Zastosowanie analizy SHAP w analizie sentymentu metodami NLP

Akademia Górniczo-Hutnicza w Krakowie WFiIS

Alicja Kałuża, Daria Kokot, Jakub Baran

Czerwiec 2024

Spis treści

1 Opis zadania

Zadanie polegało na stworzeniu modelu, który pozwoli na klasyfikację sentencji ze zbioru: https://www.kaggle.com/datasets/austinreese/trump-tweets jako negatywne, pozytywne bądź neutralne. Analizy sentymentu dokonano poprzez użycie metod NLP(Natural language processing). Ten projekt pokazuje, jak zaawansowane techniki NLP mogą być stosowane do analizy i klasyfikacji dużych zbiorów danych tekstowych, jak również jak analiza SHAP może dostarczyć cennych informacji na temat działania wytrenowanego modelu.
Kluczowe aspekty:

- Metoda analizy SHAP: Wykorzystanie do interpretacji modelu i zrozumienia, jak poszczególne słowa wpływają na klasyfikację sentymentu.
- Użycie modelu DistilBERT: Lżejsza wersja BERTa, która zapewnia wydajność przy mniejszym zużyciu zasobów.
- Porównanie modeli: Ocena różnic w skuteczności i sposobie oceny sentymentu między modelem DistilBERT a konwolucyjną siecią neuronową.
- Praktyczne zastosowanie NLP: Zastosowanie teorii NLP do analizy dużego zbioru danych tweetów polityka.

2 Struktura projektu

2.1 Przygotowanie

W celu wykonania projeku skonfigurowano środowisko do analizy danych i modelowania posiadające zasoby GPU do przyspieszenia obliczeń. Zamontowano Google Drive, co umożliwiło dostęp do danych przechowywanych na dysku Google bezpośrednio z Colaba. W dalszej kolejności zainstalowano niezbędne biblioteki, w tym transformers, shap i textblob, które są kluczowe do przetwarzania języka naturalnego oraz interpretacji modeli.

2.2 Import Bibliotek

W następnej kolejności zaimportowano wszystkie potrzebne w projekcie biblioteki

2.3 Czyszczenie Danych

W ramach projektu przetwarzania języka naturalnego, przygotowano funcje służące do oczyszczenia oraz augmentacji danych tekstowych. Do oczyszczania, czyli pozbycia się nieistotnych dla rozważanego zadania fragmentów tekstu, wykorzystano zasoby z biblioteki NLTK (Natural Language Toolkit). Aby zwiększyć zbiór danych, stworzono funkcje do generowania synonimów dla słów w tekstach oraz augmentacji tekstów przez zastępowanie losowo wybranych słów ich synonimami. Zarówno oczyszczanie danych, jak i augmentację wykonano równolegle, aby przyspieszyć te zadania.

Proces czyszczenia danych wyglądało następująco:

1. Normalizacja Tekstu poprzez konwersję na małe litery.
2. Usuwanie odnośników URL za pomocą wyrażenia regularnego.
3. Usuwanie Znaków Unicode (nie-ASCII)
4. Usuwanie Interpunkcji: Usunięcie interpunkcji za pomocą translacji znaków.
5. Tokenizacja Tekstu, czyli podzielenie tekstu na słowa.
6. Usuwanie Stopwords, czyli filtracja powszechnych słów nieistotnych.
7. Lematyzacja Słów, czyli redukcja słów do ich form podstawowych.
8. Ponowne złożenie wyczyszczonych słów w jeden ciąg znaków w celu dalszego przetwarzania

2.4 Ładowanie, Czyszczenie i Augmentacja Danych

W tej sekcji przeprowadzono proces ładowania danych, czyszczenia i augmentacji danych tekstowych przygotowanych korzystając z wcześniej przygotowanych funkcji.

Ten proces nie tylko poprawił jakość danych poprzez standaryzację i augmentację, ale również zwiększył potencjał modeli do dokładniejszej klasyfikacji i analizy tekstu.

2.5 Funkcje Pomocnicze do Treningu Modelu

W tej sekcji zdefiniowano kilka kluczowych funkcji pomocniczych, które są niezbędne do przetwarzania, trenowania i oceny modelu klasyfikacji tekstu.

2.6 Przygotowanie Danych dla Modelu

Te kroki przygotowują dane tekstowe do dalszego wykorzystania w procesie trenowania i oceny modelu klasyfikacji sentymentu przy użyciu BERTa. Dzięki tokenizacji, enkodowaniu etykiet i odpowiednim podziałom danych, zapewniono odpowiednie przygotowanie danych przed przystąpieniem do budowy modelu.

2.7 Inicjalizacja i Trenowanie Modelu

W tej sekcji zainicjowano i wytrenowano model klasyfikacji sentymentu przy użyciu DistilBERTa, a dokładniej "distilbert-base-uncased". Proces obejmował kilka kluczowych kroków:

- Określono urządzenie do treningu modelu – GPU, jeśli jest dostępne, w przeciwnym razie CPU.
- Utworzono model DistilBERTa z odpowiednią liczbą etykiet num_labels.
- Ustawiono optymalizator AdamW z parametrami learning rate $2e^{-5}$ i epsilon $1e^{-8}$.
- Użyto linearnego harmonogramu uczenia z funkcją get_linear_schedule_with_warmup, który zmienia współczynnik uczenia w zależności od liczby kroków trenowania i okresu rozgrzewki.
- Zdefiniowano kryterium CrossEntropyLoss do obliczania straty.
- Użyto GradScaler z biblioteki PyTorch do skalarnej obróbki gradientów na GPU.
- Wywołano funkcję train_model, przekazując model, zbiory treningowy i walidacyjny, kryterium straty, optymalizator, scheduler, liczbę epok num_epochs=20 oraz skalara. Funkcja ta trenowała model i śledziła straty treningowe oraz walidacyjne.
- Po zakończeniu trenowania zapisano wagi modelu do pliku model_weights.pth.

Podsumowując, przeprowadzono inicjalizację modelu DistilBERT, optymalizatora AdamW, harmonogramu uczenia i kryterium straty. Następnie model został przetrenowany na przygotowanych danych, a wagi modelu zapisano do pliku. Dzięki tym krokom, model został przygotowany do dalszej oceny i wykorzystania w zadaniach klasyfikacji tekstu. Dzięki zastosowaniu techniki Early Stopping oszczędzono zasoby, zmniejszono szansą na przetrenowanie oraz poprawiono efektywność modelu. Już po 13 epokach trening się zakończył.

2.8 Ewaluacja Modelu

Przeprowadzono ewaluację wytrenowanego modelu klasyfikacji tekstu przy użyciu BERTa, w celu oceny jego dokładności na zbiorach treningowym i walidacyjnym oraz wizualizacji strat podczas trenowania.

2.9 Predykcja sentymentu

Zaimplementowano funkcję służącą do przewidywania sentymentu danego tweeta.

2.10 Macierz pomyłek

Na podstawie wyznaczonego sentymentu utworzono macierz pomyłek reprezentującą, jak dobrze model radzi sobie z przypisywaniem sentymentu.

2.11 Analiza i statystyki zbioru Real Donald Trump

Wytrenowany podel wykorzystano do sprawdzenia sentymentu wypowiedzi polityka Donalda Trumpa. Przedstawiono jaki odsetek jego tweetów ma dany sentyment. Przedstawiono również po kilka przykładów postów z danym sentymentem.

2.12 Analiza SHAP Część 1 - Przygotowanie Danych oraz Funkcji

Przygotowano dane oraz funkcje pomocne przy analizie Shalpey'a

2.13 Analiza SHAP Część 2 - Wykresy

Przedstawione wykresy zawierają informację, które słowa mają największe znaczenie w kontekście sentymentu.

2.14 Analiza SHAP Część 3

Wykresy przedstawiają sentyment poszczególnych słów w przykładowych zdaniach.

3 Teoria, narzędzia i technologie

3.1 O DistilBert i NLP

Do rozwiązania zadania wykorzystano metody NLP. Proces rozpoczęto od tokenizacji (proces podziału ciągu tekstu na mniejsze jednostki) sentencji użytych jako zbiór treningowy.

W ten sposób przedstawiono zbiór jako zrozumiały dla modelu - każdy z tokenów został przedstawiony w postaci liczby. Zastosowano "DistilBertForSequenceClassification", czyli odchudzoną wersja modelu BERT (Bidirectional Encoder Representations from Transformers), opracowaną przez firmę Hugging Face. Model ten jest znacznie lżejszy i szybszy, co czyni go bardziej efektywnym pod względem obliczeniowym, przy jednoczesnym zachowaniu wysokiej jakości wyników.

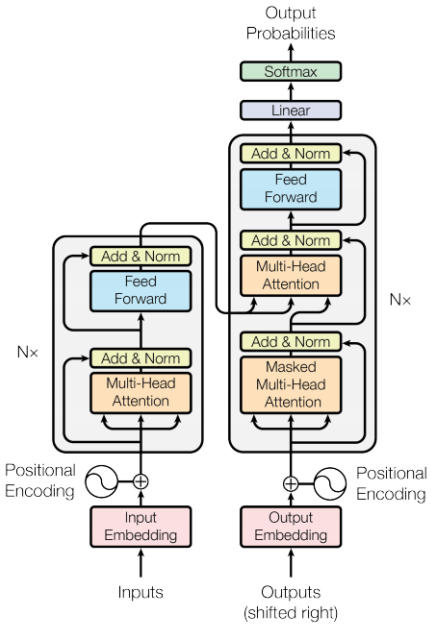
3.2 Kluczowe cechy DistilBERT

- DistilBERT jest mniejszy o 40%
- Dzięki mniejszemu rozmiarowi, DistilBERT jest około 60% szybszy podczas treningu i inferencji niż oryginalny BERT
- Porównywalna jakość (około 97% wydajności oryginalnego BERT-a)

3.3 Transformery

Sieci neuronowe typu transformer mają na celu rozwiązywanie zadań sekwencyjnych. Czyli podajemy jedną sekwencje na wejściu i otrzymujemy na wyjściu inną. Wyjście może być na przykład tłumaczeniem lub streszczeniem tekstu. Można tę sieć tak na prawdę zastosować do prawie każdego zadania dotyczącego NLP. Jest to alternatywa dla sieci rekurencyjnych, które były wcześniej powszechnie wykorzystywane do NLP.

3.4 Architektura sieci neuronowej typu transformer



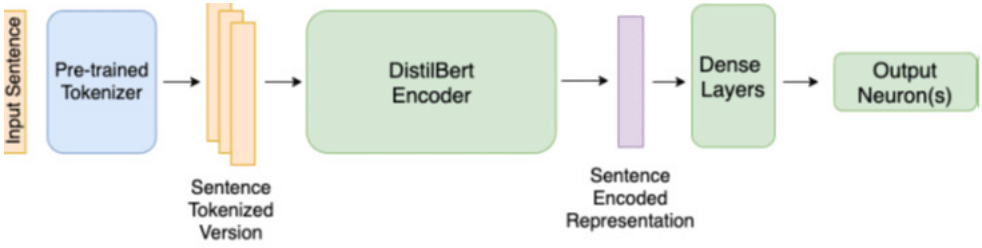
Rysunek 1: Architketura sieci neuronowej typu transfomer https://miroslawmamczur.pl/wp-content/uploads/2020/12/architektura.png

Architektura sieci:

1. Input/Output embeding - zamiana słów na liczby i osadzenie w przestrzeni wielowymiarowej.
2. Positional encoding - dodanie informacji o kontekście na podstawie pozycji słowa w zdaniu.
3. Multi-Head attention - wyliczanie jak istotne są słowa w zdaniu.
4. Feed Forward - używane w celu wprowadzenia nieliniowości i zwiększenia zdolności uczenia się skomplikowanych wzorców.
5. Masked Multi-Head attention - tutaj użycie maskowania, czyli nałożenia maski na macierz attention, która blokuje (ustawia na bardzo niską wartość) wszystkie pozycje odpowiadające przyszłym tokenom.
6. Linear - odpowiada za przekształcenie wewnętrznych reprezentacji modelu na ostateczne przewidywania tokenów.
7. Softmax - przekłada wynik na prawdopodobieństwo.

3.5 Model BERT

Model BERT jest enkoderem, czyli tylko lewą częścią rysunku. Jest w projekcie wykorzystany do stworzenia wektorowej reprezentacji tekstu, z której korzystamy przeprowadzając klasyfikację do odpowiedniego sentymentu.

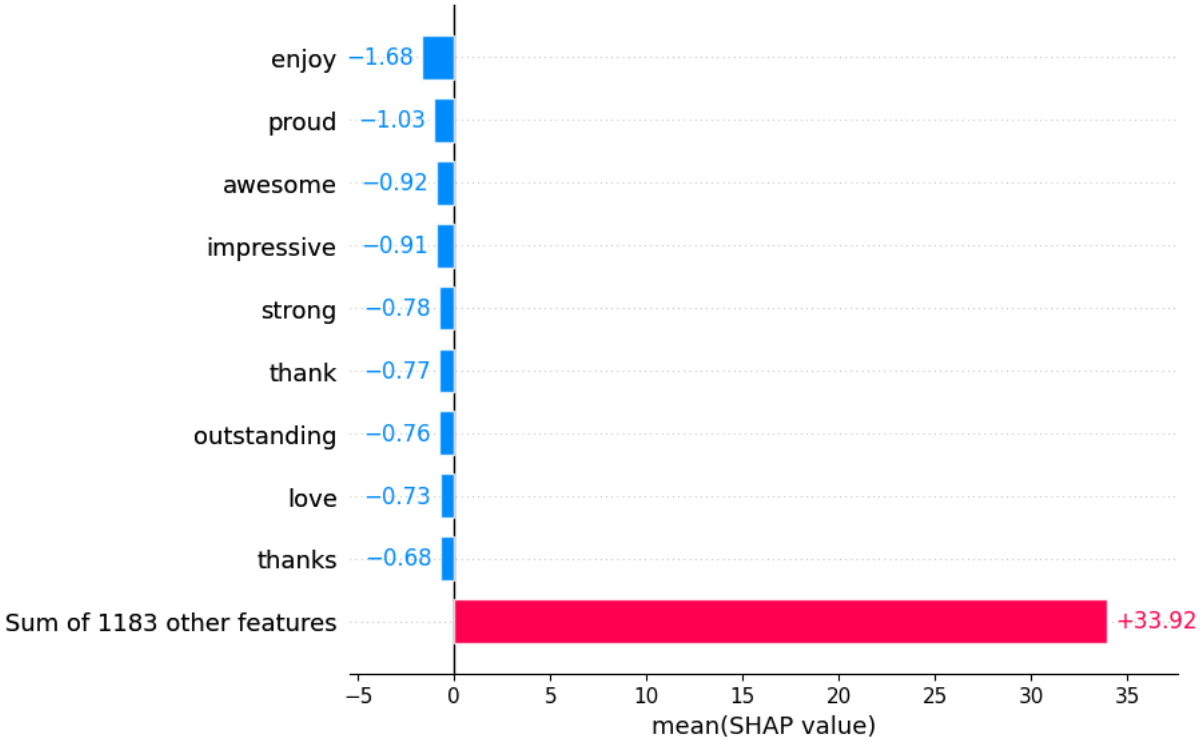


Rysunek 2: Architketura sieci neuronowej z wykorzystaniem enkodera DistilBert https://www.researchgate.net/figure/Model-Architecture-for-DistilBERT_fig1_354140736

Dokonano treningu modelu. Osiągnięto zadowalający rezultat: training accuracy - ok. 98%, Validation accuracy: 90% Następnie przetestowano model za pomocą sentencji ze zbioru z polecenia zadania. Także ztokenizowano te słowa. Według naszego modelu większość sentencji jest negatywna. Dlaczego? Tego dowiemy się z analizy SHAP.

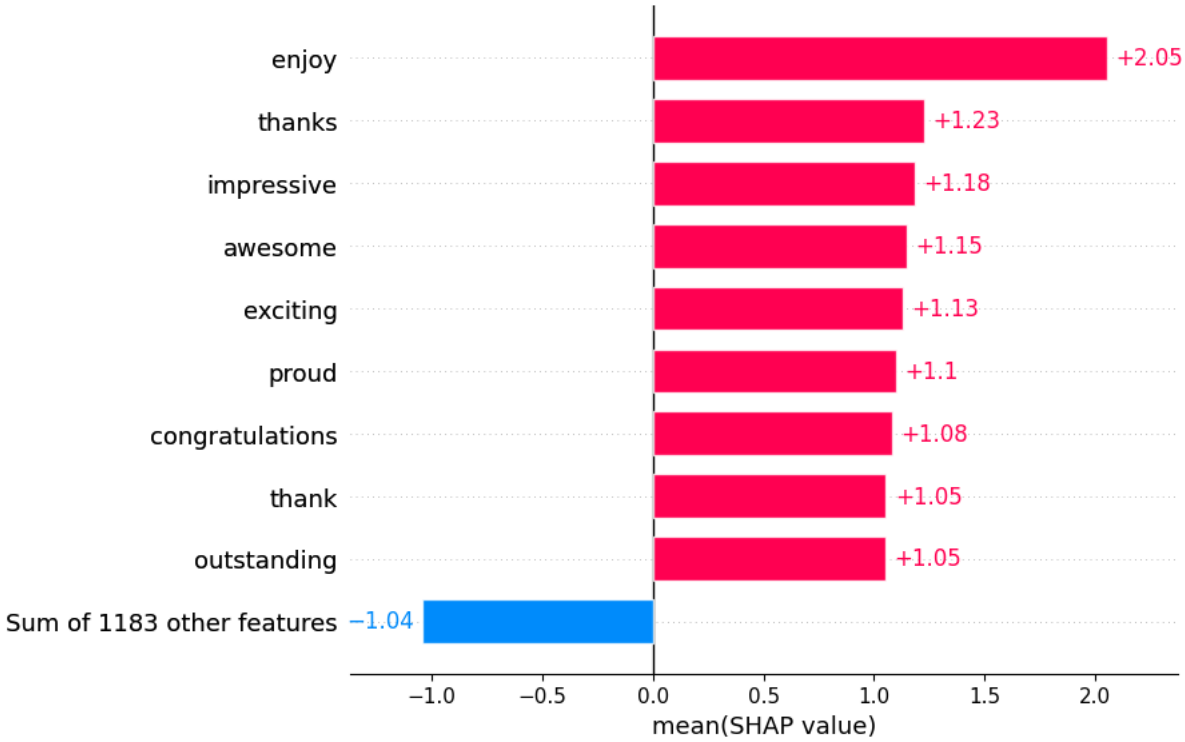
4 Analiza SHAP

Wartość Shapleya jest pojęciem z teorii gier, stworzonym w 1953 r. przez amerykańskiego matematyka Lloyda Shapleya. Określa sposób podziału zysku pomiędzy uczestników gry kooperatywnej, czyli takiej, w której gracze mogą łączyć się w koalicje celem uzyskania określonego wyniku. To jaki wkład daje każdy gracz w wynik jest istotą analizy. W przypadku naszej analizy każdy gracz - to poszczególne słowa składające się na sentencje. Do określenia konkretnej wartości wykorzystujemy specjalny wzór. W skrócie wartości SHAP wyjaśniają, średni udział każdej funkcji danych wejściowych we wszystkich możliwych kombinacjach funkcji. Analiza SHAP(SHapley Additive exPlanations) jest kluczowa do zbadania wytrenowanego modelu. Dzięki wartości Shapley'a dowiemy się, dlaczego model podjął taką a nie inną decyzję. Do analizy wykorzystano specjalne funkcje dostępne w bibliotece shap. Każda wartość SHAP pokazuje, w jaki sposób dana cecha (w przypadku naszego modelu konkretne słowo) przyczynia się do ostatecznej decyzji modelu. Aby wynik analizy był bardziej ogólny analizowano kolejno po 100 sentencji z testowanego zbioru. Poniżej przedstawiono wyniki analizy.



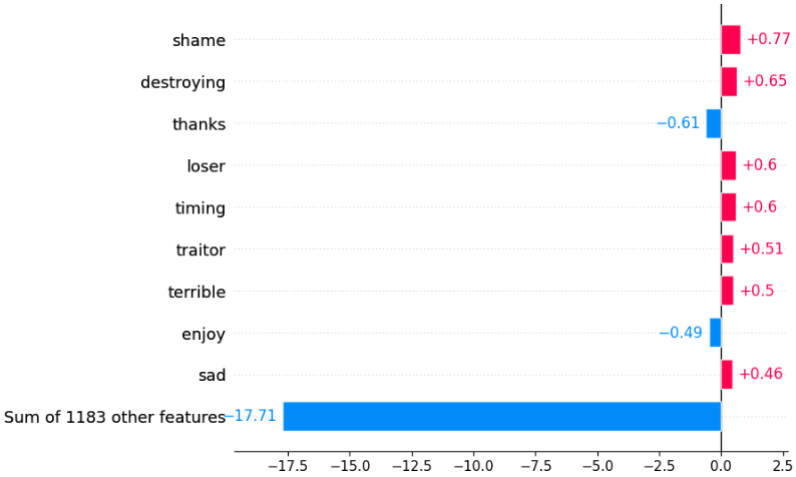
Rysunek 3: Wykres przedstawia wynik obliczenia średniego wpływu wszystkich słów ze 100 pierwszych sentencji ze zbioru na przyporządkowanie do klasy "neutral". Na wykresie umieszczono kilka słów, które mają największy wpływ na decyzję modelu.

Do słów, które najbardziej "przeszkadzają"w przypisaniu do klasy neutral należą enjoy, proud, awesome, impressive, strong. Możemy zauważyć, że wyjaśnienie dlaczego model podejmuje taką a nie inną decyzję jest logiczne - słowo awesome nie ma neutralnego wydźwięku, ani słowo strong.



Rysunek 4: Wyniki obliczenia średniego wpływu wszystkich słów na klasę "positive".

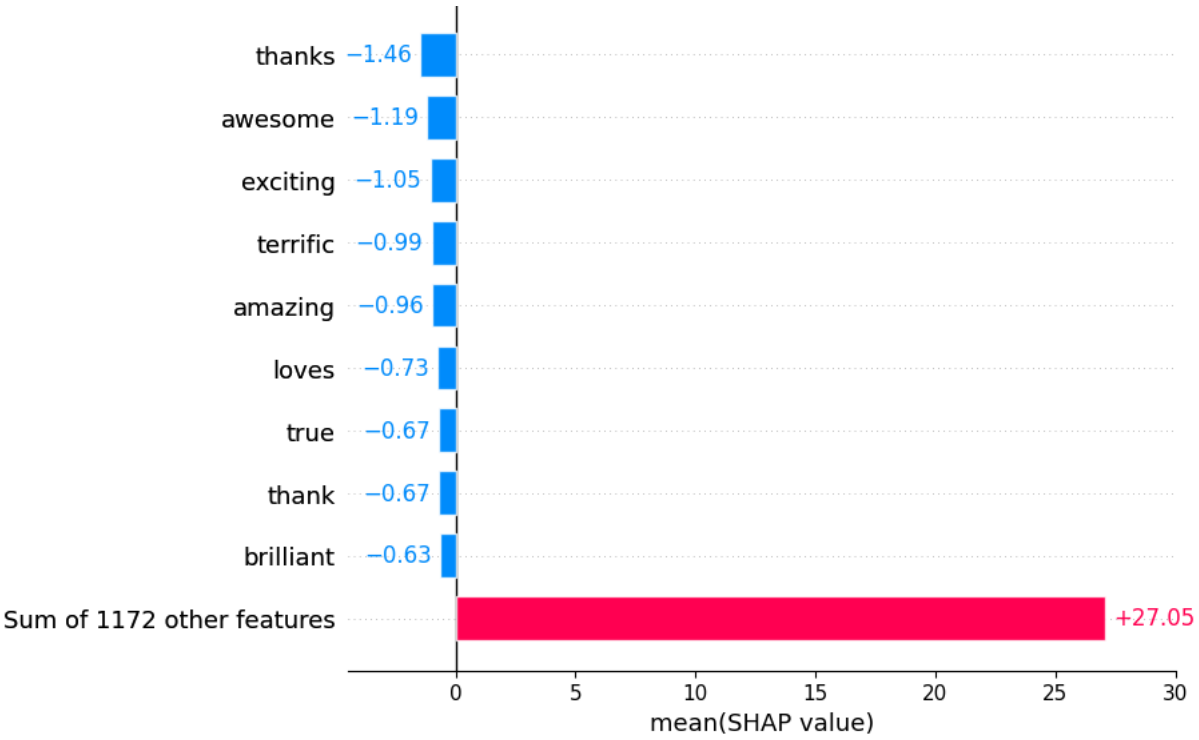
Powyższy wykres pokazuje słowa, które mają największy wpływ na podejmowaną przez model decyzję. Słowa enjoy, thanks, impressive i awesome zwiększają szansę na przypisanie sentencji do klasy "positive". Te słowa budzą emocje pozytywne. W tym przypadku także możemy zauważyć, że "myślenie"modelu jest zgodne z prawdą.



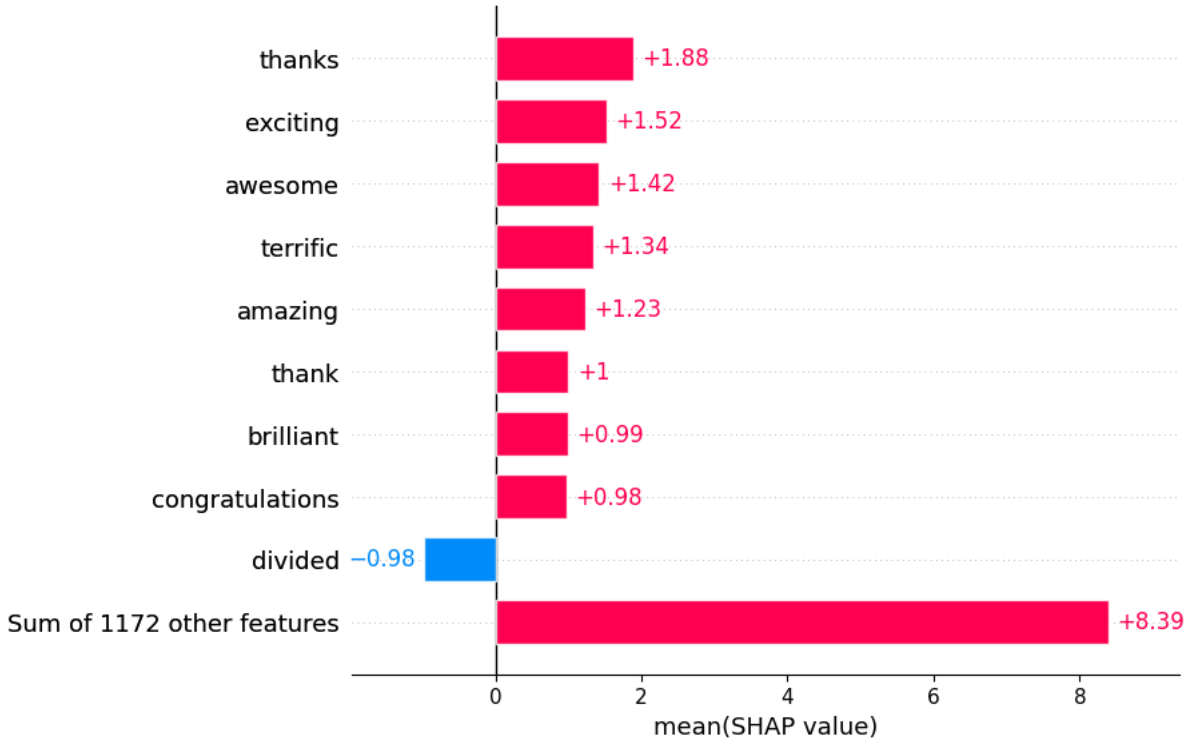
Rysunek 5: Wyniki obliczenia średniego wpływu wszystkich słów na klasę "negative".

Powyższy wykres przedstawia podstawę wnioskowania modelu przy definiowaniu sentencji jako negatywne. Na taką decyzję największy wpływ mają tokeny: shame, destroying, loser,timing,traitor,terrible. Te słowa również dla nas mają bardziej negatywny niż pozytywny wydźwięk. Natomiast jako słowo oddalające model od przypisania sentencji wydźwięku negatywnego to słowo thanks oraz enjoy. Te słowa rzeczywiście nie mają wiele wspólnego z negatywnymi emocjami.

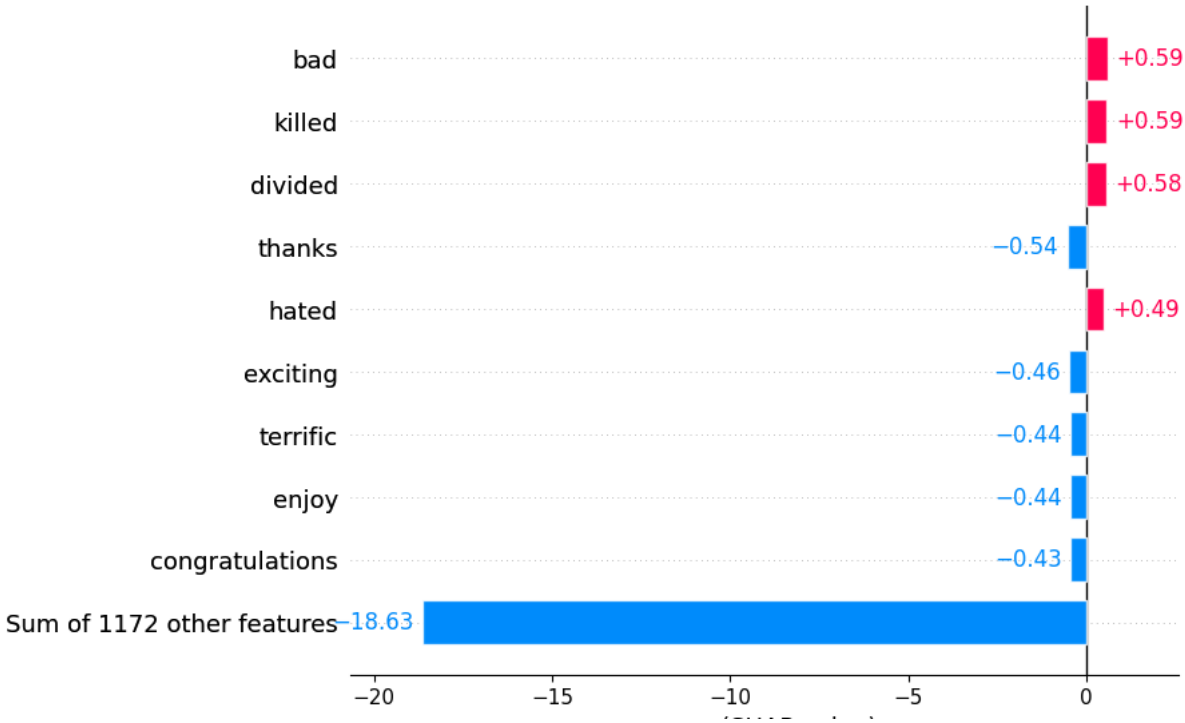
Analiza SHAP wykonana na pierwszych 100 sentencjach ze zbioru testowego daje oczekiwane rezultaty - sposób w jaki model podejmuje decyzję jest prawidłowy - analogiczny do tego, jaką człowiek podjąłby decyzję analizując pojedyńcze słowa. Przyjrzyjmy się analizie kolejnych 100 sentencji.



Rysunek 6: Wyniki obliczenia średniego wpływu wszystkich słów na klasę "neutral".

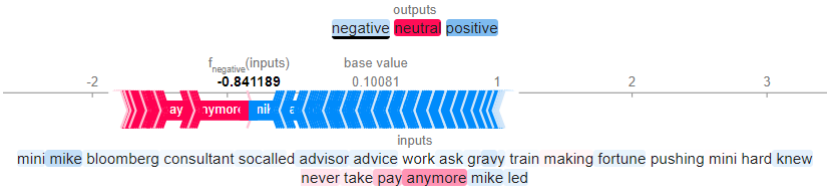


Rysunek 7: Wyniki obliczenia średniego wpływu wszystkich słów na klasę "positive".

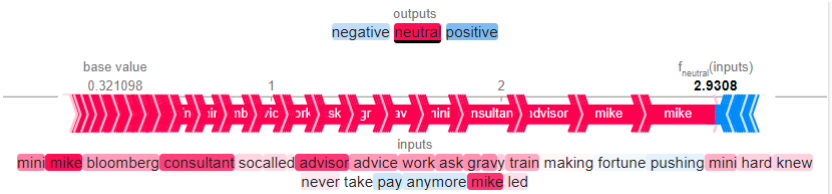


Rysunek 8: Wyniki obliczenia średniego wpływu wszystkich słów na klasę "negative".

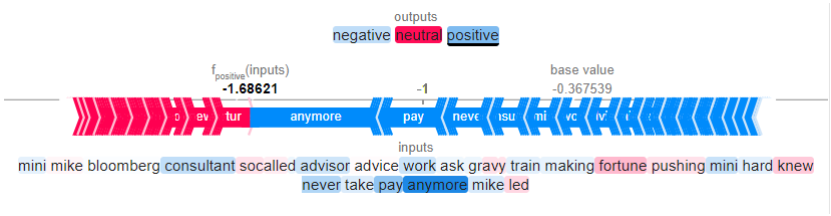
Powyższe wykresy także są dowodem na prawidłowe działanie modelu. Cecha, która najbardziej wpływa na wynik pozytywny to słowo thanks - słowo jak najbardziej uważane za pozytywne. Wykresy poniżej przedstawiają jak pojedyńcze słowa w losowym zdaniu ze zbioru wpływają na dopasowanie sentencji do danej klasy. Zdanie zostało uznane przez model za neutralne. Kolor niebieski pokazuje które słowa mają mniejsze znaczenie w klasyfikacji słowa do sprawdzanej klasy, a kolor czerwony pokazuje, które mają większe znaczenie. Możemy zobaczyć, że gdy zdanie jest neutralne, to zdecydowana większość słów jest zaznaczona na kolor czerwony na wykresie, pokazującym które ze słów i w jaki sposób wpływają na przypisanie do klasy "neutral".



Rysunek 9:



Rysunek 10:



Rysunek 11:

Wszystkie trzy wykresy pokazują, jakie znaczenie w decyzji modelu ma każde ze słów w zdaniu. W przypadku tego zdania widzimy, że zdecydowana większość słów nacechowana jest negatywnie. Wskazuje na to kolor czerwony danej etykiety. Oznacza to, że słowa negatywne w największym stopniu przyczyniają się do wyniku.

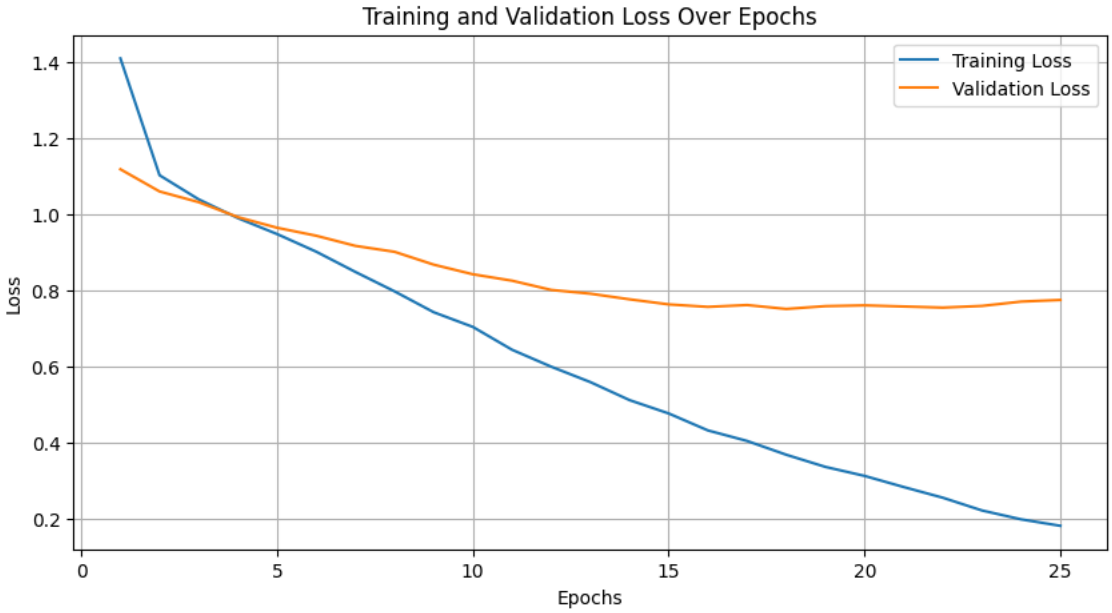
5 Porównananie wyników różnych rozwiązań

W tej sekcji przedstawione zostanie porównanie dwóch modeli, które zostały wytrenowane. Jednym z nich jest ten opisywany powyżej, natomiast drugi to konwolucyjna sieć neuronowa, której dokladność wynosi (Training Accuracy: 0.9988, Validation Accuracy: 0.6096). Ma więcej wyjść niż poprzednio opisywany model. Porównamy jak modele oceniły kilka pierwszych sentencji. Pierwsza jest ocena modelu wykorzystującego "DistilBertForSequenceClassification",druga sieci neuronowej:

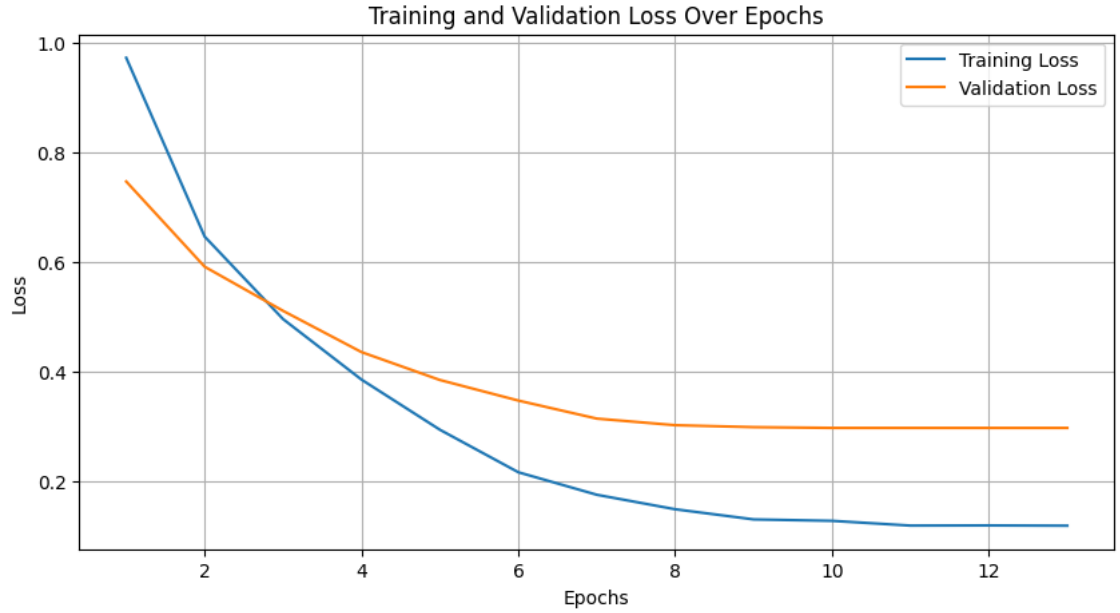
- Text: Mini Mike Bloomberg's consultants and so-called "advisors"(how did that advice work out? Don't ask!), are on the "gravy train" and all making a fortune for themselves pushing Mini hard, when they knew he never had what it takes. Don't pay them anymore Mike, they led you down....
- neutral
- negative
- .....a very dark and lonely path! Your reputation will never be the same!
- negative
- positive
- We win in our lives by having a champion's view of each moment. –Donald J. Trump http://tinyurl.com/pqpfvm Predicted Sentiment: neutral
- neutral
- negative

Jak widać wyniki obu modeli różnią się we wszystkich przypadkach. To pokazuje, że model, na którego architekturę składa się konwolucyjna sieć neuronową albo został najprawdopodobniej wytrenowany ze złymi parametrami, mimo częstych zmian i prób trenowania modelu z różną wartością parametrów.

W przypadku tego modelu nastąpiło zjawisko przeuczenia, co widać na wykresie przedstawiającym przebieg uczenia.



Rysunek 12: Wykres przedstawiający przebieg uczenia modelu wykorzystującego konwolucyjną sieć neuronową.

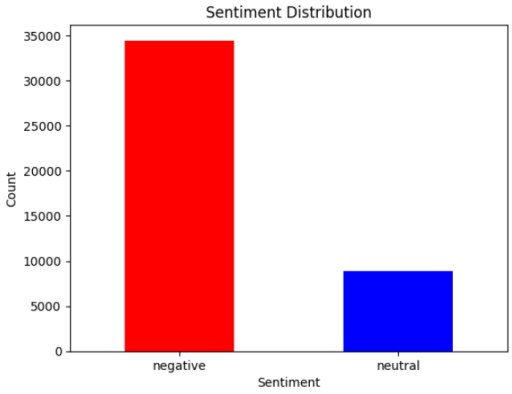


Rysunek 13: Wykres przedstawiający przebieg uczenia modelu wykorzystującego sieć neuronową typu transformer.

Mimo lepszych wyników, nadal widoczne jest przeuczenie. Zjawisko to może być spowodowane różnymi czynnikami m.in. zbyt małą ilością i różnorodnością danych treningowych, zbyt skomplikowanym modelem, zbyt długim treningiem sieci. Aby zapobiec tym zjawiskom zastosowano następujące techniki: augmentacja danych, Early Stopping (zatrzymanie treningu modelu, gdy zauważalny jest brak widocznej poprawy wyniku)

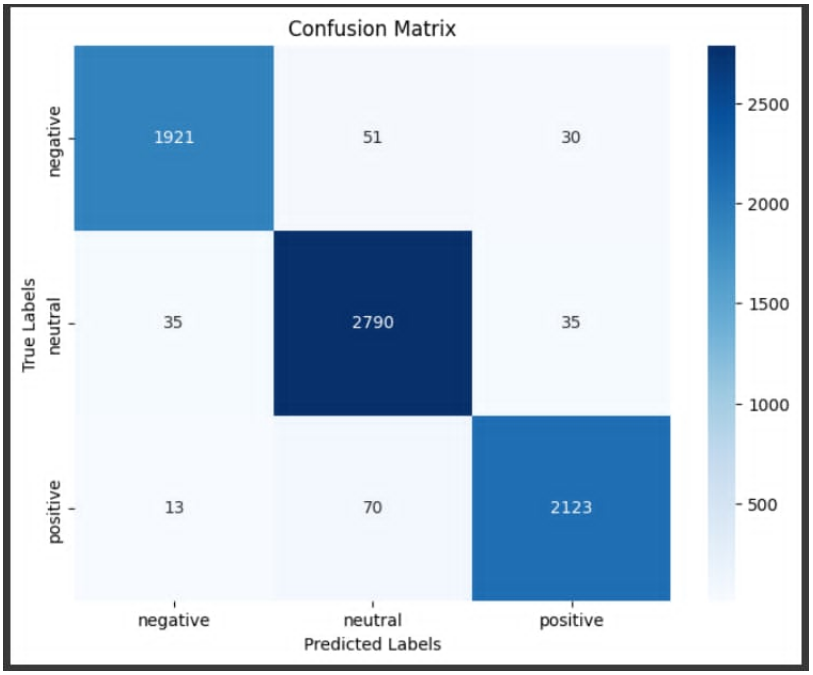
6 Podsumowanie

Wytrenowany przez nasz zespół model spełnia swoją funkcję analizatora sentencji. Model transformera okazał się najbardziej skuteczny i najszybszy spośród tych rozważanych przez nasz zespół. Jednym z rozważanych rozwiązań był model sekwencyjny wykorzystujący LSTM (Long short-term memory). Trenowanie modelu było czasochłonne i nie przynosiło oczekiwanych rezultatów. Próbowaliśmy też rozwiązać problem konwolucyjną siecią neurnową. Wyniki (Training Accuracy: 0.9073, Validation Accuracy: 0.5949) nie były tak zadawalające jak dokładność utworzonego modelu BERT. Większość sentencji zamieszczonych przez Donalda Trumpa jest negatywna, co widać na poniższym wykresie. Nie ma żadnych pozytywnych sentencji. Być może jest to spowodowane innym charakterem tweetów wykorzystanych do treningu sieci, na przykład zbyt mało politycznym.



Rysunek 14: Wykres przedstawiający ilość sentencji neutralnych, pozytywnych i negatywnych w sentencjach Donalda Trumpa.

Dowodem na to, że skonstruowany przez nasz zespół model wykorzystujący sieć typu transformer działa poprawnie jest przedstawiona poniżej macierz pomyłek.



Rysunek 15: Wykres przedstawiający, ile razy model prawidłowo i nieprawidłowo sklasyfikował dane próbki do poszczególnych klas.

7 Wnioski

- Model został satysfakcjonująco dobrze wytrenowany, co potwierdza macierz pomyłek na Rys. 15
- Zauważalne jest lekkie przeuczenie na wytrenowanym modelu. Spowodowane jest to prawdopodobnie takimi czynnikami jak niewystarczająca ilość i różnorodność danych, zbyt skomplikowany model
- Normalizacja, oczyszczanie i augmentacja danych pozwoliły znacznie polepszyć wyniki sieci
- Aby model mógł osiągnąć lepsze wyniki będzie potrzebna dogłębna analiza przygotowanego kodu, zwiększenie i urozmaicenie zbioru testowego, dostosowanie hiperparametrów modelu oraz lepsze zasoby obliczeniowe.
- Model podejmuje decyzje logicznie, co zostało pokazane w analizie SHAP.
- Wyniki potwierdzają, że zastosowanie modelu DistilBERT może być bardziej odpowiednie do tego konkretnego zadania niż sieci konwolucyjne, ze charakterystykę dostępnych danych.
- Zastosowanie technik transfer learningu z pre-trenowanymi modelami przyniosło znaczną poprawę efektywności, co sugeruje ich wartość w kontekście ograniczonych zasobów obliczeniowych.
- Dalsze badania nad dostosowywaniem strategii augmentacji danych mogą pomóc w dalszej redukcji efektów przeuczenia i zwiększeniu ogólnej stabilności modelu.
- Analiza wpływu rozmiaru zbioru danych treningowych na wyniki modelu sugeruje, że większy zbiór danych mógłby znacznie poprawić jego zdolności do generalizacji.

8 Bibliografia

- https://huggingface.co/distilbert/distilbert-base-uncased (model DistilBERT)
- https://www.kaggle.com/datasets/abhi8923shriv/sentiment-analysis-dataset/data?fbclid=IwAR3lQHVWrqNl_WAetxdhBo_CwbAv5gvQZDBtMo4ToOXQ (dane do treningu modelu)
- https://miroslawmamczur.pl/wp-content/uploads/2020/12/architektura.png (Architketura sieci neuronowej typu transfomer)
- https://www.researchgate.net/figure/Model-Architecture-for-DistilBERT_fig1_354140736 (Architketura sieci neuronowej z wykorzystaniem enkodera DistilBert)
- Ćwiczenia i wykłady z Metod Inteligencji Obliczeniowej AGH (dr hab. inż. Piotr Kowalski, dr hab. inż. Szymon Łukasik, mgr Karolina Wadowska, mgr inż. Maciej Trzciński
- https://huggingface.co/learn/computer-vision-course/unit2/cnns/introduction
- https://shap.readthedocs.io/en/latest/
- https://www.kaggle.com/datasets/austinreese/trump-tweets

9 Link do projektu

- https://github.com/DK333D/RealDonaldTrumpTweets