

Fake and real news

Abstrakt






V tomto projekte sa zaoberám prácou s datasetom falošných a pravých správ a ich správnej kategorizácii pomocou supervised learning. Medzi iným, porovnávam vplyv veľkosti datasetu na presnosť predikcie, vplyv prítomností titulku a textu na presnosť predikcie a slová ktoré najviac naznačujú pravdivostnú hodnotu správ.

Úvod

Cieľom tohto projektu je zistiť závislosť presnosti kategorizácie od veľkosti datasetu a jeho ďalšie podobné úpravy. Pri práci budem primárne využívať Python a jeho knižnice.

Dataset

Dataset je získaný zo stránky [Kaggle](#) a skladá sa z dvoch csv súborov, jeden obsahujúci pravdivé správy a druhý falošné. Dataset je tvorený 4 stĺpcami: titulok správy, text správy, téma správy a dátum vydania. Pri mojom projekte využívam len prvé dva stĺpce. Stĺpce téma správy a dátum vydania obsahujú informácie unikátne pre jednu z pravdivostných hodnôt a tým pádom robia klasifikáciu dát triviálnu.

|  title |  text |  subject |  date |
|---|---|--|--|
| The title of the article | The text of the article | The subject of the article | The date that this article was posted at |
| 20826 unique values | 21192 unique values | politicsNews 53% worldnews 47% |  13Jan16 31Dec17 |
| As U.S. budget fight looms, Republicans flip their fiscal script | WASHINGTON (Reuters) - The head of a conservative Republican faction in the U.S. Congress, who voted... | politicsNews | December 31, 2017 |
| U.S. military to accept transgender recruits on Monday: Pentagon | WASHINGTON (Reuters) - Transgender people will be allowed for the first time to enlist in the U.S. m... | politicsNews | December 29, 2017 |
| Senior U.S. Republican senator... | WASHINGTON (Reuters) - The special | politicsNews | December 31, 2017 |

Metódy a popis kódu

Načítanie datasetu

Na začiatku treba dáta načítať, na čo som využil knižnicu Pandas. Keďže dáta samé o sebe neobsahujú informáciu o pravdivosti, pridal som k nim ďalší stĺpec vyjadrujúci túto hodnotu, pre falošné správy som priradil 0 a pre pravé 1. Napriek tomu že tento stĺpec neskôr separovaný ako list Y pre samotné učenie. Jeho pripojením som uľahčil miešanie a vzorkovanie dát. Dáta však boli stále rozdelené do dvoch premenných., takže som ich

musel zlepíť, premiešať pre ich rovnomerné rozloženie a preindexovať. Z výkonnostných dôvodov som ešte dodatočne zmenšil dataset na tretinu pôvodnej veľkosti. (Obr 1)

```
false = pd.read_csv("Fake.csv")
false["truth"] = 0
true = pd.read_csv("True.csv")
true["truth"] = 1
print("Data loaded")
news_all = pd.concat([false, true])
news_shuffled = sklearn.utils.shuffle(news_all)
news_cut = news_shuffled.sample(frac = 0.3)
news_cut = news_cut.reset_index(drop = True)
print("Data merged")
```

Obr 1

Spracovanie datasetu

Takto upravený dataset, konkrétne prvé dva stĺpce, však bolo treba ďalej spracovať na formu spracovateľnú metódami knižnice sklearn. Stĺpce som spracovával oddelene na zachovanie informácie či ide o názov alebo text správy. Na spracovanie som sa inšpiroval jedným z praktických cvičení ktoré sa touto témou zaoberalo.

Ako prvé som odstránil takzvané *stop words*, teda slová, ktoré sa často používajú v jazyku, ale nemajú veľkú hodnotu pre analýzu textu. Zoznam týchto slov som získal z knižnice *nltk*. Všetok text som zároveň zmenil na malé písmená, čo zabráni duplicitne slov len následkom ich polohy vo vete. Síce sa tým mierne stráca informačná hodnota dát, keďže písanie uppercase som je typické pre polopravdivé alebo falošné správy, účelom tohto projektu je kategorizácia na základe obsahu a nie formy.

Upravený text bol síce o poznanie kratší a mohol by sa použiť na učenie, avšak slová majúce viacero stále spôsobujú duplicitné skreslenie informácií. Bolo potrebné ich dať do ich základného tvaru, načo som využil metódu *PorterStemmer* z *nltk.stem.porter*. Z textu som následne ešte odstránil nechcené symboly ako bodky, čiarky a zátvorky .

Z takto upraveného textu môžem vytvoriť moju slovnú zásobu, pričom titulok a text ju zdieľajú. Najprv som dáta tokenizoval, využil som na to zase knižnicu *nltk*. Zo zoznamu tokenov som vytvoril ich frekvenčnú tabuľku a vrchných 2500 som použil ako moju slovnú zásobu.

Po získaní slovnej zásoby som už len zmenil názov a text na polia, kde jednotlivé prvky vyjadrujú počty slov v texte. Tieto dva stĺpce som nakoniec zlúčil do jedného zoznamu, takže ich ako vstupné X mám zoznam zoznamov dĺžky 5000 prvkov (2 x dĺžka slovnej zásoby). Ako Y som jednoducho použil posledný stĺpec s pravdivosťnými hodnotami.

Trénovanie s rôznymi veľkosťami

V ďalšej časti som trénoval klasifikáciu s rôzne veľkými datasetmi. Dáta som rozdelil pomocou malej funkcie na trénovaciu a testovaciu časť a následne som natrénoval

klasifikáciu pre 10 rôzne veľkých datasetov, pričom som zaznamenával čas a presnosť odhadu, ktoré som následne využil na vytvorenie 3 grafov. (Obr 2)

Okrem toho natrénujem dve separátne inštancie pre dataset, jednu bez textu a jednu bez názvu správy. Nakoniec využijem *Random Forest Clasifier*, pomocou ktorého môžem jednoducho zistiť ako majú jednotlivé slová vplyv na klasifikovanie správ.

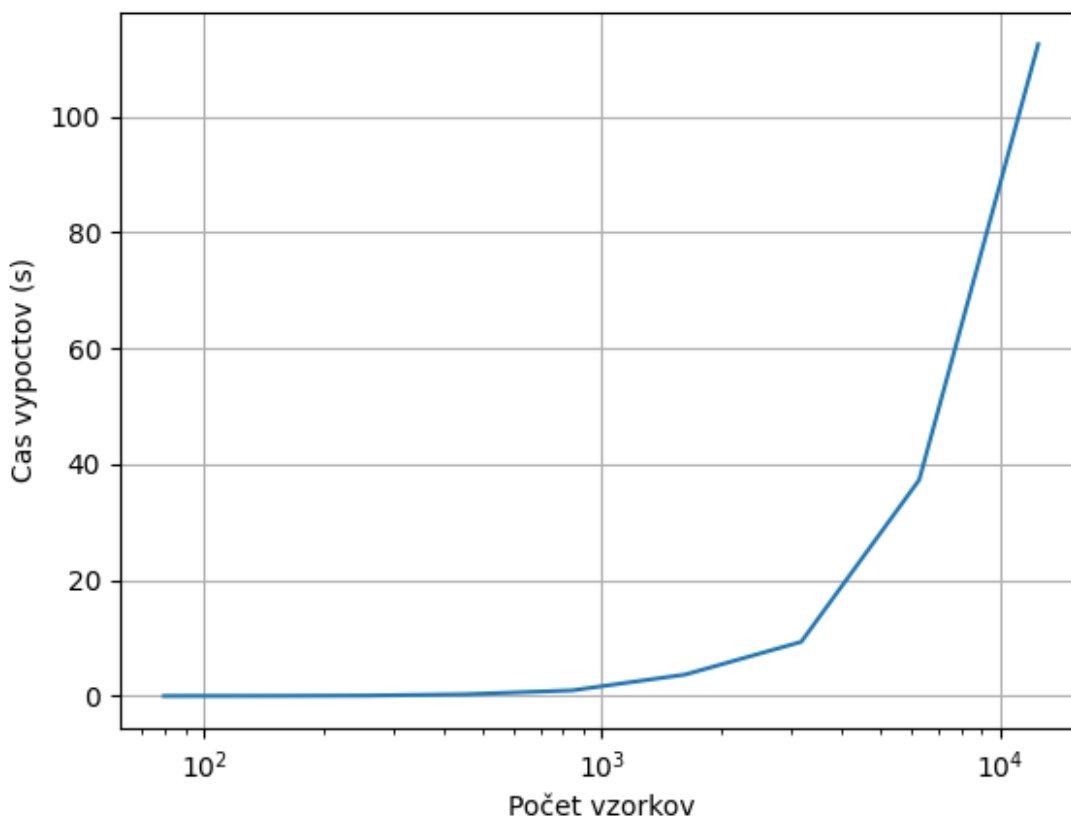
```
def cal(data, fr, size):  
    datax = split(data[0][:int(fr*size)])  
    datay = split(data[1][:int(fr*size)])  
    start_time = time.time()  
    svcF = svm.SVC(C=1, kernel='rbf')  
    svcF.fit(datax[0], datay[0])  
    return (time.time() - start_time, svcF.score(datax[1], datay[1]), fr)
```

Obr 2

Vyhodnotenie

Vplyv veľkosti datasetu na klasifikáciu

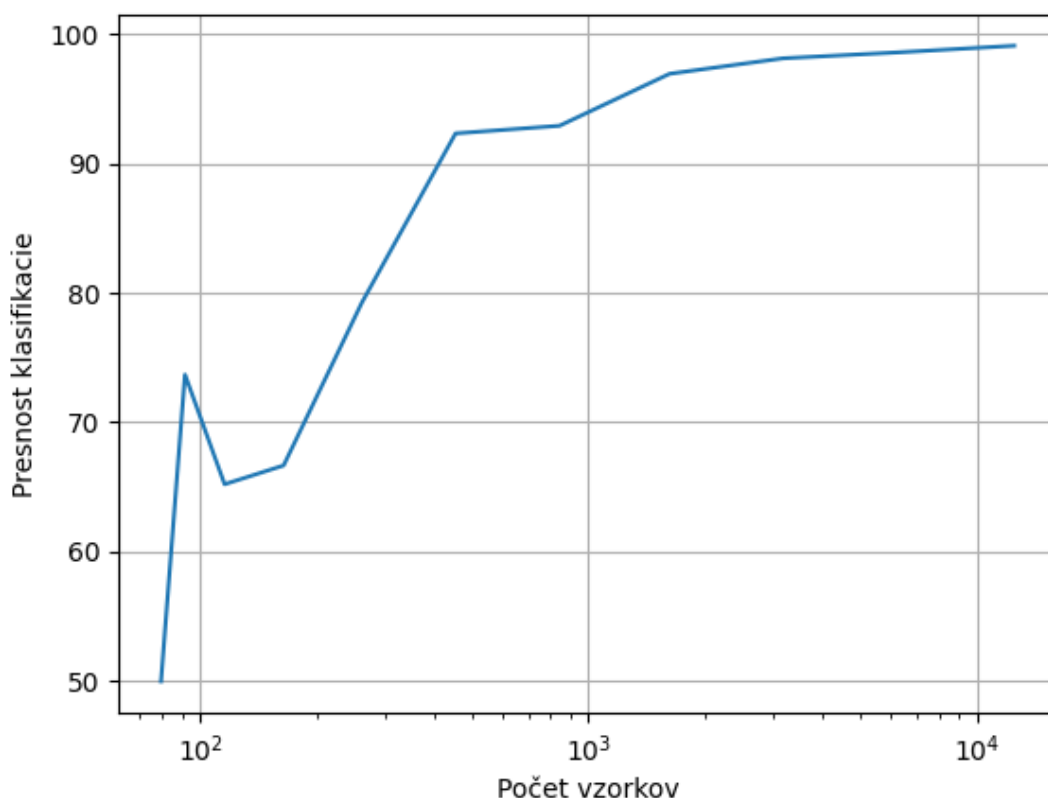
Ako prvé som vyhodnotil vplyv veľkosti datasetu na jeho klasifikáciu, konkrétne na rýchlosť výpočtov a presnosť následného modelu. Dáta som vyčítal primárne grafov.



Obr 3

Ako môžeme vidieť na obrázku 3, s veľkosťou datasetu rastie aj čas potrebný na správne fitnutie modelu. Tento rast je viditeľne exponenciálny a preto aj pomerne malý rozdiel v jeho veľkosti môže násobne urýchliť výpočty. Z obrázku môžeme vidieť že pri našom datasete dochádza k zlomu približne pri 3000-4000 vzorkách.

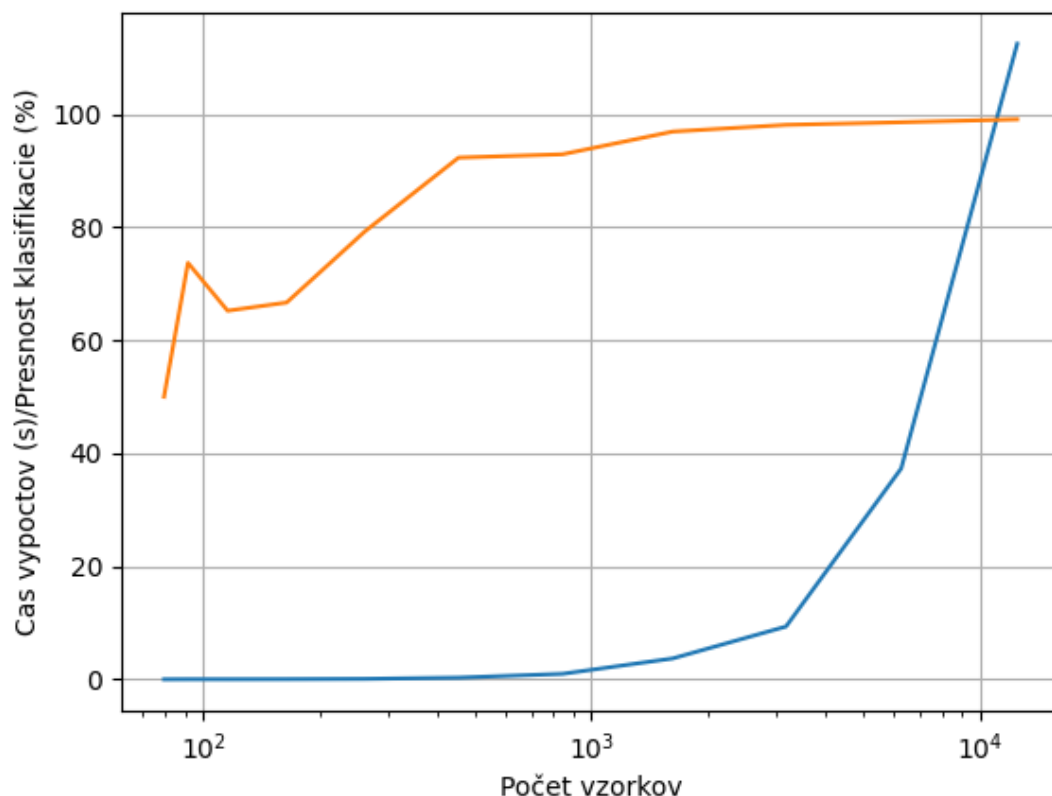
Na obrázku 4 môžeme vidieť závislosť presnosti klasifikácie od počtu vzoriek. Pri nižšom je klasifikácia nielen menej presná ale jej presnosť je ťažšie predvídateľná, väčší počet vzoriek nezaručí presnejšiu klasifikáciu. Model začína dosahovať presnejšie klasifikáciu pri 2000-3000 vzoriek, vyššie počty vzoriek síce zaručujú väčšiu presnosť ale majú klesajúce výnosy.



Obr 4

Pri výbere vhodnej veľkosti vzoriek teda musíme nájsť taký počet vzoriek, aby čas výpočtov nebol neprimerane veľký ale ani aby netrpela presnosť modelu. Keďže oba grafy zdieľajú jednu os, môžeme ich vykresliť do spoločného grafu (Obr 5).

Z tohto grafu môžeme vyčítať že vhodný počet vzoriek je medzi 1000 až 3000. Tento výber je samozrejme ovplyvnený požadovanou presnosťou respektíve rýchlosťou klasifikovania dát, avšak pre moje účely som použil 2693 vzoriek, čo je približne 20% plnej trénovacej množiny. Pomocou týchto vzoriek som testoval ďalšie časti tohto projektu.



Obr 5

Vplyv prítomnosti názvu/textu správy na presnosť predikcie

Ako ďalšie som skúmal potrebu prítomnosti názvu a textu správy pre korektnosť predikcie. Separácia dát názvu a textu bola pomerne jednoduchá, keďže prvých 2500 prvkov patrili k názvu a druhých 2500 k textu.

```

-----
Traning accuracy for all SVM: 0.9481481481481482
--- 3.137705087661743 seconds ---
No text
Traning accuracy for no text SVM: 0.8740740740740741
--- 2.0116326808929443 seconds ---
No title
Traning accuracy for no title SVM: 0.9481481481481482
--- 1.5317018032073975 seconds ---

```

Obr 6

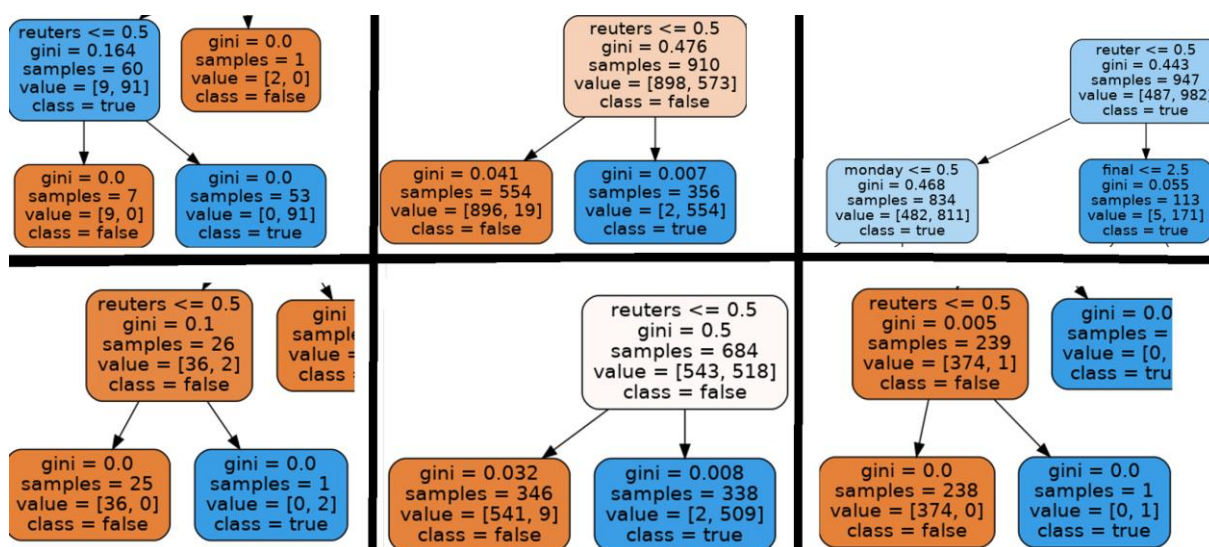
Ako je vidieť na obrázku 6, predikcia s názvom aj textom dosahuje vysoké hodnoty presnosti, avšak trpí v oblasti času. To bolo očakávané, keďže má k dispozícii najviac dát. Modeli tréňované len pomocou titulku správy majú pomerne výrazné zníženie presnosti predikcie avšak o niečo rýchlejšie tréňovanie. Používanie iba textu bez titulku naopak zachovalo presnosť plných dát pri našom rozsahu presnosti a tento výsledok bol dosiahnutý ešte

v lepšom čase. Tento čas je ešte o niečo lepší než pri trénovaní len na názvoch napriek rovnako veľkému datasetu. To je pravdepodobne spôsobené lepšie separovateľnými dátami.

Kľúčové slová

Pri rozhodovaní o pravdivosti datasetu na základe počtu slov, sú samozrejme niektoré slová viac významné ako iné. Pre odhalenie týchto slov som použil *Random Forest Classifier* a následne som zo vzniknutých stromov vyčítal, ktoré slová majú aký vplyv. Obrázky celých stromov sú priložené v prílohe, keďže ich rozmer by ich spravil nečitateľnými.

Najviac významné je pravdepodobne slovo **Reuters** (spravodajská agentúra), v 5/6 stromov je jeho absencia príznakom falošných správ. (Obr. 7)



Obr 7

Medzi ďalšie významné slová patria: via, know, trade, donald a aj fact, ktorého prítomnosť zvyšuje šancu že daná správa fakt nie je. Naopak prítomnosť trade takmer zaručuje pravdivosť informácií.

Analýza výsledkov

Z výsledkov plynie, že detekciu falošných správ vieme natrénovať na pomerne malom množstve dát, čo značí že buď:

- Falošné správy sú ľahko odhaliteľné pomocou ich slovníka
- Dataset má chabne vybrané pravé správy a falošné, čo spôsobuje skreslenie výsledkov

V našom prípade ide pravdepodobne o kombináciu oboch alternatív, pravdepodobne s miernym smerovaním k možnosti b), keďže som počas práce s ním odhalil iné nedostatky, ako napríklad nepoužiteľnosť dátumov pri klasifikácii.

Ako ďalšie som zistil, že správna kategorizácia sa opiera primárne o text, keďže absencia názvu nepriniesla merateľný rozdiel v presnosti. Preto by bolo výhodnejšie pri ďalšej práci s týmto datasetom názvy rovno ignorovať, pokiaľ sa teda výskum netýka priamo nich.

Z kľúčových slov vyplýva, že správa obsahujúca slovo Reuters, teda názov spravodajskej služby, mnohonásobne zvyšuje šancu, že správa bude pravdivá. Toto slovo by mohlo byť pravdepodobne zameniteľné za názov lokálnych novín, keďže nejde o slovo ako také ale o jeho význam ako značka

poctivosti. Ďalšie významné slovo via, by sa dalo definovať skôr ako podporné slovo k Reuters, keďže sa často používa na referovanie k zdroju správy (via Reuters). Prítomnosť slova know – vedieť, naopak znižuje pravdivosťnú hodnotu správ, keďže sa vo falošných správach redaktori často odvolávajú k vedeniu – my vieme lepšie, vieme čo je pravda. Podobná situácia nastáva aj pri slove fact. Slovo trade, keďže obchodovanie ide o pomerne nesenáčajnú tému ktorou sa ťažko posúvajú názory čitateľov.

Ako posledné sa môžeme pozrieť na top 10 slov vo frekvenčnej tabuľke

- trump – 42029
 - Trump bol prezident USA väčšinu intervalu tohto datasetu, takže je pochopiteľné že je témou číslo 1
- said – 38926
 - Veľmi časté slovo v správach, ktoré často parafrázujú slová politikov, hercov alebo iných významných ľudí
- us – 20213
 - Skratka pre USA, vzhľadom že tento dataset je silno USA-centrický, je pochopiteľné že USA bude spomínané často
- would – 16547
- state – 16274
 - Podobne ako pri us, týka sa vnútornej politiky USA
- presid – 14757
 - Koreň slova president, jeho častý výskyt je spôsobený častou kombináciou slov president Trump
- republican – 11318
 - Zase ide o vnútornú politiku USA, konkrétne o stranu vtedajšieho prezidenta
- say – 11169
- one – 11118
- peopl – 10772

Záver

Napriek tomu že niektoré ciele neboli splniteľné, v tomto projekte som prišiel na množstvo užitočných informácií. Pravdivosť článku ma menší vplyv na názov ako na samotný text. Pravdivé články často obsahujú kľúčové slová ktoré ich jednoznačne odlišujú od falošných, ako Reuters, via, trade a naopak, know, fact. Samozrejme že ide vytvoriť falošnú správu ktorá by obsahovala pozitívne slová a vyhýbala sa negatívnym, avšak v prípade Reuters, alebo iného názvu novín by to mohlo priniesť právne následky.

Prílohy

[GitHub](#)

[Dataset](#)