájsť viacero metód strojového učenia, ktoré by vedeli čo najefektívnejšie a najpresnejšie detekovať falošné správy. Rovnako tak bolo cieľom tieto modely porovnať a interpretovať

Z tohto dôvodu sme sa rozhodli manuálne zozbierať dataset článkov, ktoré sa zaoberajú témou COVID-19. Pri rozhodovaní, či sa jedná o falošné alebo pravdivé články nám pomohlo najmä hodnotenie zdroja článku od občianskeho združenia *Košpirátori.sk*. Predpríprava dát pre ďalšie použitie spočívala najmä z úpravy formátu, uloženia datasetu a úpravy tagov článku. Rozhodli sme sa používať slovné druhy a triedy pre trénovanie rozhodovacích modelov, preto bolo potrebné odstrániť ostatné prebytočné informácie z datasetu a zamerať sa iba na prvé znaky tagov. V neposlednom rade sme rozdelili dataset na trénovacie a testovacie dáta v pomere 80:20.

V ďalšom kroku sme použili rozhodovacie stromy ako klasifikátor pre rozhodovanie, či sa jedná o falošný alebo pravdivý článok. Vyskúšali sme rôzne variácie rozhodovacích stromov pre dosiahnutie čo najlepších výsledkov a podarilo sa nám dosiahnuť úspešnosť modelu približne 80 %. Podrobná interpretácia výsledkov rozhodovacích stromov bola veľká výhoda tohto modelu.

V ďalšom modeli sme použili neurónové siete na rozlišovanie skutočných a falošných správ. Po odskúšaní rôznych štruktúr neurónových sietí sme najlepšie výsledky dosiahli pri použití troch vrstiev. Úspešnosť sa pohybovala okolo hodnoty 93 %, a preto sme mohli potvrdiť predpoklad, že neurónové siete sú presnejšie.

Napokon sme vyhodnotili výsledky a opísali niektoré zaujímavé zistenia, ktoré sme mohli pozorovať v našej práci. Na základe týchto zistení môžeme odporučiť, aby pri príprave datasetu v slovenskom jazyku neboli odstránené zámená ako jedna zo zložiek stopslov. Ako dôvod uvádzame naše výsledky z implementácie rozhodovacích stromov. Tie preukázali dôležitosť zámen pri rozlíšení falošných správ od skutočných.

Ako ďalšie odporúčanie uvádzame prednostné použitie neurónových sietí ako klasifikátora metódy strojového učenia pred použitím rozhodovacích stromov. Odôvodňujeme to dosiahnutou vyššou presnosťou v prípade použitia neurónových sietí.

Pri práci boli využité a aplikované vedomosti nadobudnuté počas štúdia na Katedre informatiky UKF Nitra a zároveň nové poznatky získané pri vypracovávaní tejto diplomovej práce. Oboznámili sme sa podrobnejšie s problematikou detekcie falošných správ a s implementáciou metód strojového učenia v danej oblasti.

Týmto môžeme skonštatovať, že na základe horeuvedeného sa nám podarilo cieľ práce naplniť a veríme, že výsledky našej práce poslúžia pre ďalšie použitie v praxi, prípadne pre ďalšie podrobnejšie skúmanie implementácie rozhodovacích stromov a neurónových sietí ako perspektívnych nástrojov automatizovanej detekcie fake news.

ZOZNAM BIBLIOGRAFICKÝCH ODKAZOV

BARTHOLOMEW R. E., RADFORD B.,2003, *Hoaxes, Myths, and Manias: Why We Need Critical Thinking*, Prometheus Books, 2003, 229 s., ISBN 978-1591020486

CONROY N., RUBIN V., CHEN Y., 2016, *Automatic deception detection: Methods for finding fake news*, [online]. [cit. 2021-02-21]. Dostupné na Internete: < https://asistdl.onlinelibrary.wiley.com/doi/full/10.1002/pra2.2015.145052010082>

FENG S., BANERJEE R., CHOI Y., 2012, *Syntactic Stylometry for Deception Detection*, [online]. [cit. 2021-03-01]. Dostupné na Internete: <https://www.aclweb.org/anthology/P12-2034.pdf>

FIGUEIRA A.,GUIMARAES N., TORGO L., 2018, *Current State of the Art to Detect Fake News in Social Media: Global Trendings and Next Challenges*, [online]. [cit. 2021-02-10]. Dostupné na Internete: < https:// www.researchgate.net/publication/327913024_Current_State_of_the_Art_to_Detect_Fake_News_in_Social_Media_Global_Trendings_and_Next_Challenges >

GRIEVE J.,2018, *The Language of Fake News*,[online].[cit. 2021-04-01]. Dostupné na Internete: <https://www.birmingham.ac.uk/news/thebirminghambrief/items/2018/09/the-language-of-fake-news.aspx>

JURKOVIČ, M., ČAVOJOVÁ, V., BREZINA, I. 2019. *Prečo ľudia veria nezmyslom.* Premedia Groups s.r.o. Bratislava, 2019. 312 s. ISBN 978-80-8159-757-2.

KAPUSTA J. OBONYA J., 2020, *Improvement of Misleading and Fake News Classification for Flective Languages by Morphological Group Analysis*, [online].[cit.2020-12-21]. Dostupné na Internete: <https://www.researchgate.net/publication/339071502_Improvement_of_Misleading_and_Fake_News_Classification_for_Flective_Languages_by_Morphological_Group_Analysis>

KŘUPKA J., KAŠPAROVÁ M., JIRAVA P., 2010, *Modelování kvality života pomocí rozhodovacích stromů,* [online].[cit.2020-11-30]. Dostupné na Internete: <http://www.ekonomie-management.cz/download/1331826751_3c30/11_krupka.pdf>

KUMAR S.,SHAH N., 2018, *False Information on Web and Social Media: A Survey*. . [online]. [cit. 2021-01-19]. Dostupné na Internete: <https://arxiv.org/pdf/1804.08559.pdf>

MICHELIS, C. G. De. 2004. *Protocolli Dei Savi Di Sion*. U of Nebraska Press, 2004. 419 s. ISBN 978-0-8032-1727-0.

ORAVEC, J., 1988. *Súčasný slovenský spisovný jazyk.* Bratislava: SPN, 1988 s. 27-29.

ORLOWSKI J. 2020, *Sociálni dilema*,,[online].[cit.2020-02-15].Dostupné na Internete: <https://www.netflix.com/title/81254224?s=a&trkid=13747225&t=fbm>

PRZECZEK, L. 2007. *Citáty slávnych ľudí.* [online]. [cit. 2021-01-10]. Dostupné na Internete: <https://citaty-slavnych.sk/citaty-o-lzi/?page=3>

PRZYBYLA P. 2019, *Capturing the Style of Fake News*, [online].[cit.2021-02-21]. Dostupné na Internete: <https://home.ipipan.waw.pl/p.przybyla/bib/capturing.pdf >

RUCHANSKY N.,SEO S., LIU Y., 2017, *CSI: A Hybrid Deep Model for Fake News Detection*, [online].[cit.2020-12-21]. Dostupné na Internete: <*https://arxiv.org/pdf/1703.06959.pdf*>

SHIVAM B. P., PRADEEP K. A., 2018, *Media-Rich Fake News Detection: A Survey*, , [online]. [cit. 2021-03-03]. Dostupné na Internete: <*https://ieeexplore.ieee.org/document/8397049*>

SHU K.,SLIVA A., WANG S., TANG J., 2017, *Fake News Detection on Social Media: A Data Mining Perspective*. [online]. [cit. 2021-03-05]. Dostupné na Internete: <https://arxiv.org/pdf/1708.01967.pdf>

SLOVENSKÁ INFORMAČNÁ SLUŽBA BRATISLAVA. 2020. *Správa o činnosti Slovenskej informačnej služby za rok 2019.* Bratislava, [online]. [cit. 2020-11-20]. Dostupné na Internete: <https://www.sis.gov.sk/pre-vas/sprava-o-cinnosti.html>

ULDRICH M., JURCZYK T., 2014, *Neuronové sítě a jejich využití*, [online]. [cit.2021-02-21]. Dostupné na Internete: <http://www.statsoft.cz/file1/PDF/Statsoft_neuronove_site.pdf?fbclid=IwAR0kUN1gdtsRjljXXY4WoJwstT87RVvtk7A_Oz9YvQQV8zYQuUygKSH_XZo>

ZHOU X.,ZAFARANI R., 2018, *A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities*, [online]. [cit. 2021-02-22]. Dostupné na Internete: < https://ui.adsabs.harvard.edu/abs/2018arXiv181200315Z/abstract >

ŽÚBOROVÁ V., BORÁROSOVÁ I., 2019, *DE-FAKE IT !*, Bratislava Policy Institute, 2019, 144 s., ISBN 978-80-973236-5-3