FAKE NEWS PROJECT

Adam John - kwz566
Institute of Data Science
University of Copenhagen
kwz566@alumni.ku.dk

Mikkel Keller - trq281
Institute of Data Science
University of Copenhagen
trq281@alumni.ku.dk

Rasmus Voss - khp836
Institute of Data Science
University of Copenhagen
khp836@alumni.ku.dk

Søren Strøyberg - lds147
Institute of Data Science
University of Copenhagen
lds147@alumni.ku.dk

4. april 2024

Indhold

1 Part 1: Data Processing

Part 1 af denne opgave går ud på at læse, strukturerer og rense vores datasæt. Til at læse datasættet gør vi brug af $pandas$-bibloteket, pga det er effektiv databearbejdning og det arbejder godt sammen med andre python-bibloteker. Et skridt i databehandlingen er at håndtere manglende værdier. I vores projekt har vi brugt pandas' $fillna$ funktion til at erstatte NaN-værdier med tomme strenge. Dette valg blev truffet for at sikre, at vores efterfølgende tekstbehandlingsmetoder, såsom $tokenization$ og $stemming$, kan arbejde uden afbrydelse. Re-biblioteket anvendes til at udføre regulære udtryksoperationer, hvilket gør det muligt at søge, ændre og manipulere tekst baseret på mønstre for tekstrensning og behandling. Re-bibloteket bruger vi til at lave en manuel rensningsfunktion. Funktionen renser tekst ved at normalisere den, erstatte specifikke mønstre med placeholders og fjerne uønskede tegn. Dette mener vi er unødvendig $noise$ i datasæt som vi fjerner til videre analyse.

Når dataen er renset udnytter vi $NLTK$-bibloteket til at lave en $datapreprocessingpipeline$ som skal $tokenize$ dataen, fjerne $stopwords$ og udføre $stemming$ på datasættet. $Tokenization$ bryder teksten ned i mere håndterbare enheder, fjernelse af stopord reducerer størrelsen af datasættet, men bibeholder den vigtigste information, mens stemming kombinerer flere varianter af et ord til en enkelt repræsentation. Den generelle ide med disse metoder er som nævnt at reducere meningsløs data, hvilket vi tydeligt ser i outputtet. Denne sektion viser at vi har reduceret $vocabulary$ fra 16.827 til 11.519. Nu fortsætter vi til det omfattende datasæt med navnet "**995,000 rows.csv**". Vi læser datasættet på samme måde som tidligere, og for at få et overblik over dataene samt identificere mønstre, udfører vi følgende handlinger:

- Undersøger hvor mange af hver type nyhed der er i datasættet, vha. et histogram. Her udnyttes $matplotlib$-bibloteket som ligesom $NLTK$ nemt arbejder sammen med $pandas$. Det fremkommer at typerne "$reliable$" og "$political$" har den største repræsentation. Yderligere bemærker vi, at der er 43.534 artikler, som er kategoriseret som "$unknown$". Derudover observerer vi flere typer, som potentielt kan grupperes enten som "$reliable$"eller "$fake$", eller ingen af disse kategorier.
- Undersøger de 10 største domæner og herunder hvilke type artikler de udgiver. Det viser sig at hver domæne kun udgiver en type artikler, det stiller så spørgsmålet om det virkelig kan være en objektiv kategorisering?
- Undersøger hvor mange domæner datasættet indeholder i alt og herunder hvor mange domæner der udgiver "$reliable$" artikler. Resultatet viser at der er 683 domæner hvoraf 80 af dem har udgivet artikler af typen "$reliable$". Dette giver os en af de første indsigter i nutidens medier, som viser en tendens til at prioritere mændgen af output højere end sandhed/kvalitet af output.

Efter disse indsigter er blevet noteret kører vi vores "data preprocessing pipeline" på det store datasæt, hvor vi ligesom før fjerner unødig "støj" fra datasættet. Til sidst deler vi det bearbejdet datasæt tilfældigt op i en fordeling af 80/10/10. Ved at opdele vores datasæt til 80 procent træning, 10 procent validering og 10 procent test sikrer vi, at vi har en solid mængde data til at træne vores modeller, mens vi stadig har tilstrækkeligt med data til at validere og teste modellernes performance. Træningssættet anvendes til at opbygge modellens viden, valideringssættet bruges til at finjustere parametre og forhindre overfitting, og testsættet giver en objektiv vurdering af modellens endelige effektivitet. Til slut grundet dataens størrelse gemmer vi disse opdelinger af datasættet som csv filer, sådan at vi nemt kan tilgå dem igen i fremtiden uden at skulle køre de gamle koder igennem.

2 Part 2: Simple Model

I denne del af opgaven deler vi dataen op i to kategorier og opretter labels, som vi skal bruge vores machine-learning model til at sigte efter at kunne forudsige. Vi deler altså kolonnen 'type' op i 1 taller og 0 taller, hvor 1 står for reliable news og 0 for fake news. Vi har valgt at dele typerne op på følgende måde:

- **Pålidelig = Reliable og political** Vi antager at nyhedsartikler som (reliable og political) kan betragtes som troværdige og upartiske. De leverer faktuel information og undgår at fremsætte udokumenterede påstande eller vildledende information.
- **Fake = Fake, bias, conspiracy, hate, junksci, rumor, unreliable og satire** Kategorier som 'unreliable', 'fake', 'clickbait', 'conspiracy', 'junksci', 'bias', 'political' og 'hate' er alle kategorier, som prøver at vildlede læseren altså der manipuleres med læseren til at tænke i en bestemt retning. Der er også tit en masse overdrivelse i disse typer af news, hvilket fører til fake news. De kan sprede misinformation, propaganda eller konspirationsteorier.

2.1 Valg af Baseline-model

Vi har valgt at bruge en naive-bayes som baseline-model til vores fake news genkendelse. Den er udelukkende trænet på tekst fra kolonnen "stemmed_content". Den starter med at indlæse og forbehandle dataene og som tidligere beskrevet konverteres "type" til en binær værdi for at repræsentere "fake" og "reliable" nyheder Koden udnytter en TFIDF-Vectorizer til at transformere tekstindholdet til numeriske repræsentationer, som afspejler vigtigheden af ord. En MultinomialNB (Naive Bayes) klassifikator bliver derefter trænet på disse data. Modellen evalueres ved hjælp af valideringssættet, og nøjagtighed og F1-score af modellen rapporteres.

Task 1: Naive Bayes klassifikator udgør en simpel og effektiv baseline model. Den er valgt grundet dens evne til at håndtere tekstdata, dens skalerbarhed og robuste egenskaber overfor manglende data i tekst. Modellen kræver minimal parametertuning, hvilket gør modellen til et godt udgangspunkt.

Task 2: I task 2 overvejer vi på hvilke meta-features der kunne give mening at inddrage. Først prøvede vi at inkludere meta-feature "domain", dette resulterede i en validation accuracy på 1.0, dette tog vi som et tegn på overfitting. Istedet har vi valgt at inkludere meta-feature "*authors*" i modellen. Vi tænkte det potentielt kunne give yderligere signaler, der hjælper med at differentiere mellem fake og reliable news. Resultatet viste sig at give en forøgelse i nøjagtigheden på 9,3%, hvilket vi derfor antog som værende en rigtig god meta-feature at inkludere. Dette fik os også til at tænke, at det sikkert ville være en god ide at bruge denne meta-feature til part 3 hvor vi skulle skabe den bedst mulige advanced model.

Task 3: I denne opgave bøvlede vi en smule med at den nye data ikke have en 'type' kolonne, men vi valgte at antage at de alle var reliable i og med de blev scraped fra bbc.com. Vi oprettede derfor en 'type' kolonne hvor vi satte alle binære værdier til '1', som står for reliable i vores train_data, som den skulle sammensættes med. Som vi kan se på resultatet er nøjagtigheden blevet øget med ca. 0,3% hvilket giver god mening efter som vi har tilføjet 5767 artikler til train_data som i forvejen bestod af 796000 artikler. Da det kun er 5767 artikler vi har tilføjet til train_data havde vi også forventet en meget lille forbedring i nøjagtigheden. Vi har altså kun øget træningssættet med 0,72% og dermed skulle det ikke forbedre nøjagtigheden vildt meget. Ud fra dette resultat, har vi besluttet os for ikke at arbejde videre med dette datasæt, da forøgelsen af nøjagtighed er så lille. Men ud fra resultatet, kan man konkluderer at mere data at træne modellen med er lig med bedre nøjagtighed.

3 Part 3: Advanced Model

Denne del af opgaven går ud på at lave en advanced model altså den bedste model vi kan komme på med vores givne data og viden. Logistisk regression var vores første udkast til baseline model, men da vi testede forskellige modeller til

baseline modellen, kom vi frem til at logistisk regression var den model vi fik bedst resultater med. Ligesom i baseline modellen udnytter vi TfidfVectorizer til at transformere tekstindholdet til numeriske repræsentationer.

Logistisk regression er en statistisk metode, der bruges til at forudsige sandsynligheden for, at en bestemt begivenhed vil forekomme. Formålet er at finde ud af, hvordan en eller flere uafhængige variabler påvirker sandsynligheden for den ønskede begivenhed. I vores datasæt er den begivenhed, som vi ønsker at forudsige sandsynligheden for, om en nyhed er "reliable" (1) eller "fake" (0). Vi bruger uafhængige variabler: "stemmed-content" og "authors" fra vores rensede datasæt. De blev valgt til, under vores test med logistisk regression, her prøvede vi først at andvende meta data: "domain", hvor vi fik en validation accuracy på 1.0 altså 100 procent. Da vi tidligere i opgaven kom frem til at hvert domain kun udgiver 1 type artikler, viste dette et tegn på overfitting. I denne kontekst forventede vi at modellen lærte at genkende hvilket domain der udgav en specifik type artikel og på den måde forudsagde typen uden rigtigt at lære noget konkret. Som i baseline modellen havde vi mere held med at inkludere "authors" som meta data, denne inklusion øgede vores validation accuracy uden at vise tegn på overfitting.

I anvendelsen af logistisk regression justeres en logistisk sigmoid kurve til vores data, hvor x-aksen repræsenterer værdierne af de uafhængige variabler, og y-aksen repræsenterer sandsynligheden for, at en nyhed er pålidelig (1). Denne sigmoid-kurve skifter fra en lav sandsynlighed for falske nyheder til en høj sandsynlighed for pålidelige nyheder baseret på de uafhængige variablers værdier. Under modellens træning tilpasses koefficienterne for hver uafhængig variabel således, at kurven bedst afspejler dataen, hvilket lærer modellen at genkende mønstre, der indikerer, om en nyhed er "reliable" eller "fake".

Efter modellen er trænet, anvender vi den til at estimere sandsynligheden for, at andre nyheder fra vores validation sæt er "reliable". Denne proces indebærer at beregne en sandsynlighed baseret på de uafhængige variablers værdier i den trænede model. For at afgøre nyhedens pålidelighed benyttes der i logistisk regression en beslutningsgrænse: Hvis sandsynligheden for den positive klasse "reliable" (1) er større end eller lig med 0,5, klassificeres observationen som "reliable" (1). Hvis sandsynligheden er mindre end 0,5, klassificeres observationen som "Fake" (0).

Vores resultater af den logistiske regression efter at have trænet den og testet den på valideringsættet giver os en Validation accuracy: 0.9423316582914573. Dette viser modellens evne til korrekt at klassicifere artiklerne fra validation sættet som "reliable" eller "fake". Da vi kørte modellen uden "authors" fik vi ca 0.92 som afspejler den troværdige forbedring nævnt tidligere. Dette resultat går hånd i hånd med vores resultat af en lav Mean Squared Error (MSE) på 0.0433, dette indikerer, at de forudsagte sandsynligheder for klassificeringerne er meget tæt på de faktiske værdier. F1-scoren på 0.9284 viser, at vores model er rigtig god til både at identificere korrekte oplysninger og undgå fejl. Disse resultater er selvfølgelig baseret på hvordan modellen forudsiger på valideringsættet efter at have trænet på træningssættet. Strukturen og mønstrene i disse 2 sæt vil altså være meget ens, så mens modellen viser stærke resultater på valideringssættet, er det ikke til at konkludere hvordan det vil virke på andre sæt. Dette er en vigtig pointe at holde forøje som vi vil komme ind på senere i rapporten.

Egenskab	**Beskrivelse**
Effektivitet	På trods af sin enkelhed kan logistisk regression være meget effektiv til at løse binære klassifikationsproblemer, hvilket passer til vores behov for at klassificere nyheder som troværdige eller utroværdige.
Robusthed	Logistisk regression er relativt robust over for overfitting, hvilket er vigtigt, når vi arbejder med komplekse datasæt. Dette gør det til en pålidelig model selv i tilfælde af store datasæt med mange træk.
Skalering	Det er nemt at skalere logistisk regression til store datasæt og store antal funktioner, hvilket gør den velegnet til vores behov for at behandle store mængder nyhedsdata.

Generelt er vi tilfredse med resultaterne af denne model nogle af de styrker vi mener har været vigtige ved logistisk regression iforhold til vores opgave kan findes i ovenstående tabel. Som altid understreger det vigtigheden i at vælge den korrekte model til den specifikke situation og træne modellen til situationen med hensigtsmæssig data.

4 Part 4: Evaluation

Vi foretog en systematisk evaluering af vores advanced- og baseline-model ved anvendelse på separate testdatasæt med det formål at bedømme deres evne til præcist at skelne mellem falske og pålidelige artikler. Denne tilgang tillod os at vurdere modellernes generelle generaliserbarhed og effektivitet på ukendte datasæt, hvilket er afgørende for at opnå pålidelig ydeevne i praksis. Resultaterne af disse evalueringer, som præsenteret i nedenstående tabel, demonstrerer en imponerende præstationsgrad for begge modeller, hvilket indikerer en stærk kapacitet til at arbejde med nye og ukendte datasæt.

FakeNewsCorpus resultater på test sæt:

Dataset	Model	Test accuracy	F1-score
FakeNewsCorpus	Advanced	0.940593	0.92613
FakeNewsCorpus	Baseline	0.829386	0.897483
LIAR	Advanced	0.624309	0.048
LIAR	Baseline	0.631413	0.158559

FakeNewsCorpus forbedring ved brug af advanced model:

Dataset	Model	Test accuracy	F1-score
FakeNewsCorpus	Advanced	0.111207	0.028647
LIAR	Baseline	-0.007104	-0.11056

Figur 1: Tabel over resultater

I analysen af 'FakeNewsCorpus' data opnåede den avancerede model en testnøjagtighed på 94,05% og en F1-score på 0,926, hvilket afspejler en høj grad af præcision og pålidelighed i klassificeringen. Tilsvarende resultater for baseline-modellen, med en nøjagtighed på 82,93% og en F1-score på 0,898, understreger en stærk baseline ydeevne. Disse resultater blev opnået ved hjælp af en række forudbehandlings- og klassificeringsteknikker som implementeret i den underliggende kode. Funktionerne $read\_csv$ fra $pandas$-biblioteket blev brugt til at indlæse data, mens $fillna$ håndterede manglende værdier, hvilket er kritisk for at opretholde datasættets integritet. $apply$-funktionen blev brugt sammen med en brugerdefineret funktion map_to_binary for at omdanne kategoriske labels til en binær form, hvilket er nødvendigt for binær klassifikation.

Evalueringen på LIAR-datasættet viste en nedgang i præstationen for begge modeller, hvilket illustrerer de unikke udfordringer ved forskellige datasæt. Baseline-modellen overgik den avancerede model på LIAR, hvilket indikerer dens styrke i at modstå overfitting gennem dens simplificerede modelarkitektur. Dette kan skyldes en mere generaliseret tilgang i baseline-modellen, hvilket muliggør bedre performance på et datasæt, der har mindre kompleksitet og variation end 'FakeNewsCorpus'.

Yderligere analyse blev udført ved brug af $accuracy\_score$ og $F1\_score$ fra $scikitlearn$, som er standardmålinger til evaluering af klassificeringsmodeller. Disse metrikker er især anvendelige til at vurdere, hvor godt en model præsterer over for en balanceret datasetdistribuering, ved at kombinere præcision og recall i en enkelt måling. Desuden blev $predict\_proba$ og $mean\_squared\_error$ anvendt for at estimere sandsynlighederne af klassetilhørsforhold og kvantificere fejlmarginerne, som giver yderligere indsigt i modellens prædiktive kapabiliteter.

Konklusionen på denne evaluering fremhæver vigtigheden af ikke blot at udvikle avancerede prædiktive modeller, men også at forstå, hvordan og hvorfor forskellige modeller præsterer som de gør på tværs af diverse datasæt. Dette giver os

en holistisk forståelse af modellernes robusthed og deres anvendelighed under forskelligartede omstændigheder, hvilket er afgørende for den videre udvikling af effektive maskinlæringsløsninger.

5 Conclusion

I løbet af udviklingen af dette projekt har vi udforsket og analyseret en stor mændge nyheds data. Det bestod bla af grundig databehandling, udvikling af en simpel og en avanceret model og en evaluering af disse modellers effektivitet. Målet i dette projekt har været at kunne udnytte maskinlærings teknikker til at identificere "fake" news. Gennem denne analyse har vi demonstreret hvordan Logistisk regression suppleret med meta-feature: "authors", leverer stærke resultater af testnøjagtighed på vores givne data sæt "FakeNewsCorpus". Selvom vores nøjagtighed for denne avanceret model er betydelig, synes vi det er vigtigt at anerkende nogle begrænsninger i vores tilgang. Vores model er trænet på en variation af samme datasæt som det forsøger at forudsige på, derfor kan dens generaliseringsevne overfor ukendte datasæt være tvivlsom. Vi ser jo f.eks. også når vi evaluere modellen på "LIAR" datasættet at vi får en markant lavere testnøjagtighed, dette skyldes bla at strukturen af "LIAR" datasættet er helt anderledes. LIAR datasættet bruger et statement som værende det tætteste vi kan komme på content kollonen, derfor giver det god mening at vi får en lavere nøjagtigheden når strukturen modellen har trænet på er helt anderledes. På baggrund af disse resultater fik vi en indsigt i det essentielle ved at have en god balance mellem avancerede modeller og høj kvalitet af data som begge spiller en stor rolle. Det fremstår, at mens avancerede modeller tilbyder komplekse værktøjer til mønsteridentifikation og klassifikation, så er deres effektivitet strengt afhængig af dataens kvalitet. Igennenm uddannelsen vil vi lære at udvikle et neuralt netværk, som havde været et logisk skridt frem, for at skabe forbedring i denne opgave. Ved at udvide vores forskning til at inkludere disse mere avancerede teknikker, kan vi muligvis udvikle endnu mere præcise og pålidelige værktøjer til at bekæmpe spredningen af misinformation.