Projekat

Klasifikacija tweet-ova

Predmet: Vještačka inteligencija

Profesor: dr Savo Tomović

Student: Luka Božović

Indeks i smjer: 24/19 C

Sadržaj:

1. OPIS PROBLEMA

Naš zadatak bilo je implementiranje modela koji analizira postove sa društvene mreže Twitter i prepoznaje da li se oni odnose na upozorenje na neki hitni događaj, na primjer vremensku nepogodu. Ovakav model bio bi dio sistema koji bi pratio objave korisnika Twitter-a i generisao upozorenja odgovarajućim službama. Na raspolaganju je dataset koji sadrži tekst objave, ključne riječi i lokaciju. Objave se klasifikuju kao realno upozorenje (labela 1) ili ne (labela 0). Dataset je predat u vidu csv fajla koji sadrži 7613 vrsta.

2. Učitavanje dataseta i preprocesiranje

Jedna od ključnih stvari koja doprinosi kvalitetnom kreiranju modela mašinskog učenja jeste pravilno odrađeno preprocesiranje podataka, naravno kada imamo vjerodostojno skupljene podatke.

Učitavanje zadatog csv fajla izvršeno je pomoću biblioteke ,,pandas'', tačnije, pomoću njene funkcije read_csv. Nakon toga za preprocesiranje se koriste biblioteke ,,re'' za regularne izraze i ,,nltk'' od koje se primarno koriste ,,WordNetLematizer'' i ,,wordnet''.

Ključni koraci koji su odrađeni prije pretvaranja zadatog teksta u format nad kojim se mogu primjeniti algoritmi mašinskog učenja su:

- Uklanjanje linkova iz teksta
- Pretvaranje svih slova u mala
- Uklanjanje brojeva, specijalnih simbola kao i znakova interpukcije i suvišnih bjelina
- Uklanjanje riječi koje nam ne nose kvalitetnu informaciju, npr. the, a, an, is, are – odnosno, tzv. stopwords.
- I kao najbitniji i najkompleksniji korak, svođenje riječi na korijenski oblik pomoću lematizer-a i tokenizer-a, gdje se koriste funkcije koje određuju službu riječi u rečenici, a nakon toga ih svode na osnovni oblik.

S obzirom na robustnost biblioteke ,,nltk“, samim tim što se prolazi kroz ogromnu bazu riječi engleskog jezika, kao i na ostale korake koji se sprovode, preprocesiranje datog teksta oduzima određeno vrijeme, pa je dati ,,očišćeni tekst“ sačuvan u poseban fajl, pomoću ,,joblib“ biblioteke. Na taj način se skratilo vrijeme čekanja da se naš program izvrši, jer se algoritami mašinskog učenja primjenjuju na spremnom tekstu koji se učitava iz fajla.

Pošto ti algoritmi ne mogu da rade na tekstu, bilo je potrebno pretvoriti isti u njima pogodan oblik odnosno u vektor, što je odrađeno pomoću biblioteke ,,sklearn“ i njenim klasama ,,CountVecotrizer“ i ,,TfidTransformer“, posle čega su naši podaci unutar ,,numpy“ niza. Za pretvaranje je eksperimentalno utvrđeno da se na našem datasetu algoritmi najbolje ponašaju kada se koristi oko 1500 relevantnih riječi (u programskom kodu nazvanih features), gdje su granice da bi se neka riječ smatrala relevantom: minimalni broj pojavljivanja 10 i maksimalni broj 80%.

Takođe, prije primjene je dataset podijeljen na dva dijela – trening i test, u razmjeri 80:20.

3. Primjena algoritama mašinskog učenja za klasifikaciju

Klasifikatori koji su korišćeni za naš zadatak (zbog globalnog standarda na engleskom jeziku):

- Logistic regression
- Naive Bayes
- Suport vector machine
- Kneighbors
- Random forest
- Neuronska mreža

Dati algoritmi su implementirani u ,,sklearn" biblioteci, odakle su importovani. Parametri u njihovim pozivima su eksperimentalno podešeni, tako da davaju što bolje rezultate.

Termini koji se pominju u narednom tekstu i koje je potrebno razjasniti, a koji su opet predstavljeni na engleskom jeziku, jer se kao takvi koriste svuda su:

True positives (TP): Predviđeni pozitivno su i stvarno pozitivni

False positives (FP): Predviđeni pozitivno, a u stvari su negativni

True negatives (TN): Predviđeni negativno i stvarno su negativni

False negatives (FN): Predviđeni negativno, a u stvari su pozitivni

Za evaluaciju dobijenih rezultata korišćene su sledeće metode:

- Accuracy - TP + TN / TP + TN + FP +FN
- Precision - TP / TP + FP
- Recall - TP / TP + FN
- F1 score - 2 * precision * recall / precision + recall

4. Rezultati nakon završnog testiranja

	Logistic regression		Naive Bayes		Suport vector machine		Kneighbors		Random forest		Neural network	
Accuracy	0.80		0.77		0.81		0.74		0.79		0.76	
Precision	0	0.79	0	0.73	0	0.79	0	0.72	0	0.79	0	0.78
	1	0.81	1	0.85	1	0.85	1	0.78	1	0.81	1	0.73
Recall	0	0.88	0	0.92	0	0.91	0	0.89	0	0.88	0	0.82
	1	0.69	1	0.58	1	0.67	1	0.53	1	0.67	1	0.69
F1 score	0	0.83	0	0.81	0	0.85	0	0.80	0	0.83	0	0.80
	1	0.74	1	0.69	1	0.75	1	0.64	1	0.73	1	0.72

U datoj tabeli možemo vidjeti rezultate testiranja i primjećuje se da imamo zadovoljavajuće procente uspješnosti. Vjerovatno očekivano, Suport vector machine dao je ponajbolje rezultate, a približno dobri su Logistic regression i Random forest.

Na zadatom datasetu imali smo ne preveliku razliku u broju labelisanih primjera klase 0 i klase 1. Ipak, ako realno sagledamo zadati problem, u praksi ćemo vidjeti mnogo više primjera klase 0. Takođe, ako pravimo ovakav neki klasifikator željeli bismo da nam da malo bolje rezultate za klasu 1 od ovih postojećih, konkretnije malo bolji recall. Iz tog razloga biće predstavljena još jedna tabela, koja se dobija nakon davanja veće važnosti klasi 1, odnosno onoj klasi koja prepoznaje upozorenja. Neki od ovih algoritama nisu podobni za ovakvo skaliranje pa će za njih ostati rezultati iz prethodne tabele.

	Logistic regression		Naive Bayes		Suport vector machine		Kneighbors		Random forest		Neural network	
Accuracy	0.77		0.77		0.78		0.74		0.79		0.76	
Precision	0	0.83	0	0.73	0	0.80	0	0.72	0	0.79	0	0.78
	1	0.71	1	0.85	1	0.75	1	0.78	1	0.78	1	0.73
Recall	0	0.76	0	0.92	0	0.82	0	0.89	0	0.86	0	0.82
	1	0.79	1	0.58	1	0.73	1	0.53	1	0.69	1	0.69
F1 score	0	0.79	0	0.81	0	0.81	0	0.80	0	0.82	0	0.80
	1	0.75	1	0.69	1	0.74	1	0.64	1	0.73	1	0.72

Radi eventualnog sagledavanja podataka iz drugog ugla podaci iz prvog primjera će biti predstavljeni i pomoću matrice koja sadrži informacije u formatu

TP FP

FN TN

tzv. confusion matrix.

Logistic regression	
782	110
188	443

Naive Bayes	
815	63
273	372

Suport vector machine	
769	93
210	451

Kneighbors	
779	102
323	319

Random forest	
784	74
244	421

Neural network	
699	160
196	468