
positive
negative

Analiza sentimentelor

Membrii echipei:

- Buhuși Cătălin-Constantin
- Iscu Dan-Adrian
- Lungu Alexandra-Maria
- Roșu Vlad-Gabriel
- Viziteu Paula

Repozitoriu GitHub: https://github.com/Alexandra-MariaL/PSDT_Project

1. Scurtă descriere a proiectului

Recenziile clienților hotelieri

Unui număr tot mai mare de călători le place să-și împărtășească experiența și sentimentele despre sejururile la hotel prin intermediul rețelelor sociale, generând un volum mare de recenzii online la hoteluri. Comentariile generate de utilizatori conțin preferințele acestora pentru diferite aspecte ale hotelurilor, care sunt utile pentru hotelieri pentru a îmbunătăți serviciile hotelurilor. Găsirea unui hotel potrivit în funcție de nevoile și accesibilitatea calatorului este un proces complex de luare a deciziilor. În prezent, disponibilitatea unui număr amplu de recenzii online făcute de clienți ne ajută în acest sens. Recenziile pot prezenta diferite sentimente ale clienților față de un hotel și fiecare recenzie poate fi clasificată în funcție de diferite aspecte, cum ar fi curățenia, valoarea, serviciul etc. Dorința este de a identifica sentimentele detaliate față de atributele hotelului. O metodă bazată pe valorile generale de sentiment ale atributelor hotelului este utilizată pentru a măsura preferințele clienților pentru a sprijini analiza serviciilor hoteliere.

De ce sunt utile recenziile?

Recenziile sunt utilizate în mod obișnuit ca un etalon al satisfacției clienților și ca o sursă de informații despre caracteristicile și serviciile care trebuie îmbunătățite pentru un furnizor de servicii. Recenziile de produs sunt cel mai bun mod prin care să crești vânzările în mod organic, fără vreo investiție (sau cu investiții minime) în reclame. Un produs cu recenzii proaste va avea vânzări mai mici, în timp ce un produs fără niciun review poate semnala faptul ca nimeni nu a fost interesat de el până în prezent. Și cine ar vrea sa fie primul care sa încerce ceva netestat? Nu prea multa lume. Conform unui studiu de caz, 91% dintre persoane citesc în mod regulat sau ocazional review-uri (feedback, opinii, recenzii, testimoniale, păreri), iar 84% dintre clienți au încredere în ele la fel de mult ca într-o recomandare personală din partea unui prieten sau a unei rude.

Clienții vorbesc de obicei despre produse pe rețelele sociale și pe forumurile de feedback ale clienților. Aceste date pot fi colectate și analizate pentru a evalua răspunsul general al clientului. Făcând acest lucru un pas mai departe, tendințele datelor pot fi, de asemenea, examinate. De exemplu, clienții dintr-un anumit grup de vârstă și demografice pot răspunde mai favorabil unui anumit produs decât altora. Pe baza informațiilor colectate, companiile își pot poziționa apoi produsul diferit sau își pot schimba publicul țintă.

Analiza sentimentelor

Diversitatea opiniilor prezente în datele textuale, furnizate de utilizatori, au o complexitate nedorită, deoarece prelucrarea unor astfel de date uriașe este o sarcină aproape imposibilă pentru oameni. În acest scop, informaticienii furnizează unele instrumente/algoritmi de extragere a datelor care pot ajuta utilizatorul să extragă informații relevante din cantitatea mare de date.

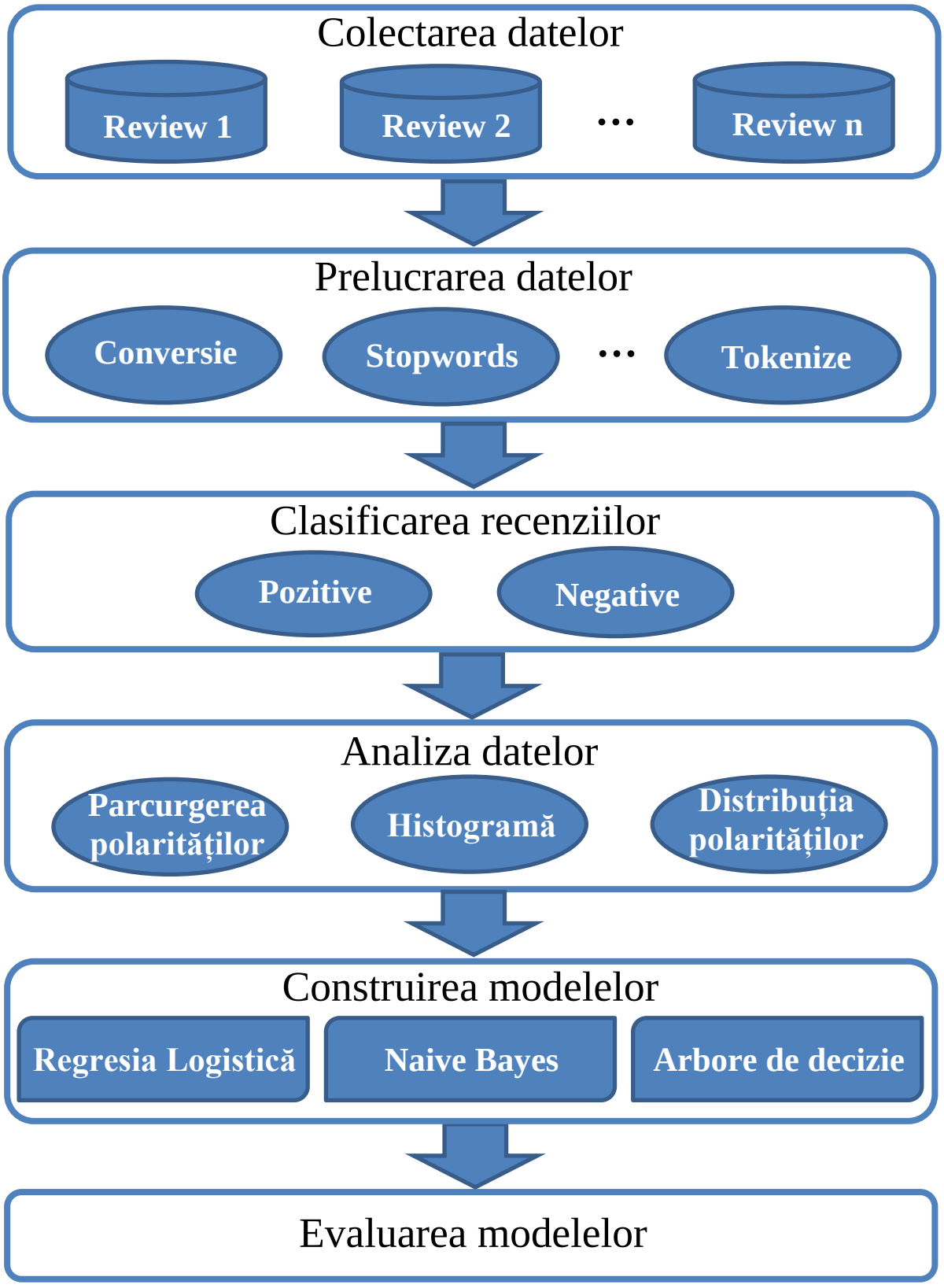
Analiza sentimentelor este procedura de colectare și analiză a opiniilor, sentimentelor, atitudinilor, emoțiilor, evaluărilor și aprecierilor despre produse, servicii, organizații, persoane fizice, evenimente, probleme, subiecte și atributele acestora, care sunt prezente în mod obișnuit în postările de blog, comentarii, recenzii sau rețelele sociale.

Analiza sentimentelor este o arie a procesării limbajului natural (NLP), lingvistică computațională, care urmărește să determine emoțiile, personalitatea etc. ale unui scriitor (recenzor) al unor subiecte specifice. În ultimii ani, mulți cercetători au propus diverse modele de analiză a sentimentelor pe diverse subiecte de turism, finanțe, social media etc.

Datorită creșterii recente a numărului de comunități online, metodele de analiză a recenziilor online scrise în principal de consumatori pentru a-și exprima opiniile, precum și recenziile asupra unui anumit produs au fascinat mulți cercetători. Multe modele recente au fost folosite de diverși cercetători pentru analiza unor astfel de recenzii. Rezervarea unui hotel online a devenit recent o sarcină plictisitoare, cu mii de hoteluri din care să alegeți, pentru fiecare destinație. Sunt probleme care apar destul de des. De exemplu, din reclamă oamenii pot fi convinși cu ușurință, dar s-ar putea să nu fie conștienți de un costurile ascunse pe care agențiile de publicitate nu le dezvăluie. Dar, pentru ei, clienții care au petrecut acolo, feedback-ul lor poate fi considerat de o importanță foarte valoroasă. Motivați de importanța acestor situații, mulți cercetători au încercat să conceapă sisteme care să rezolve această sarcină și să ușureze procesul persoanelor aflate în nevoie. Procesul de proiectare a unor astfel de sisteme bazate pe recenzii a fost elaborat de diverși cercetători cu ajutorul a numeroase tehnici. O astfel de tehnică celebră este prin analiza sentimentelor recenziilor online extrase făcute de diverși consumatori pe site-urile web online.

Proiectul își propune realizarea unui sistem de analiză big-data, pentru determinarea și clasificarea sentimentelor din review-uri clienților. Această analiză se va realiza pe baza unor date de intrare reale, iar rezultatele obținute reflectă sentimente din lumea actuală.

2. Arhitectura



3. Motivații

Deoarece oamenii își exprimă gândurile și sentimentele mai deschis ca niciodată, analiza sentimentelor devine rapid un instrument esențial pentru monitorizarea și înțelegerea sentimentului în toate tipurile de date.

Analiza automată a feedback-ului clienților, cum ar fi opiniile din răspunsurile la sondaje și conversațiile de pe rețelele de socializare, permite mărcilor să afle ce îi face pe clienți fericiți sau frustrați, astfel încât să poată adapta produsele și serviciile pentru a satisface nevoile clienților lor. De exemplu, utilizarea analizei sentimentului pentru a analiza automat peste 2.000 de răspunsuri deschise în sondajele privind satisfacția clienților ar putea ajuta să descoperim de ce clienții sunt fericiți sau nemulțumiți de cazarea dispusa de către Hotelul, subiectul din aceasta analiza.

- Se urmărește sentimentul mărcii, astfel încât să se poată detecta imediat clienții nemulțumiți și să se răspundă cât mai curând posibil.
- Se compară sentimentul pentru o parte din feedback-urile pentru o anumita gama pentru a vedea dacă trebuie luate anumite masuri de calitate PR., Administrative etc.
- Se pot analiza in detaliu anumite categorii în datele calitative pentru a vedea de ce sentimentul este în scădere sau în creștere.

Analiza sentimentelor poate identifica probleme critice în timp real, de exemplu, criza de PR pe social media escaladează etc. Modelele de analiză a sentimentelor pot ajuta să identificam imediat aceste tipuri de situații.

Prin utilizarea unui sistem centralizat de analiză a sentimentelor, companiile pot aplica aceleași criterii tuturor datelor lor, ajutându-le să îmbunătățească acuratețea și să obțină informații mai bune. Așadar, pentru a ajuta să înțelegem modul în care analiza sentimentală ar putea aduce beneficii afacerilor.

4. Provocări în analiza sentimentală

Când vine vorba de provocările de analiză a sentimentelor, există destul de multe lucruri cu care companiile se luptă pentru a obține acuratețea analizei sentimentale. Analiza sentimentelor sau a emoțiilor poate fi dificilă în procesarea limbajului natural doar pentru că mașinile de învățare trebuie să fie instruite să analizeze și să înțeleagă emoțiile așa cum o face un creier uman. Acest lucru este în plus față de înțelegerea nuanțelor diferitelor limbi, iar pe măsură ce știința datelor continuă să evolueze, software-ul de analiză a sentimentelor este capabil să abordeze mai bine aceste probleme.

Câteva din principalele obstacole în analiza sentimentului:

1. Problematica tonalității

Tonul poate fi dificil de interpretat verbal și chiar mai dificil de deslușit în cuvântul scris. Lucrurile devin și mai complicate atunci când cineva încearcă să analizeze un volum masiv de date care pot conține atât răspunsuri subiective, cât și obiective. Companiile se pot confrunta cu dificultăți în găsirea sentimentelor subiective și analizarea corectă a acestora pentru tonul dorit.

Soluție: Baza oricărui software de analiză a sentimentelor bune include capacitatea de a descifra declarații subiective din cele obiective și apoi de a găsi tonul potrivit în el.
De exemplu: "Produsul este superb, dar nu la acest preț" este un sentiment subiectiv, dar cu o tonalitate care spune că prețul face produsul mai puțin atractiv. Cu un API inteligent sentiment, companiile pot descifra astfel de nuanțe în ton, la scară largă.

2. Problematica Polarității

Cuvinte precum "dragoste" și "ură" sunt mari pe scoruri pozitive (+1) și negative (-1) în polaritate. Acestea sunt ușor de înțeles. Dar există conjugări între cuvinte, cum ar fi "nu atât de rău", care poate însemna "medie" și, prin urmare, se află în mijlocul polarității (-75). Uneori, fraze ca acestea sunt lăsate afară, ceea ce diluează scorul sentimental.

Soluție: Instrumentele de analiză a sentimentelor pot găsi cu ușurință aceste fraze și cuvinte polare medii pentru a oferi o viziune holistică asupra unui comentariu. În acest context, o analiză a sentimentelor bazată pe subiect poate oferi o analiză bine rotunjită, dar cu o analiză a sentimentului bazată pe aspect, se poate obține o imagine aprofundată a multor aspecte într-un comentariu.

3. Problematica Ironiei

Oamenii folosesc ironia și sarcasmul în conversații ocazionale și meme-uri pe rețelele de socializare. Actul de exprimare a sentimentului negativ folosind complimente back-handed poate face dificilă pentru instrumentele de analiză a sentimentului pentru a detecta contextul adevărat a ceea ce implică de fapt răspunsul. Acest lucru poate duce adesea la un volum mai mare de feedback "pozitiv", care este de fapt negativ.

Soluție: Un API de analiză a sentimentelor de nivel superior va fi capabil să detecteze contextul limbajului utilizat și orice altceva implicat în crearea sentimentului real atunci când o persoană postează ceva. Pentru aceasta, setul de date lingvistice pe care a fost instruit modelul de analiză a sentimentelor trebuie să fie nu numai precis, ci și masiv.

4. **Problema cu Emoji-urile**

Problema cu conținutul social media care este bazat pe text, cum ar fi Twitter, este că acestea sunt saturate cu emoji- uri. Sarcinile NLP sunt instruite să fie specifice limbajului. Deși pot extrage text chiar și din imagini, emoji-urile sunt un limbaj în sine. Majoritatea soluțiilor de analiză a emoțiilor tratează emoji-urile ca personaje speciale care sunt eliminate din date în timpul procesului de extragere a textului. Dar acest lucru înseamnă că acele companii nu vor primi informații holistice din date.

Soluție: Pentru a face față provocărilor de analiză a sentimentelor de acest fel, o companie trebuie să utilizeze un instrument de analiză a emoțiilor care să poată decoda limba în emoji-uri și să nu le asiste cu personaje speciale, cum ar fi virgule, spații sau opriri complete. Aceasta în sine este o aplicație foarte avansată în care modele precum Repustate sunt instruite special pentru aceasta. Oamenii de știință analizează mai întâi dacă oamenii folosesc emoji-uri mai frecvent în evenimente pozitive sau negative și apoi antrenează modelele pentru a învăța corelația dintre cuvinte și diferite emoji-uri.

5. **Problema de idiom**

Programele de învățare automată nu înțeleg neapărat o figură de stil. De exemplu, un idiom de genul "nu este genul meu" va încurca algoritmul pentru că înțelege lucrurile în sensul literal. Prin urmare, atunci când un idiom este folosit într-un comentariu sau într-o recenzie, propoziția poate fi interpretată greșit de algoritm sau chiar ignorată. Pentru a depăși această problemă, o platformă de analiză a sentimentelor trebuie să fie instruită în înțelegerea expresiilor. Când vine vorba de mai multe limbi, această problemă devine multiplă.

Soluție: Singurul mod în care această provocare poate fi întâmpinată cu acuratețea analizei sentimentale este dacă rețelele neuronale dintr-un API de extragere a emoțiilor sunt instruite să înțeleagă și să interpreteze expresiile. Expresiile sunt cartografiate în funcție de substantive care denotă emoții precum furie, bucurie, determinare, succes etc. și apoi modelele sunt antrenate în consecință. Este suficient să spunem că numai atunci poate un instrument de analiză a sentimentului să ofere perspective exacte dintr-un astfel de text.

6. **Problema negațiilor**

Negații, date de cuvinte cum ar fi nu, niciodată, nu pot, nu au fost, etc. pot confunda modelul ML. De exemplu, un algoritm de mașină trebuie să înțeleagă că o frază care spune: "Nu pot merge la reuniunea mea de clasă", înseamnă că persoana intenționează să meargă la reuniunea clasei.

Soluție: O platformă de analiză a sentimentelor trebuie să fie instruită pentru a înțelege că dublul negativ se depășește reciproc și pentru a transforma o propoziție într-un pozitiv. Acest lucru se poate face numai atunci când există suficient corpus pentru a instrui algoritmul și are numărul maxim de cuvinte de negație posibil pentru a face numărul optim de permutări și combinații.

7. **Problema propozițiilor comparative**

Propozițiile comparative pot fi dificile, deoarece este posibil să nu facă referință întotdeauna la o opinie. O mare parte din ea trebuie dedusă. De exemplu, atunci când cineva scrie, "Galaxy S20 este mai mare decât Apple iphone12", propoziția nu menționează nicio emoție negativă sau pozitivă, ci mai degrabă afirmă o comandă relativă în ceea ce privește dimensiunea celor două telefoane.

Soluție: Acuratețea analizei sentimentale poate fi obținută în acest caz atunci când un model sentimental poate compara măsura în care o entitate are o proprietate într-o măsură mai mare sau mai mică decât o altă proprietate. Aceasta nu este o problemă de a avea pur și simplu un corpus de cuvinte negative sau pozitive specifice sentimentului, ci în formarea mașinii de inteligență artificială pentru a aduna de fapt informații din graficul său de cunoștințe și pentru a analiza relația dintre entități, cuvinte și emoții.

8. **Problema multilingvă**

Analiza multilingvă a sentimentelor constituie toate problemele enumerate mai sus se complică atunci când se aruncă un amestec de limbi. Fiecare limbă are nevoie de un etichetator, lematizator și constructe gramaticale unice pentru a înțelege negațiile. Deoarece fiecare limbă este unică, nu poate fi tradusă într-o limbă de bază, cum ar fi engleza, pentru a extrage informații. Un exemplu simplu fiind, dacă expresia „precum un pește se duce la apă” este tradus în germană, expresia și-ar fi pierdut sensul.

Soluție: Singurul mod în care aceste provocări de analiză a sentimentelor pentru datele multilingve pot fi depășite este calea cea mai grea. Acest lucru înseamnă că modelul de analiză a sentimentelor trebuie să aibă o platformă cu pregătire unică și un model de recunoaștere a entităților numit pentru fiecare limbă pe care o are Repustate. Nu există nicio scurtătură în acest sens, deoarece modelul trebuie să fie instruit manual în fiecare limbă de către oamenii de știință de date. Dar rezultatele merită pentru că vă va oferi cele mai mari scoruri de precizie a analizei sentimentale.

5. State of the art

Irum Hafeez Sodhar et al, Sentiment analysis of Romanized Sindhi text, Feb. 2020

Metodologia de cercetare

Metodologia acestei cercetări este împărțită în cinci pași care sunt: Input text, Text in sentence form, Process the text on machine learning tool, Sentiment analysis and evaluation of results depinde de acești pași pe analiza sentimentelor prezentată în Fig. 2.

Fig. 2. Sentiment analysis procedure.

Date text

În această cercetare cuvintele din textul romanizat sindhi au fost folosite pentru analiza de sentimente, sentimentul textului este pozitiv, neutru sau negativ. Setul de date este componenta de bază a acestei cercetări. Textul folosit în această cercetare este sub formă de propoziție (Subiect +Verb + Determinanți).

Analiza textului

În această lucrare de cercetare au fost folosite instrumente Python pentru Analiza textului sindhi romanizat. Câteva dintre sarcinile importante care sunt: Stemming, Analiza sentimentelor, Etichetarea părților vorbirii, Recunoașterea și tokenizarea entității numite. Analiza textului include și evaluarea opiniilor utilizatorilor din diferite perspective.

Table 1
Data set of Romanized Sindhi text and their sentiment analysis

Data Set of (English + Romanized Text + Sindhi Text) and Results of Sentiment Analysis				
S. No.	English Sentence	Romanized Sindhi Text	Sindhi Text	Sentiment Analysis
1	You are doctor	Tou ahen doctor	تون آهي ڊاڪٽر	Neutral
2	I am student	Ma ahyan shagrid	مان آهيان شاگرد	Neutral
3	I play game	Ma khedan rand	مان کيڈان راند	Neutral
4	I drink water	Ma piyan pani	مان پيان پاڻي	Neutral
5	Bed is very dirty	Dadho kharab ahy bad	ڏاڍو خراب آهي بيڊ	Negative
6	Doctor call the patient	Doctor sadyo mariz khe	ڊاڪٽر سڏيو مريض کي	Neutral
7	I write letter	Ma likhyo khat	مان لکيو خط	Neutral
8	I read book	Ma parhyo kitab	مان پڑهيو ڪتاب	Neutral
9	You fire the home	Tou sadyo ghar	تون ساڙيو گھر	Neutral

Rezultate experimentale și discuții

Analiza sentimentelor a fost efectuată pe setul de date de o sută de propoziții. Toate propozițiile au fost aceleași structura: subiect, verb și obiect. Rezultatele experimentale au fost obținute din instrumentul Python prin utilizarea clasificării de analiză a sentimentelor, așa cum se arată în Tabelul 1. Acest tabelul conține trei propoziții diferite care sunt în engleză, romană și sindhi cu rezultate ale sentimentelor clasificate.

Textul trebuie clasificat în trei atribute ale sentimentelor (pozitiv, neutru și Negativ). Dacă textul este pozitiv și negativ, atunci conține probabilitatea 1, atunci când textul este neutru înseamnă că probabilitatea este mai mare de 0,5. Rezultatele din setul de date arată că nouăzeci și șapte de propoziții sunt neutre, iar trei propoziții sunt negative. Nicio propoziție nu este pozitivă.

Fig. 4. Word cloud of sentiment analysis of Romanized Sindhi.

Probleme de clasificare a sentimentelor

Multe dificultăți și probleme au apărut în timpul clasificării de Text/date folosind instrumentul Python.

Issues in sentiment classification by using Python tool

Issues	Sentiment	Description of Issues
Issue 1	Positive	Python tool did not recognize the positive sentences from data set.
Issue 2	Neutral	Used any word for the input result achieved as neutral.
Issue 3	Neutral	When used interrogative sentence as input, shows output as neutral.
Issue 4	Neutral	When used punctuation in sentence as input, shows output as neutral.
Issue 5	Neutral	Used any sentence into tool and the result was neutral.
Issue 6	Negative	English word negative (bad) were used in Romanized Sindhi sentence shows output as negative.
Issue 7	Positive	No any sentence of Romanized Sindhi Text takes as positive result achieved from the tool.
Issue 8	Neutral	In Romanized sentences used any country name take as neutral.
Issue 9	Neutral	English word negative (ignore) were used in Romanized Sindhi sentence and the result came as neutral.
Issue 10	Negative	Without subject in sentence take as negative (ignore me).

Concluzie

Rezultatele textului romanizat Sindhi sugerează că instrumentele de învățare automată au efectuat o clasificare mai bună de text și a îmbunătățit eficiența sentimentului analiză.

6. Scurtă descriere a fiecărui modul din arhitectură

a. Limbajul de programare folosit

Python este un limbaj de programare dinamic, interpretat, multi-paradigmă. Popularitatea în creștere, dar și puterea limbajului de programare Python au dus la adoptarea sa ca limbaj principal de dezvoltare de către programatori specializați și la predarea limbajului în unele medii universitare. În ceea ce privește paradigma de programare, Python poate servi ca limbaj pentru software de tipul object-oriented, dar permite și programarea imperativă, funcțională sau procedurală.

b. Date de intrare

Fișier de tip JSON cu 1745 de înregistrări de tip cheie-valoare, ce reprezintă recenziile unor persoane oferite online hotelurilor.

c. Prelucrarea și analiza datelor

Pentru prelucrarea datelor am făcut următoarele:

- am definit 8 funcții cu rol în curățarea datelor de intrare (tokenizare - despărțirea string-urilor în liste de cuvinte; transformarea tuturor caracterelor mari în caractere mici; eliminarea numerelor din textele recenziilor; eliminarea semnelor de punctuație; eliminarea cuvintelor irelevante cu ajutorul Stopwords; normalizarea recenziilor);

```
# Tokenization (despartim string-urile intr-o lista de cuvinte)
def sentence_tokenize(text):
    return nltk.sent_tokenize(text)

def word_tokenize(text):
    # returneaza lista de cuvinte
    return nltk.word_tokenize(text)

# ne asiguram ca fiecare litera este lowercase
def lower_words(text):
    return text.lower()

# eliminam numerele
def remove_numbers(text):
    # take string input and return a clean text without numbers, use regex to discard the numbers
    output = ''.join(c for c in text if not c.isdigit())
    return output

def remove_punctuation(text):
    return ''.join(c for c in text if c not in punctuation)

# eliminam cuvintele irelevante
def remove_stopwords(sentence):
    # removes all the stop words like "is,the,a, etc."
    stop_words = stopwords.words("english")
    return ' '.join([w for w in nltk.word_tokenize(sentence) if w not in stop_words])
```

```
# normalizare
def lemmatize(text):
    wordnet_lemmatizer = WordNetLemmatizer()
    lemmatized_word = [wordnet_lemmatizer.lemmatize(word) for word in nltk.word_tokenize(text)]
    return " ".join(lemmatized_word)
```

- am convertit datele de intrare din formatul JSON în formatul dataframe din librăria pandas;
- am eliminat coloana *sent_id* din dataframe deoarece nu era utilă;
- am redenumit coloana *text* în *Phrase* pentru a sugera că e vorba de fraza recenziei;
- am adăugat în dataframe o nouă coloană numită *Sentiment,* pe care am utilizat-o mai apoi la clasificarea recenziilor în categoriile pozitive și negative. Întrucât datele de intrare dispuneau de un vector numit *opinions* ce stoca subpropoziții din enunțul fiecărei recenzii și totodată polaritatea fiecărei subpropoziții, dar nu și categoria din care făcea parte recenzia per total, aceasta a trebuit să fie determinată;

```
train_data = pd.read_json("train.json")  # from json to dataframe
train_data.drop("sent_id", axis="columns", inplace=True)  # drop useless column "sent_id"
train_data.rename(columns={"text": "Phrase"}, inplace=True)  # rename column  "text" into "Phrase"
train_data.insert(2, "Sentiment", "", True)  # insert new column "Sentiment"
```

```
# clasificam recenziile in pozitive si negative
# compute new column "Sentiment" based on column "opinions"
review_score = 0
for i in range(len(train_data["Phrase"])):
    if len(train_data["opinions"][i]) > 0:
        for j in range(len(train_data["opinions"][i])):
            if train_data["opinions"][i][j]["Polarity"] == "Positive":
                if train_data["opinions"][i][j]["Intensity"] == "Standard":
                    review_score = review_score + 1
                elif train_data["opinions"][i][j]["Intensity"] == "Strong":
                    review_score = review_score + 2
            elif train_data["opinions"][i][j]["Polarity"] == "Negative":
                if train_data["opinions"][i][j]["Intensity"] == "Standard":
                    review_score = review_score - 1
                elif train_data["opinions"][i][j]["Intensity"] == "Strong":
                    review_score = review_score - 2
        if review_score > 0:
            train_data["Sentiment"][i] = "Positive"
        else:
            train_data["Sentiment"][i] = "Negative"
        review_score = 0
    else:
        train_data["Sentiment"][i] = "None"
```

- am eliminat recenziile care nu aveau opinii (vectorul *opinions* era gol). În urma acestei etape numărul de recenzii s-a redus de la 1745 la 1400;
- la final am eliminat coloana *opinions* deoarece nu va mai fi utilă.

```
# eliminam recenziile fara opinii
train_data.drop(train_data[train_data["Sentiment"] == "None"].index, inplace=True)
train_data.reset_index(drop=True, inplace=True)  # reset the index after rows drop
train_data.drop("opinions", axis="columns", inplace=True)  # drop column "opinions" because no longer needed
```

Pentru analiza datelor am făcut următoarele:

- am apelat funcția *info* asupra dataframe-ului pentru a obține informații de bază despre setul de date;

```
RangeIndex: 1400 entries, 0 to 1399
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Phrase     1400 non-null   object
 1   Sentiment  1400 non-null   object
dtypes: object(2)
```

- am parcurs toate recenziile (fraza și sentimentul fiecărei recenzii);
- am afișat distribuția sentimentelor (numărul de recenzii pozitive și negative) împreună cu procentele acestora;

```
# calculam si afisam procentele sentimentelor pozitive si negative
positive_reviews = []
negative_reviews = []
positive_count = 0
negative_count = 0
total_count = len(train_data)

for i in range(total_count):
    if train_data["Sentiment"][i] == "Positive":
        positive_count += 1
        positive_reviews.append(train_data["Phrase"][i])
    elif train_data["Sentiment"][i] == "Negative":
        negative_count += 1
        negative_reviews.append(train_data["Phrase"][i])

positive_percent = round(positive_count * 100 / total_count, 2)
negative_percent = round(negative_count * 100 / total_count, 2)

print("\nPositive sentiments percent: " + str(positive_percent) + "%")
print("Negative sentiments percent: " + str(negative_percent) + "%")
```

- am afișat în browser o histogramă pe baza sentimentelor recenziilor;

```
# afisam in browser o histograma pe baza sentimentelor
color = sns.color_palette()
fig = px.histogram(train_data, x=train_data["Sentiment"])
fig.update_traces(marker_color="turquoise", marker_line_color='rgb(8,48,107)', marker_line_width=1.5)
fig.update_layout(title_text='Review Sentiment')
fig.show()
```

- am afișat grafice cu recenziile pozitive și negative folosind wordcloud.

d. Metode folosite

Inițial datele de intrare au fost împărțite în 80% date pentru antrenare și 20% date pentru testare.

Regresia logistică

Pentru realizarea regresiei logistice s-au folosit noțiunile *CountVectorizer*, funcțiile *fit* și *predict*, iar la final rezultatele au fost afișate prin intermediul raportului de clasificare (ce conține printre altele precizia clasificării recenziilor în negative și pozitive și acuratețea modelului).

Tot aici a fost determinată și matricea de confuzie ce reprezintă rezumatul performanțelor de clasificare ale algoritmului (numărul de clasificări TP, FP, TN, FN).

```
index = train_data.index

# adaugam o nova coloana (random_number) in df si o populam cu valori aleatorii
train_data["random_number"] = np.random.randn(len(index))

# impartim datele de intrare in 80% date pentru antrenament si 20% date pentru testare
train = train_data[train_data["random_number"] <= 0.8]
test = train_data[train_data["random_number"] > 0.8]

vectorizer = CountVectorizer(token_pattern=r'\b\w+\b')  # vectorizam dataframe-urile test si train
train_matrix = vectorizer.fit_transform(train["Phrase"])
test_matrix = vectorizer.transform(test["Phrase"])

lr = LogisticRegression()  # aplicam regresia logistica
```

```
# definim variabilele de antrenare si testare
X_train = train_matrix  # contine vectori de cuvinte din datele de antrenare
X_test = test_matrix  # contine vectori de cuvinte din datele de testare
y_train = train["Sentiment"]  # contine coloana polarity din datele de antrenare
y_test = test["Sentiment"]  # contine coloana polarity din datele de testare

lr.fit(X_train, y_train)  # antrenam modelul astfel incat sa functioneze pe date de intrare noi (de test)

# facem predictii avand ca intrare date de test (review-uri cu care nu a fost antrenat modelul)
predictions = lr.predict(X_test)
confusion_matrix = confusion_matrix(predictions, y_test)  # afisam matricea de confuzie

print("\nLogistic Regression model classification report:")
print(classification_report(predictions, y_test))  # afisam acuratetea modelului
```

Naive Bayes

Acest model se bazează pe conceptul Pipeline și este singurul din cele trei modele care are rezultat determinist (întotdeauna acuratețea este în valoare de 85%), față de celelalte modele care au rezultate nedeterministe (valori diferite ale acurateței obținute la fiecare nouă execuție a programului).

Cele trei nivele care compun Pipeline-ul sunt bag of words, tfidf (term frequency–inverse document frequency) și clasificatorul Naive Bayes.

```
bow = CountVectorizer(analyzer=preprocess)
classifier = MultinomialNB()
tfidf = TfidfTransformer()

pipeline = Pipeline([
    ("bow", bow),  # bag of words
    ("tfidf", tfidf),  # term frequency-inverse document frequency
    ("classifier", classifier),  # Naive Bayes classifier
])
pipeline.fit(train_data["Phrase"], train_data["Sentiment"])
print("Naive Bayes model accuracy: " + str(round(pipeline.score(train_data["Phrase"], train_data["Sentiment"]), 2)))
```

În urma mai multor execuții s-a observat că modelul determinist produce cele mai bune valori pentru acuratețe (deși cu mici diferențe față de celelalte modele)

Arbori de decizie

Obiectul clasificator bazat pe arbori de decizie a fost creat cu ajutorul funcției *DecisionTreeClassifier*, antrenat cu funcția *fit* căreia i-au fost furnizate date de antrenare. În final au fost realizate predicții prin intermediul funcției *predict* căreia i-au fost furnizate date de test. S-a constatat că de obicei acest model produce valori mai mici ale preciziei față de celelalte modele deși diferențele sunt mici (de doar câteva procente).

```
# cream obiectul de clasificare bazat pe arbori de decizie
clf = DecisionTreeClassifier()

# antrenam clasificatorul arborelui de decizie
clf = clf.fit(X_train, y_train)

# afisam acuratetea
y_pred = clf.predict(X_test)
print("Decision Tree model accuracy: ", round(metrics.accuracy_score(y_test, y_pred), 2))
```

e. Date de ieșire:

- la linia de comandă: acuratețea pentru fiecare din cele 3 modele de clasificare abordate;

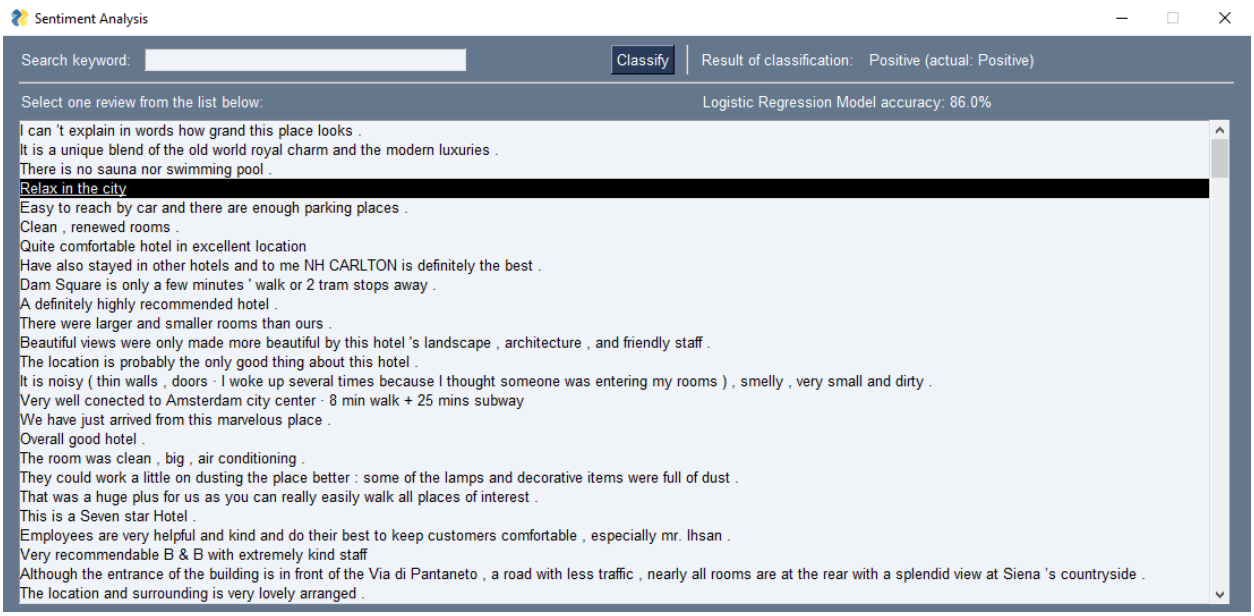
```
Sentiment distribution:
Positive    991
Negative    409
Name: Sentiment, dtype: int64

Logistic Regression model classification report:
              precision    recall  f1-score   support

    Negative       0.59      0.68      0.63        63
    Positive       0.91      0.86      0.88       221

    accuracy                           0.82       284
   macro avg       0.75      0.77      0.76       284
weighted avg       0.84      0.82      0.83       284

Pipeline model accuracy: 0.85
Decision Tree model accuracy:  0.81
```

- pe interfață: categoria în care un model clasifică o recenzie dată de utilizator.

Pentru realizarea interfeței s-a folosit framework-ul *PySimpleGUI*.

Interfața aplicației este reprezentată de o fereastră principală pe care se află mai multe elemente grafice, printre care un panou (listbox) ce conține o listă cu recenzii pe care utilizatorul le poate selecta, un buton de clasificare (pentru a clasifica recenzia selectată cu ajutorul cursorului în panou), un mesaj cu soluția clasificării și un label cu acuratețea modelului.

Modelul de regresie logistică folosit de către interfață pentru clasificarea unei recenzii este antrenat înainte de lansarea interfeței.



Fereastra principală este formată dintr-un layout ce conține elementele grafice descrise mai sus.

Pe interfață este executată o buclă infinită prin intermediul căreia pot fi înregistrate evenimentele de interacțiune ale utilizatorului (selecție cu mouse-ul, apăsare de buton, scrierea într-un textbox etc.).

De asemenea, am definit o variabilă globală *g_review*, care stochează textul recenziei selectate din listbox. Astfel, este posibilă realizarea clasificării de către model la apăsarea butonului *Classify*. Rezultatul clasificării este mai apoi scris pe interfață în dreptul etichetei *Result of classification*.

Altă funcționalitate a interfeței este reprezentată de către *searchbox-ul* în care utilizatorul poate căuta în toate recenziile după un cuvânt cheie. Căutarea se face automat, nemaifiind necesară apăsarea altui buton de căutare. Rezultatele căutării vor apărea la consolă pentru a nu aglomera foarte tare interfața în cazul în care există multe recenzii care conțin cuvântul cheie căutat.

7. Descrierea integrării diferitelor module

Probleme prevăzute

Problema prevăzută a fost reprezentată de acuratețea slabă a modelului folosit (mai mică de 70%). De aceea s-a decis implementarea mai multor modele, iar pentru interfață s-a folosit cel mai bun dintre acestea.

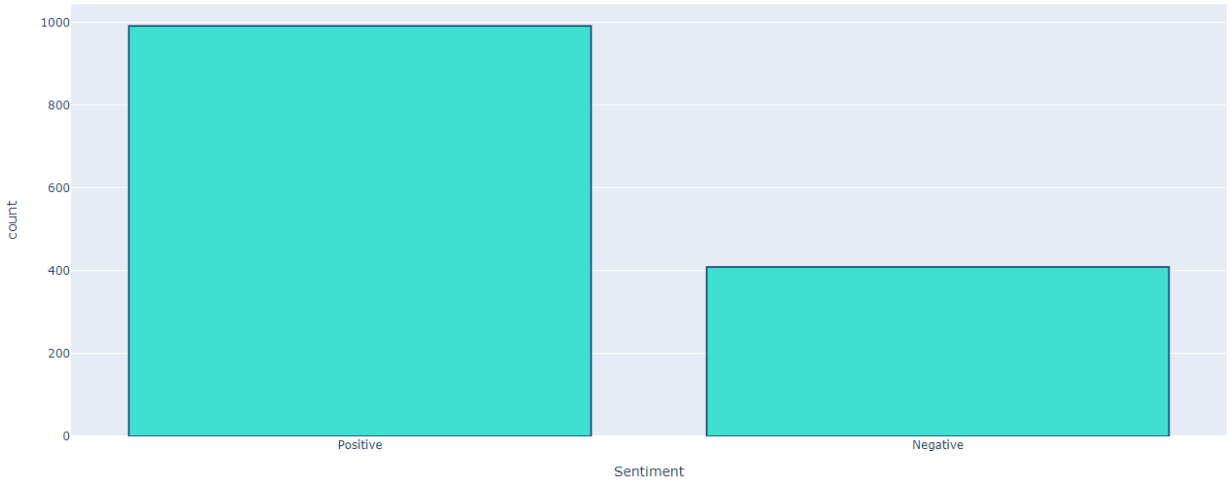
Soluție: submodulul în care este implementat modelul de învățare automată, și anume modelul *Logistical Regression,* a fost integrat în modulul interfeței deoarece permite accesul către variabila ce conține lista cu recenziile de testare care vor fi afișate pe interfață.

La prima vedere, este doar o problemă de clasificare a textului, dar dacă aprofundăm, vom afla că există o mulțime de probleme care afectează serios acuratețea analizei sentimentelor. Acestea ar putea fi:

- ◊ Ironia și sarcasmul;
- ◊ Tipurile de negații;
- ◊ Ambiguitatea cuvântului.

8. Statistici

Histograma sentimentelor



În aceasta histogramă putem observa faptul ca sentimentele pozitive sunt predominante, adică cele mai multe review-uri sunt pozitive. Un lucru bun din perspectiva unei persoane care activează în domeniul hotelier.

Distribuția recenziilor

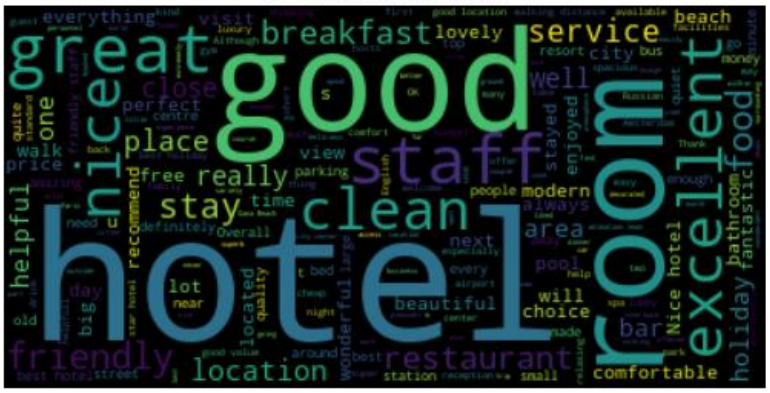
După cum am precizat și văzut în histogramă, majoritatea sentimentele sunt pozitive dar totuși ar trebui să aflăm un număr mai precis. De aceea, am calculat procentele recenziilor pozitive și a celor negative.

```
Sentiment distribution:
Positive    991
Negative    409
Name: Sentiment, dtype: int64

Positive sentiments percent: 70.79%
Negative sentiments percent: 29.21%
```

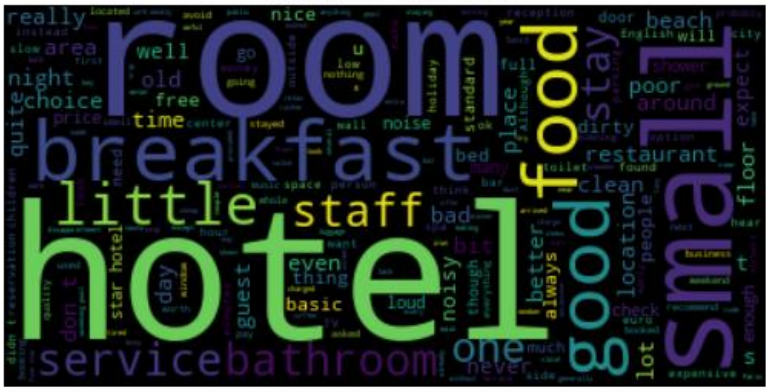
În output putem observa că 70.79% sunt recenzii pozitive și 29.21% sunt recenzii negative.

Wordclouds

positive wordcloud



negative wordcloud



9. Măsurări calitative în evaluarea proiectului

Numărul de adnotări:

- Inițial → 0;
- După prelucrarea datelor de intrare → 1400 (991 recenzii pozitive și 409 recenzii negative).

Numărul de recenzii procesate:

1745 dintre care 345 eliminate datorită lipsei opiniilor (baza de calcul pentru a determina categoria din care face parte o recenzie).

Performanțele modelelor folosite:

- sunt reprezentate de acuratețea acestora (raportul dintre clasificările determinate în mod corect de către fiecare model și numărul total de recenzii clasificate)

```
Sentiment distribution:
Positive    991
Negative    409
Name: Sentiment, dtype: int64

Logistic Regression model classification report:
              precision    recall  f1-score   support

    Negative       0.59      0.68      0.63        63
    Positive       0.91      0.86      0.88       221

    accuracy                           0.82       284
   macro avg       0.75      0.77      0.76       284
weighted avg       0.84      0.82      0.83       284

Pipeline model accuracy: 0.85
Decision Tree model accuracy:  0.81
```