**Projekt: Zastosowanie analizy SHAP w analizie sentymentu metodami NLP**

**Błażej Czaicki**

**Maksymilian Buczek**

**Mikhail Lemiasheuski**

**1. Opis aplikacji:**

Stworzona przez nas aplikacja służąca do analizy sentymentu, docelowo zaprojektowana do analizy tweetów. W tym przypadku, analizujemy zbiór danych tweetów od Donalda Trumpa, ale narzędzie to może być zastosowane do dowolnego zestawu danych tweetów, pod warunkiem, że są one odpowiednio przetworzone i sformatowane.

Analiza sentymentu, którą wykonuje ta aplikacja, polega na przypisaniu tonu emocjonalnego do danego tekstu, innymi słowy określenie, czy jest on pozytywny, negatywny, czy neutralny. Może to mieć zastosowanie w takich dziedzinach jak marketing, analiza mediów społecznościowych, analiza klienta, itp.

Aplikacja korzysta z różnych bibliotek Pythona do przetwarzania języka naturalnego, uczenia maszynowego i interpretacji modelu. Tekst jest procesowany i wektoryzowany do formatu, który może być użyty do trenowania modelu.

Na koniec, używamy biblioteki SHAP do interpretacji modelu, aby zrozumieć, jakie cechy najbardziej wpływają na prognozy modelu.

**2. Wykorzystane biblioteki:**

* **pandas**: Biblioteka do manipulacji i analizy danych. Jest wykorzystywana do ładowania danych z plików CSV i manipulowania ramami danych.
* **re**: Moduł do operacji na wyrażeniach regularnych. Jest używany do przetwarzania tekstu, takiego jak usuwanie linków i znaków specjalnych.
* **nltk**: Natural Language Toolkit, czyli zestaw bibliotek i programów do symbolicznego i statystycznego przetwarzania języka naturalnego. Jest wykorzystywany do lematyzacji słów i usuwania stopwords (czyli często używanych słów, które nie niosą dużo informacji, takich jak "is", "the", "a").
* **shap**: Biblioteka do wyjaśniania prognoz modeli uczenia maszynowego. Jest używana do interpretacji modelu SVM i wizualizacji wyników.
* **textblob**: Biblioteka do przetwarzania danych tekstowych. Jest używana do obliczania polarności sentymentu.
* **sklearn**: Scikit-learn to biblioteka do uczenia maszynowego. Jest używana do różnych zadań, takich jak dzielenie danych na zestawy treningowe i testowe, wektoryzacja TF-IDF i trenowanie modelu SVM.
* **xgboost**: Biblioteka do optymalizacji gradientowej, która implementuje algorytm gradient boosting. Mimo że nie jest używana bezpośrednio w tym kodzie, może być używana do trenowania alternatywnego modelu klasyfikacji.

**3. Wstępne przetwarzanie tekstu:**

Stworzyliśmy funkcję pomocniczą **preprocess\_text()**, która wykonuje kilka operacji na surowym tekście:

* Usunięcie linków: Linki internetowe nie niosą istotnej informacji dla analizy sentymentu, więc są one usuwane. Używamy do tego wyrażeń regularnych.
* Usunięcie znaków specjalnych i cyfr: Znaki specjalne i cyfry również nie są zwykle ważne dla analizy sentymentu. Usuwamy je, ponieważ mogą zakłócać proces analizy.
* Zamiana na małe litery: Zmieniamy wszystkie litery na małe, aby uniknąć rozróżniania słów na podstawie wielkości liter (np. "biden", "Biden" i "BIDEN" są traktowane jako to samo słowo).
* Lematyzacja: Redukujemy słowa do ich podstawowej formy. Na przykład, "running", "runs" i "ran" są redukowane do "run". Dzięki temu różne formy tego samego słowa nie są traktowane jako oddzielne słowa.
* Usunięcie stopwords: Stopwords to często używane słowa, które zazwyczaj nie niosą istotnej informacji (np. "is", "the", "a"). Usuwamy je, aby skupić się na słowach, które faktycznie niosą informacje o sentymencie.

Przetwarzanie tekstu jest kluczowe dla późniejszych etapów analizy sentymentu. Czysty, dobrze przetworzony tekst jest łatwiejszy do analizy i może prowadzić do bardziej dokładnych wyników.

**4. Określanie sentymentu:**

Głównym celem aplikacji jest określanie sentymentu tweetów, co jest realizowane przez funkcje **get\_sentiment()** i **to\_sentiment\_category()**.

* **get\_sentiment()**: Ta funkcja używa biblioteki TextBlob do obliczania polarności sentymentu tekstu. Polarność sentymentu to miara, która określa emocjonalny ton tekstu. Jest to wartość z zakresu od -1 do 1. Wartość -1 oznacza, że tekst jest całkowicie negatywny, wartość 1 oznacza, że tekst jest całkowicie pozytywny. Wartość 0 oznacza neutralność.
* **to\_sentiment\_category()**: Ta funkcja przyjmuje polarność sentymentu jako argument i przypisuje jej kategorię sentymentu: 'positive' (dla wartości większych niż 0), 'negative' (dla wartości mniejszych niż 0) lub 'neutral' (dla wartości równej 0). Ta funkcja jest używana do przygotowania etykiet dla uczenia modelu.

Kategorie sentymentu są następnie dodawane do dataframe jako nowa kolumna 'sentiment\_category'. Takie przygotowanie zestawu pozwala na przekształcenie surowego tekstu w informacje numeryczne, które są łatwe do przetworzenia przez algorytmy uczenia maszynowego.

**5. Wektoryzacja:**

Wektoryzacja jest kluczowym krokiem przygotowaniu danych do uczenia modelu. Ma na celu przekształcić nasze surowe dane tekstowych w formę, którą model uczenia maszynowego mógłby zrozumieć.

Przetwarzamy tekst na numeryczne wektory za pomocą metody TF-IDF (Term Frequency-Inverse Document Frequency). Wektory te mogą już zostać użyte do trenowania modelu. TF-IDF to metoda, która ocenia ważność słowa w dokumencie na podstawie jego częstości w danym dokumencie i w całym korpusie dokumentów. Wykorzystujemy do tego klasę **TfidfVectorizer** z biblioteki **sklearn**.

**6. Trenowanie i ocena modelu:**

Po wstępnym przetworzeniu tekstu, wyznaczeniu sentymentu i przygotowaniu danych, następnym krokiem jest trenowanie modelu.

* **Trenowanie modelu:** Wykorzystujemy algorytm Support Vector Machine (SVM) z biblioteki **sklearn**. SVM to popularny algorytm uczenia maszynowego używany do klasyfikacji i regresji. Trenujemy model na przetworzonym zestawie danych treningowych (X\_train\_tfidf, y\_train), gdzie X\_train\_tfidf to przetworzone tweety a y\_train to odpowiadające im kategorie sentymentu. Używamy parametru probability=True, który zezwala na estymację prawdopodobieństwa.
* **Ocena modelu:** Po wytrenowaniu modelu, oceniamy jego wydajność na zestawie testowym. Wykorzystujemy do tego funkcję **score** z biblioteki **sklearn**, która zwraca średnią dokładność na dane testowe i etykiety.

Tak wytrenowany model pozwala na wykorzystanie go do zbadania sentymentu danych nie użytych dotychczas do uczenia.

**7. Interpretacja modelu za pomocą SHAP:**

Do interpretacji modelu używamy biblioteki SHAP (SHapley Additive exPlanations). SHAP jest narzędziem do wyjaśniania wyników modelu uczenia maszynowego.

* **Wyjaśnienie modelu:** Używamy **KernelExplainer** z biblioteki SHAP do wyjaśnienia naszego modelu SVM. **KernelExplainer** jest ogólnym narzędziem do wyjaśniania modeli, które można użyć do dowolnego modelu uczenia maszynowego. Funkcja **shap\_values** oblicza wartości SHAP dla naszego modelu na danych testowych.
* **Wizualizacja:** Używamy funkcji **summary\_plot** z biblioteki SHAP do wizualizacji wyników. Ta funkcja pokazuje, jakie cechy mają największy wpływ na wyniki modelu. Na wykresie, cechy są uszeregowane według ważności, z najważniejszymi na górze.

Interpretacja modelu jest ważna, ponieważ pomaga zrozumieć, jak model podejmuje decyzje. Dzięki SHAP, możemy zobaczyć, które cechy (słowa) mają największy wpływ na analizę sentymentu. Ta wiedza może pomóc w dalszym udoskonalaniu modelu i lepszym zrozumieniu wyników.

**8. Wnioski:**

Uzyskaną poprawność (około 90 procent) można uznać za dobry wynik (ponieważ sam Donald Trump nie wydaje się, żeby zawsze rozumiał, jakie zabarwienie emocjonalne mają jego posty)

Na podstawie uzyskanej estymacji możemy również wnioskować o tym, jakie słowa bardziej lub mniej wpływają na określenie semantyki wypowiedzi.