# Przetwarzanie języka naturalnego - dokumentacja końcowa

Temat projektu: Odtwarzanie danych o pogodzie na podstawie tweetów

Maksim Makaranka, Franciszek Sakowski

14 czerwca 2024

## 1 Opis problemu

Badane przez nas zagadnienie polega na przetwarzaniu tekstowych wiadomości na temat pogody. Skupimy się na tak zwanych "tweetach", czyli krótkich sekwencjach słownych publikowanych przez ludzi z całego świata na platformie X.com (dawniej twitter.com). Celem projektu jest zastosowanie technik przetwarzania języka naturalnego oraz uczenia maszynowego do analizu sentymentu - identyfikowania informacji o pogodzie zawartej w wiadomości tekstowej i określenia czy autor ocenia ją pozytywnie czy negatywnie.

## 2 Studia literaturowe

W niniejszym rozdziale zaprezentujemy przegląd prac naukowych, które dotyczą zagadnień związanych z naszym projektem. Skupimy się na analizie istniejących rozwiązań, metod i narzędzi, które wykorzystują tweety jako źródło danych o pogodzie, oraz na ocenie ich zalet i ograniczeń.

- Praca naukowa *Multi-class Weather Forecasting from Twitter Using Machine Learning Approaches*[1], przedstawiona na 5. Międzynarodowej Konferencji Nauk Komputerowych i Inteligencji Obliczeniowej (ICCSCI) w 2020 roku. Jej celem było automatyczne klasyfikowanie pogody na podstawie tweetów za pomocą technik wydobywania tekstu i uczenia maszynowego. Autorzy użyli trzech metod klasyfikacji: Support Vector Machine (SVM), Multinomial Naive Bayes (MNB) i Logistic Regression (LR). Dane treningowe i testowe pochodziły z Twittera, a klasami określanymi przez model były `Sunny`, `Rainy`, `Cloudy`, `Thunderstorm` i `Heavy rain`. Wyniki pokazały, że SVM osiągnęła najlepszą dokładność w porównaniu z MNB i LR. Praca ta dowiodła dobrej aplikacji uczenia maszynowego na danych z Twittera o pogodzie, ale różni się od naszego projektu tym, że model określa stan pogody, a nie nastroj autora w związku z nią.

- W pracy naukowej *Twitter-based classification for integrated source data of weather observations*[2] opublikowanej w "International Journal of Artificial Intelligence" w marcu 2023 roku autorzy analizowali teksty związane z pogodą w Indonezji na podstawie wyszukiwań na Twitterze. W tym przypadku klasami określanymi przez model były `Sunny`, `Rainy`, `Cloudy`, `Light rain` i `Heavy rain`, co wskazuje na podobieństwo tego problemu do problemu z uprzednio wspomnianej pracy. Przetestowano trzy algorytmy uczenia maszynowego: Support Vector Machine (SVM), Multinomial Logistic Regression (MLR) i Multinomial Naive Bayes (MNB), a także wstępnie nauczonych reprezentacji enkodera dwukierunkowego BERT, który został dostrojony (ang. fine-tuning), aby zapewnić skuteczną klasyfikację. Dokładność modelu BERT, obliczona za pomocą F1-score wyniosła 99 procent, co było wyższe niż w przypadku innych metod uczenia maszynowego. Wnioski z pracy wskazują, że SVM jest najlepszym algorytmem klasyfikacji automatycznej w porównaniu z MNB i MLR.

## 3 Szkic rozwiązania

### 3.1 Opis danych

Dane, które użyliśmy do naszego projektu, pochodzą z portalu data.world[3] i zawierają 1000 rekordów opisujących nastrój `sentiment` autorów tweetów w stosunku do pogody. Autor udostępnia dwa datasety:

- Pierwszy dataset zawiera ocenę sentymentu na podstawie tekstu, zebraną od 20 ludzi, wraz ze współczynnikiem pewności `confidence` wskazującym na poziom emocji wyrażonych przez autora tekstu. Możliwe są 4 sentymenty: `Positive, Negative, Neutral/author is just sharing information, Tweet not related to weather condition`. Z tego wynika, że przed nami zadanie klasyfikacji z odpowiednio 4 klasami. Współczynnik `confidence` oznacza, jak bardzo oceniający byli pewni swojej oceny sentymentu. Im wyższy współczynnik, tym większa pewność.

- Drugi dataset zawiera informacje, czy pierwsza ocena sentymentu jest poprawna. W tym zadaniu poproszono 10 ludzi, aby sprawdzili dokładność oryginalnej oceny sentymentu. Każdy z nich mógł odpowiedzieć tak lub nie,

a następnie obliczono procent poprawnych odpowiedzi. Wycena poprawności poprzedniej oceny jest podana w kolumnie `is_the_category_correct_for_this_tweet` i przyjmuje wartości `Yes`, `No` lub `I can't say`.

Dla nauczenia naszego modelu zastosujemy drugi dataset, z którego usuniemy wszystkie rekordy, które mają wartość `No` lub `I can't say` w kolumnie i`s_the_category_correct_for_this_tweet`. W ten sposób otrzymamy bardziej wiarygodne dane do treningu i testowania naszego modelu.

## 3.2 Rozszerzenie danych przy użyciu GPT-4

W trakcie implementacji rozwiązania zobaczyliśmy, że ilość danych nie była wystarczająca, aby wytrenować nasze modele z wysoką dokładnością.

Aby temu zaradzić, postanowiliśmy wygenerować ponad 500 dodatkowych tweetów, używając chat bota o nazwie Copilot, który jest zasilany przez LLM (ang. Large language model) GPT-4. Dostarczyliśmy chat botowi kilka słów kluczowych i przykładowych tweetów, a on wygenerował realistyczne i zróżnicowane tweety o pogodzie z przypisanymi sentymentami i współczynnikami `confidence`.

Korzystając z tych uzupełnionych danych, byliśmy w stanie poprawić wydajność naszych modeli, ale taka decyzja też miała swoje konsekwencje. W notatniku o nazwie **Data analysis.ipynb** można znaleźć szczegółową analizę danych oryginalnych i uzupełnionych, wraz z ich porównaniami i wnioskami co do użycia LLM w generowaniu danych na potrzeby NLP.

## 3.3 Opis rozwiązania

Opracowanie rozwiązania planujemy rozpocząć od analizy zebranych danych celem ich lepszego poznania i zrozumienia. Następnie skupimy się na przetworzeniu posiadanego zbioru danych, to jest na przeprowadzeniu:

- Usunięcia niepotrzebnych z punktu widzenia problemu oceniania sentymentu elementów tweetów, takich jak linki czy oznaczenia innych użytkowników (@mention)
- Unifikacji wielkich i małych liter, zamieniając wszystkie litery w tekście na małe
- Usunięcia liczb i znaków oprócz spacji i słów
- Tokenizacji tekstu, czyli podzielenia tekstów na oddzielne tokeny, ew. słowa
- Usinięcia stopword'ów, czyli wyrazów nieznaczących
- Stemming, czyli usunięcie ze słow końcówek fleksyjnych

Kolejnym krokiem naszego rozwiązania będzie przetestowanie jakości działania dostępnych modeli uczenia maszynowego. Zdecydowaliśmy się na wybór następujących modeli:

- Random Forest - podstawowy algorytm oparty na drzewach decyzyjnych
- SVM - z uwagi na to, jak dobrze ten algorytm radził sobie w przytoczonych w studium literaturowym przykładach
- XGBoost - uważany za bardzo mocny model w klasycznych zadaniach klasyfikacji, jesteśmy ciekawi jak poradzi sobie w kontekście problemu przetwarzania języka naturalnego

Jeśli starczy nam czasu planujemy także wykorzystanie sieci neuronowych w kontekście naszego zadania, a konkretnie algorytmów:

- LSTM,
- BERT.

Podsumowując, celem naszej pracy jest porównanie modeli uczenia maszynowego, których wejściem będzie wiadomość tekstowa (tweet), a wyjściem przypisana klasa określająca sentyment wraz z wartością współczynnika confidence. Dodatkowo chcielibyśmy uwzględnić i przetestować wpływ różnych parametrów na jakość predykcji wspomnianych modeli.

# 4 Implementacja

Do zaimplementowania opisanego rozwiązania wykorzystaliśmy język Python wraz z następującymi bibliotekami:

- Pandas
- Numpy
- Matplotlib
- Seaborn
- nltk
- pickle

Wszelkie parametry dotyczące modelu, procesu trenowania czy też przetwarzania danych przechowywane są w konfiguracyjnym pliku tekstowym.

# 5 Instrukcja obsługi

Struktura plików naszego repozytorium została zorganizowana w następujący sposób:

- Folder **/data** zawiera pliki ze zbiorami danych w formacie csv:
  - **weather-sentiment.csv** zawiera jedynie rekordy z oryginalnego zbioru danych z portalu data.world
  - **weather-sentiment-gpt.csv** rozszerza podstawową wersję zbioru o nowe rekordy wygenerowane za pomocą narzędzia Copilot (GPT-4)
- Folder **/models** służy do przechowywania wytrenowanych algorytmów celem ich późniejszego użycia. W tym folderze umieszczono także inne elementy potrzebne do poprawnego przetwarzania danych (vectorizer oraz label encoder).
- Wszelkie pliki źródłowe zawierające definicje klas umieszczono w folderze **/src**. Składa się on z:
  - Assessor.py - klasa pozwalająca na ocenianie sentymentu fragmentów tekstu wprowadzanych przez użytkownika, zarówno przy pomocy konkretnego wybranego modelu, jak i wszystkich dostępnych
  - TweetPreprocessor.py - klasa modyfikująca zawartość tweetów przed użyciem ich w procesie trenowania, implementuje takie operacje jak zmiana wszystkich liter na małe, usuwanie niepotrzebnych elementów(linki, @mention), tokenizacja, stemming czy usuwanie słów znajdujących się na stop liście
  - WeatherSentimentData.py - klasa reprezentująca zbiór danych
- Config.py zawiera definicje stałych oraz scieżek do danych.
- **Custom input.ipynb** - pozwala użytkownikowi na zdefiniowanie własnego tekstu wejściowego(tweeta), a następnie sprawdzenie jaki sentyment przewidywany jest przez każdy z wytrenowanych przez nas modeli
- **Data analysis.ipynb** - zawiera analizę zbioru danych
- **RF, XGB, SVM comparison - Basic approach.ipynb** - porównuje między sobą 3 algorytmy w podejściu podstawowym
- **RF, XGB, SVM comparison - Complex classes approach.ipynb** - porównuje między sobą 3 algorytmy w podejściu z dodatkowymi klasami
- **RF, XGB, SVM comparison - Confidence weights approach.ipynb** - porównuje między sobą 3 algorytmy, waga tweetów zależna jest od wartości wskaźnika `confidence`
- **RF regressor.ipynb** - zawiera przetrenowanie RF regressor'a dla klasyfikacji na podstawie `confidence` w kontekście zadania. Jedno z pierwszych przetestowanych podejść, nie brane pod uwagę w testach i zostawione jako ciekawostka
- **text-preprocessing.ipynb** - zawiera step-by-step preprocessing tekstu

# 6 Testy

## 6.1 Algorytmy i podejścia

Przeprowadzone przez nas porównanie obejmuje przetestowanie 3 wymienionych w sekcji 3.3 algorytmów - Random Forest, SVM oraz XGBoost. Każdy z nich został wytrenowany w 3 wariantach, różniących się między sobą sposobem przygotowania danych. Przetestowaliśmy następujące podejścia:

- Podejście **Basic** polegające jedynie na usunięciu ze zbioru treningowego wszystkich tweetów o niskim wskaźniku `confidence` $\leqslant$ `threshold`
- Podejście **Complex Classes** polegające jedynie na modyfikacji zbiorów treningowego i testowego i dodaniu 4 nowych klas **Highly Positive**, **Slightly Positive**, **Highly Negative**, **Slightly Negative** zastępujących 2 bazowe klasy **Positive** oraz **Negative**. Zamiana przebiega w następujący sposób:
  - **Highly Positive** - jeśli sentyment to **Positive** i `confidence` $>$ `threshold`
  - **Slightly Positive** - jeśli sentyment to **Positive** i `confidence` $\leqslant$ `threshold`
  - **Highly Negative** - jeśli sentyment to **Negative** i `confidence` $>$ `threshold`
  - **Slightly Negative** - jeśli sentyment to **Negative** i `confidence` $\leqslant$ `threshold`
- Podejście **Confidence Weights** przypisujące tweetom ze zbioru treningowego wagi zgodne z wartością wskaźnika `confidence`.

## 6.2 Metryki

W efekcie wytrenowaliśmy 9 różnych algorytmów uczenia maszynowego i wykorzystaliśmy każdy z nich do policzenia typowych dla zadań klasyfikacji metryk, którymi są:

- Accuracy,
- Precision,
- Recall,
- F1 Score.

Wartości metryk, porównanie zaimplementowanych algorytmów oraz zestawienie ich wyników z wnioskami płynącymi ze studia literaturowego znajdują się w rodziale Wnioski.

# 7 Wnioski

## 7.1 Wartości metryk

Tabela 1: Accuracy na całym zbiorze danych(macro average)

| | | Podejście | | |
|---|---|---|---|---|
| | | *Basic* | *ComplexClasses* | *ConfidenceWeights* |
| Algorytm | Random Forest | 0.836 | 0.544 | 0.836 |
| | SVM | 0.810 | 0.538 | 0.793 |
| | XGBoost | 0.828 | 0.513 | 0.819 |

Tabela 2: Precision na całym zbiorze danych(macro average)

| | | Podejście | | |
|---|---|---|---|---|
| | | *Basic* | *ComplexClasses* | *ConfidenceWeights* |
| Algorytm | Random Forest | 0.85 | 0.60 | 0.86 |
| | SVM | 0.85 | 0.39 | 0.85 |
| | XGBoost | 0.85 | 0.45 | 0.84 |

Tabela 3: Recall na całym zbiorze danych(macro average)

| | | Podejście | | |
|---|---|---|---|---|
| | | *Basic* | *ComplexClasses* | *ConfidenceWeights* |
| Algorytm | Random Forest | 0.83 | 0.49 | 0.83 |
| | SVM | 0.79 | 0.46 | 0.79 |
| | XGBoost | 0.84 | 0.46 | 0.82 |

Tabela 4: F1 score na całym zbiorze danych(macro average)

| | | Podejście | | |
|---|---|---|---|---|
| | | *Basic* | *ComplexClasses* | *ConfidenceWeights* |
| Algorytm | Random Forest | 0.84 | 0.47 | 0.84 |
| | SVM | 0.81 | 0.41 | 0.79 |
| | XGBoost | 0.84 | 0.44 | 0.82 |

### 7.2 Porównanie algorytmów

Uzyskane przez nas wysokości metryki Accuracy wskazują Random Forest jako najlepiej radzący sobie z zadaniem algorytm, i to w każdym z przetestowanych podejść. Nieco gorsze rezultaty w podejściu Basic oraz ConfidenceWeights uzyskał XGBoost, a najsłabiej wypadł SVM. Jedynie w podejściu ComplexClasses zdołał uzyskać lepszy wynik od XGBoosta i zbliżyć się do Random Forest. Rezultaty dla pozostałych metryk prezentują się podobnie. Random Forest uzyskał najlepsze wyniki każdej z metryk. Drugim najlepszym algorytmem w znakomitej większości przypadków był XGBoost, a najsłabiej wypadał SVM.

Porównując podejścia między sobą można dojść do wniosku, że Basic oraz ConfidenceWeights dały porównywalne rezultaty, szczególnie dla algorytmu Random Forest. W przypadku SVM i XGBoost wariant Basic przyniósł nieco lepsze efekty. Podejście ComplexClasses wypadło najgorzej, należy jednak pamiętać, że zawiera dwie klasy więcej, co przekłada się na niższe wartości metryk. Również, jak zauważono podczas implementacji, rozszerzenie datasetu o rekordy wygenerowane okazało najlepszy wpływ na ten właśnie algorytm, zatem z wysoką wiarygodnością można stwierdzić, że na większym zbiorze danych wyniki tego modelu byłyby znacznie lepsze.

Studia literaturowe wskazują SVM jako algorytm nauczania maszynowego dobrze radzący sobie z zadaniem przetwarzania i klasyfikacji tweetów. Uzyskane przez nas wyniki sugerują, że SVM może co prawda dawać zadowalające rezultaty, jednakże algorytmy oparte na drzewach decyzyjnych uzyskują nieco wyższe wartości metryk. Potwierdza to naszą tezę, że algorytmy drzewowe radzą sobie dobrze zarówno w klasycznych zadaniach klasyfikacji, jak i w zadaniach przetwarzania języka naturalnego.

### 7.3 Pozostałe wnioski

Jakość predykcji dla każdego z algorytmów oprócz regressora znacznie wzrosła po wygenerowaniu większej liczby danych. Możliwe, że aktualne rozmiary zbioru danych wciąż nie są wystarczające i można by uzyskać lepsze wyniki mając jeszcze więcej danych.

Algorytmy pozwalały uzyskać najlepsze rezultaty, wykorzystując w procesie wektoryzacji jedynie krótkie n-gramy (unigramy, bigramy i trigramy). Użycie dłuższych n-gramów byłoby niepraktyczne ze względu na wielkość zbioru danych oraz fakt, że tweety są krótkimi, zazwyczaj składającymi się jedynie z kilku słów, formami przekazu.

Korzystanie z dużych modeli językowych, takich jak GPT-4, może pomóc generować więcej danych tekstowych i zwiększyć słownictwo modeli do analizy sentymentu. Jednak, wartości współczynnika `confidence` przypisywane przez bota czatu nie są bardzo wiarygodne do przewidywania sentymentu tweetów. Dlatego należy unikać korzystania z wartości numerycznych generowanych przez duże modele językowe i opisujących cechy generowanych tekstów.

## Literatura

[1] Kartika Purwandari, Join W.C. Sigalingging, Tjeng Wawan Cenggoro, Bens Pardamean, Multi-class Weather Forecasting from Twitter Using Machine Learning Aprroaches, Procedia Computer Science, Volume 179, 2021, Pages 47-54, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.12.006. (https://www.sciencedirect.com/science/article/pii/S187705092032442X) Abstract: Now in the era of big data, many are applying information methods accurately especially by social media. The aims of this study to classify the weather based on Twitter automatically using text mining by using Support Vector Machine (SVM), MultinomialNaive Bayes (MNB), and Logistic Regression (LR) method. The experimental results show that SVM substantially outperforms various other machine learning algorithms for the task of text classification with an accuracy value of 93Keywords: weather classification; twitter; machine learning; support vector machine; multinomial naive bayes; logistic regression.

[2] Purwandari, Kartika & Cenggoro, Tjeng Wawan & Sigalingging, Join & Pardamean, Bens. (2023). Twitter-based classification for integrated source data of weather observations. IAES International Journal of Artificial Intelligence (IJ-AI). 12. 271-283. 10.11591/ijai.v12.i1.pp271-283.

[3] CrowdFlower - Weather sentiment evaluated dataset

https://data.world/crowdflower/weather-sentiment-evaluated