Akademii Górniczo-Hutniczej im. Stanisława Staszica

w Krakowie

Zastosowanie sieci rekurencyjnych do analizy stanu psychicznego w tekstach

Sieci neuronowe i uczenie głębokie

Barbara Kania

Informatyka i Ekonometria

Spis treści:

Wprowadzenie

Współczesne technologie oparte na sztucznej inteligencji odgrywają coraz większą rolę w zrozumieniu i poprawie zdrowia psychicznego. Analiza tekstów, takich jak wpisy w mediach społecznościowych, dzienniki osobiste czy rozmowy z terapeutami, może dostarczyć cennych informacji o stanie psychicznym jednostki. Dzięki zastosowaniu sieci rekurencyjnych możliwe jest skuteczne przetwarzanie danych sekwencyjnych, takich jak tekst, w celu wykrywania stanów emocjonalnych i psychicznych.

1. Opis problemu i cel projektu

Celem projektu jest wykorzystanie sieci rekurencyjnych, takich jak LSTM (Long Short-Term Memory) do analizy tekstów pod kątem stanów psychicznych. Projekt skupia się przede wszystkim na klasyfikowaniu tekstów do różnych kategorii stanu psychicznego.

Tego typu modele mogą znaleźć zastosowanie w detekcji stanów psychicznych w takich obszarach jak wykrywanie objawów depresji, lęku lub myśli samobójczych na wczesnym etapie, automatyczna analiza emocji w tekstach w celu wspomagania pracy terapeutów oraz monitorowanie nastrojów w populacji na podstawie publicznych wpisów. Ponadto może być przydatne w aplikacjach terapeutycznych, np. gdzie chatboty wspierają osoby zmagające się z trudnościami emocjonalnymi i do wykrywanie wczesnych sygnałów depresji, lęku, myśli samobójczych itp. aby odpowiednio móc dopasować specjalistę.

1.1 Dane

Dane zostały pozyskane ze strony https://huggingface.co/datasets , zawierają informacje na temat stanów psychicznych użytkowników na podstawie ich wypowiedzi tekstowych. Poniżej szczegółowy opis struktury danych oraz ich klas. Dane zawierają 52 681 wierszy.

Dane są podzielone na 7 klas, reprezentujących różne stany psychiczne:

Klasa	Liczba	Opis
Normal	16,351	Teksty, które nie wskazują na żadne znaczące zaburzenia psychiczne.
Depression	15,404	Wpisy sugerujące depresję, np. smutek, brak motywacji.
Suicidal	10,653	Wypowiedzi związane z myślami samobójczymi lub skrajnie negatywnym nastrojem.
Anxiety	3,888	Wpisy wskazujące na lęki, nerwowość lub napięcie.
Bipolar	2,877	Teksty odzwierciedlające objawy związane z zaburzeniem dwubiegunowym.
Stress	2,669	Wpisy związane z odczuwaniem stresu, przeciążenia emocjonalnego.
Personality disorder	1,201	Wypowiedzi sugerujące obecność zaburzeń osobowości.

Jak można zauważyć dane są silnie niezbalansowane. Model może faworyzować klasy dominujące, ignorując rzadziej występujące przypadki. W celu poprawy równowagi

zastosowano oversampling, Dodano więcej przykładów do klas mniejszościowych, aby model miał większą szansę nauczyć się ich wzorców.

1.2 Przegląd literatury

Zastosowanie sieci neuronowych do analizy tekstów w celu klasyfikacji stanu psychicznego zyskało na popularności dzięki postępowi w dziedzinie przetwarzania języka naturalnego (NLP). Modele oparte na głębokim uczeniu, takie jak RNN (Recurrent Neural Networks), LSTM (Long Short-Term Memory), GRU (Gated Recurrent Units) oraz bardziej zaawansowane modele transformatorowe, np. BERT (Bidirectional Encoder Representations from Transformers) umożliwiają wykrywanie złożonych wzorców w tekstach. Oto kilka publikacji dotyczących wykorzystania sieci neuronowych do klasyfikacji stanu psychicznego na podstawie tekstów:

1. „Sieci neuronowe w modelowaniu zaburzeń neuropsychologicznych i chorób psychicznych”, Włodzisław Duch

Włodzisław Duch przedstawia zastosowanie sztucznych sieci neuronowych w badaniach nad zaburzeniami psychicznymi. Sieci te umożliwiają analizę procesów poznawczych, takich jak pamięć czy percepcja, oraz modelowanie objawów chorób, takich jak schizofrenia, depresja czy choroba Alzheimera. Autor podkreśla ich zdolność do wykrywania złożonych wzorców w danych klinicznych i symulowania deficytów poznawczych. Jednocześnie zauważa ograniczenia tych metod, związane z uproszczoną strukturą modeli i trudnościami w interpretacji wyników.

2.“Adapting Deep Learning Methods for Mental Health Prediction on Social Media", Ivan Sekulić, Michael Strube (2020)

Praca ta bada zastosowanie głębokich sieci neuronowych do przewidywania stanu zdrowia psychicznego na podstawie danych z mediów społecznościowych. Autorzy wykorzystują hierarchiczną sieć uwagi (Hierarchical Attention Network) do klasyfikacji użytkowników cierpiących na różne zaburzenia psychiczne, osiągając lepsze wyniki niż wcześniejsze metody.

3. "Mental Illness Classification on Social Media Texts using Deep Learning and Transfer Learning", Iqra Ameer, Muhammad Arif, Grigori Sidorov, Helena Gómez-Adorno, Alexander Gelbukh (2022)

Praca ta skupia się na klasyfikacji pięciu powszechnych chorób psychicznych na podstawie danych z platformy Reddit. Autorzy wykorzystują tradycyjne metody uczenia maszynowego, głębokie sieci neuronowe oraz transfer learning do detekcji zaburzeń takich jak depresja, lęk, zaburzenie afektywne dwubiegunowe, ADHD i PTSD.

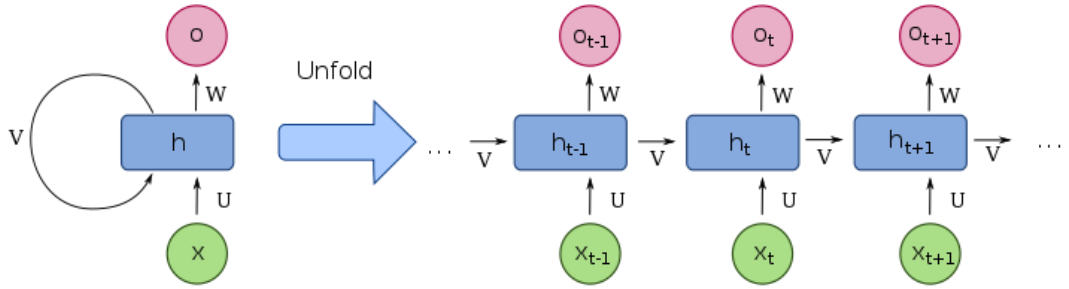
2. Metodologia

Sieci neuronowe to systemy obliczeniowe z połączonymi węzłami, które działają podobnie do neuronów w ludzkim mózgu. Wykorzystanie algorytmów pozwala im rozpoznawać ukryte prawidłowości i korelacje w nieprzetworzonych danych, grupować je i klasyfikować przy czym nieustannie się uczyć i doskonalić. Wyróżnia się kilka typów sieci neuronowych takich jak proste Sieci Neuronowe (FNN), Rekurencyjne Sieci Neuronowe

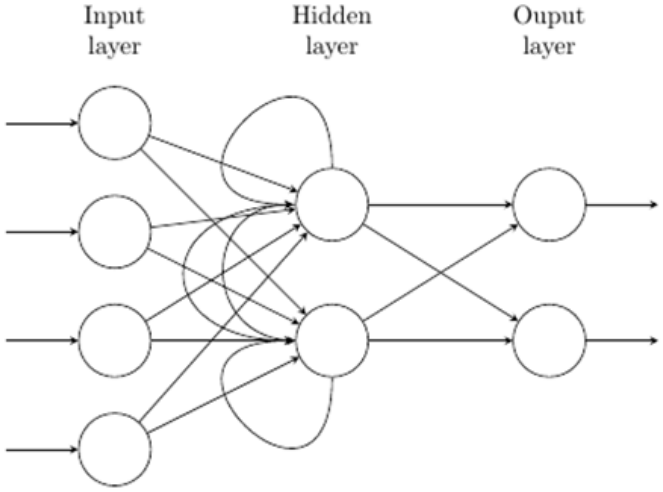
(RNN), Konwolucyjne Sieci Neuronowe (CNN), Sieci typu Transformer oraz Generative Adversarial Networks (GAN).

W projekcie zastosowano sieci rekurencyjne RNN, które są rodzajem sieci neuronowej. Są kluczowym elementem dziedziny sztucznej inteligencji, wykorzystywanym w takich zadaniach jak klasyfikacja, regresja, rozpoznawanie obrazów, przetwarzanie języka naturalnego (NLP) czy prognozowanie czasowe.

Struktura Rekurencyjnej Sieci Neuronowej (RNN)



W sieciach neuronowych można wyróżnić 3 typy warstw:



Warstwa wejściowa (Input layer): Odbiera dane wejściowe, które mogą być reprezentowane jako wektory liczbowych wartości np. zakodowane słowa w analizie tekstu. Następnie dane wejściowe są przekazywane do warstwy ukrytej.

Warstwa ukryta (Hidden layer): Przetwarza dane oraz zachowuję informacje o poprzednich krokach sekwencji dzięki swojej strukturze rekurencyjnej. Stan ukryty jest aktualizowany na każdym kroku czasowym, łącząc informacje z bieżącego wejścia z wcześniejszym kontekstem.

Warstwa wyjściowa (Output layer): Warstwa zwracająca ostateczny wynik działania sieci. Mogą to być np. rozpoznane kategorie słów w tekście, bądź wykryte na obrazie obiekty.

Główną i najważniejszą cechą RNN jest jej stan ukryty, który zapamiętuje pewne informacje o sekwencji. Oznacza to, że podczas podejmowania decyzji RNN bierze pod uwagę nie tylko bieżące dane wejściowe i wyjściowe, ale także informacje z wcześniejszych etapów przetwarzania. Najbardziej znanymi typami rekurencyjnych sieci neuronowych są klasyczne RNN, sieci LSTM (Long Short-Term Memory) oraz GRU (Gated Recurrent Unit).

W projekcie zostaną zastosowane sieci LSTM (Long Short-Term Memory). Są to sieci długiej pamięci krótkotrwałej. Klasyczne sieci RNN mają trudności z przetwarzaniem długich sekwencji z powodu problemu zanikającego gradientu. Oznacza to, że ich zdolność do zapamiętywania informacji z wcześniejszych kroków czasowych szybko maleje, co czyni je nieskutecznymi w zadaniach wymagających długoterminowego kontekstu. Natomiast sieci LSTM rozwiązują problem zanikającego gradientu, uwzględniają dane wejściowe z poprzednich kroków czasowych, jednocześnie modyfikując pamięć modelu oraz wagi wejściowe. Analizowane teksty mogą być niepełne, niegramatyczne, a emocje mogą być wyrażone w złożony sposób, natomiast LSTM, dzięki swojej zdolności do uchwycenia długoterminowych zależności, może zrozumieć pełny kontekst wypowiedzi.

2.1 Opis wybranego modelu i podejścia

W ramach projektu zastosowano model oparty na głębokiej sieci neuronowej z rekurencyjnymi warstwami LSTM. Zastosowanie dwukierunkowego LSTM (Bidirectional LSTM) umożliwi uwzględnienie zarówno kontekst z przeszłości (poprzednie słowa), jak i przyszłości (następne słowa). Dzięki temu sieć jeszcze lepiej interpretuje znaczenie tekstu, co dodatkowo zwiększa skuteczność w klasyfikacji stanów psychicznych. Model został zaprojektowany do analizy sekwencyjnych danych tekstowych w celu klasyfikacji wieloklasowej (określenie stanu psychicznego na podstawie danych tekstowych). Wybrana architektura i hiperparametry zostały zoptymalizowane w procesie eksperymentu.

2.2 Implementacja

W projekcie do implementacji modelu została wybrana biblioteka TensorFlow w połączeniu z biblioteką Keras. Środowisko sprawdza się do realizacji zadań związanych z głębokim uczeniem, takich jak klasyfikacja danych sekwencyjnych za pomocą sieci LSTM. TensorFlow to otwarta biblioteka programistyczna do uczenia maszynowego, która pozwala na budowanie i trenowanie różnych modeli ML. Dzięki prostej składni pozwala na szybkie definiowanie warstw, takich jak LSTM, Embedding czy Dropout. Umożliwia również optymalizację dzięki wbudowanym algorytmom, takim jak Adam. Framework umożliwia modyfikację architektury modelu dzięki pełnej kontroli nad poszczególnymi elementami.

Architektura modelu:

```
def build_model(lr, lstm_units, dropout_rate):
    model = Sequential([
        Embedding(input_dim=10000, output_dim=128),
        Bidirectional(LSTM(lstm_units, return_sequences=True)),
        Dropout(dropout_rate),
        Bidirectional(LSTM(lstm_units)),
        Dense(128, activation='relu'),
        Dropout(dropout_rate),
        Dense(len(label_encoder.classes_), activation='softmax')
    ])
    model.compile(optimizer=Adam(learning_rate=lr), loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    return model
```

1.Warstwa osadzeń (Embedding Layer)

Przekształca indeksy słów na wektory reprezentujące ich znaczenie w przestrzeni wielowymiarowej.

2. Pierwsza dwukierunkowa warstwa LSTM (Bidirectional LSTM- Hidden State)

LSTM Units to liczba jednostek w warstwie LSTM jest zmienna w zależności od testowanego hiperparametru, a return_sequences=True umożliwia przekazanie pełnej sekwencji wyjściowej do kolejnej warstwy. Dwukierunkowa warstwa LSTM analizująca dane sekwencyjne w obu kierunkach (przód i tył) i generująca pierwsze ukryte stany.

3. Warstwa Dropout

Mechanizm losowego "wyłączania" neuronów w celu zapobiegania nadmiernemu dopasowaniu modelu (overfitting).

4. Druga dwukierunkowa warstwa LSTM (Bidirectional LSTM - Hidden State)

5. Kolejna warstwa Dropout

6. Warstwa Dense (gęsto połączona)

Gęsto połączona warstwa z 128 neuronami, używająca funkcji aktywacji ReLU do przekształcania danych.

7. Warstwa wyjściowa (Output Layer)

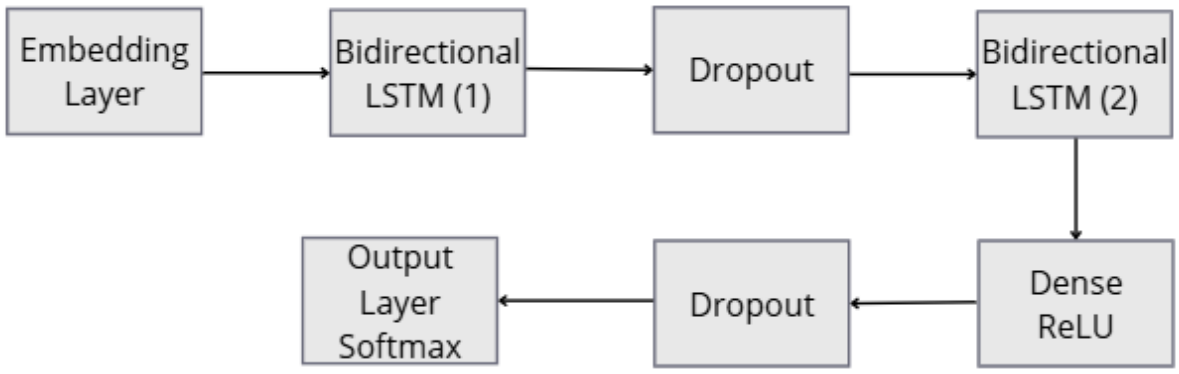
Warstwa wyjściowa generująca prawdopodobieństwa przynależności do poszczególnych klas za pomocą funkcji aktywacji Softmax.

8. Optymalizator i funkcja straty

- Optymalizator Adam, dostosowywany za pomocą różnych wartości learning_rate.

- Funkcja straty (sparse_categorical_crossentropy) odpowiednia dla problemów wieloklasowej klasyfikacji z etykietami numerycznymi.

Schemat przepływu danych przez modelu



Warstwa wyjściowa generująca prawdopodobieństwa przynależności do poszczególnych klas za pomocą funkcji aktywacji Softmax.

2.3 Testowanie

Zaprojektowany model pozwala na przeprowadzanie eksperymentów z różnymi parametrami, umożliwiając ocenę ich wpływu na jakość klasyfikacji.

Parametry do testowania:

```
learning_rates = [0.001, 0.0005, 0.0001]
lstm_units = [64, 128, 256]
dropout_rates = [0.2, 0.3, 0.4]
batch_sizes = [16, 32, 64]
epochs = [3, 5]
```

Model został przetestowany na różnych kombinacjach hiperparametrów, aby znaleźć optymalną konfigurację.

- **Learning rate** decyduje o wielkości kroku w procesie optymalizacji.

```
learning_rates = [0.001, 0.0005, 0.0001]
```

- **LSTM units** (liczba jednostek w warstwie LSTM): `lstm_units = [64, 128, 256]`

Większa liczba jednostek pozwala uchwycić bardziej złożone zależności, ale zwiększa koszt obliczeniowy.

-Dropout rate: `dropout_rates = [0.2, 0.3, 0.4]`

Dropout zmniejsza ryzyko overfittingu, regularizując model.

- **Batch size** (rozmiar partii danych): `batch_sizes = [16, 32, 64]`

Batch size określa liczbę próbek przetwarzanych podczas jednej aktualizacji wag modelu.

- **Epochs** (liczba epok): `epochs = [3, 5]`

Liczba epok definiuje, ile razy model przechodzi przez cały zestaw danych treningowych.

3. Analiza wyników

W ramach analizy przeprowadzono testowanie modelu LSTM do klasyfikacji tekstów pod kątem różnych stanów psychicznych. Celem było sprawdzenie, jak dobrze model radzi sobie z przewidywaniem siedmiu kategorii: Normal, Depression, Suicidal, Anxiety, Bipolar, Stress, Personality Disorder.

3.1 Wyniki

Po przetestowaniu różnych kombinacje parametrów osiągnięto 109 wyników. Wszystkie otrzymane wyniki zostały zapisane w pliku hyperparameter_results.csv, który został załączony do projektu. Poniżej zaprezentowano 5 najlepszych osiągniętych rezultatów:

learning_rate	lstm_units	dropout	batch_size	epchos	accuracy
0.001	**256**	**0.2**	**16**	**3**	**0.764661**
0.001	256	0.2	64	3	0.751945
0.0005	128	0.4	32	3	0.750996
0.0005	256	0.4	16	3	0.750996
0.001	64	0.2	32	3	0.748150

Wnioski z analizy wyników:

1.Wpływ Learning Rate: Wyniki pokazują, że dla LR = 0.001 i 0.0005 uzyskano wyższe testowe accuracy w porównaniu do LR = 0.0001, co sugeruje, że niższa wartość LR może prowadzić do niedouczenia modelu. Learning_rate = 0.001 wydaje się być najlepszym wyborem spośród przetestowanych wartości.

2. Wpływ liczby jednostek LSTM: Wartości LSTM Units = 128,256 dawały lepszy wynik accuracy niż 64. Dla LSTM = 256 model osiągnął najwyższą wartość accuracy (ok. 77%), co sugeruje, że większa liczba jednostek poprawia zdolność modelu do rozpoznawania wzorców w danych.

3. Wpływ Dropoutu: Dropout na poziomie 0.2 i 0.3 dawał lepsze wyniki niż 0.4, co sugeruje, że zbyt wysoki dropout może prowadzić do utraty istotnych informacji. Optymalna wartość dropout wydaje się wynosić 0.2 lub 0.3.

4. Wpływ Batch Size: Batch size = 16 i 32 często dawały lepsze wyniki accuracy niż 64, co wskazuje na to, że mniejsze partie mogą sprzyjać lepszemu dostosowaniu modelu. Dla batch size = 64 wyniki były bardziej niestabilne.

5. Wpływ liczby epok: W większości przypadków liczba epok wynosiła 3 dawała lepsze lub porównywalne wyniki do epok = 5, co sugeruje, że dłuższe trenowanie niekoniecznie poprawia wyniki accuracy. Może to sugerować lekkie przeuczenie modelu przy większej liczbie epok.

Podsumowując, najwyższe accuracy (ok. 77%) uzyskano dla:

LR = 0.001, LSTM Units = 256, Dropout = 0.3, Batch Size = 16, Epochs = 3

Wyniki sugerują, że model działa najlepiej przy umiarkowanym dropout, większej liczbie jednostek LSTM i stosunkowo małym batch size. W kontekście klasyfikacji tekstu 77% może być uznane za przyzwoity wynik, ale do optymalizacji. Może to świadczyć o tym, że model nadal ma problem z dobrze wyważonym rozpoznawaniem różnych klas. Możliwe rozwiązania to fine-tuning parametrów lub testowanie innych architektur.

3.2 Czas uczenia modelu

Proces trenowania modelu trwał 7872 minuty, co odpowiada około 131 godzinom. Taki długi czas treningu może wynikać z kilku czynników, takich jak złożoność modelu, rozmiar danych, wielkość batch size i liczba epok. Ponadto długi czas uczenia modelu może być spowodowane wydajnością sprzętową – użycie CPU zamiast GPU/TPU może znacząco spowolnić obliczenia. Długi czas treningu sugeruje, że w przyszłości można rozważyć optymalizację modelu, np.:

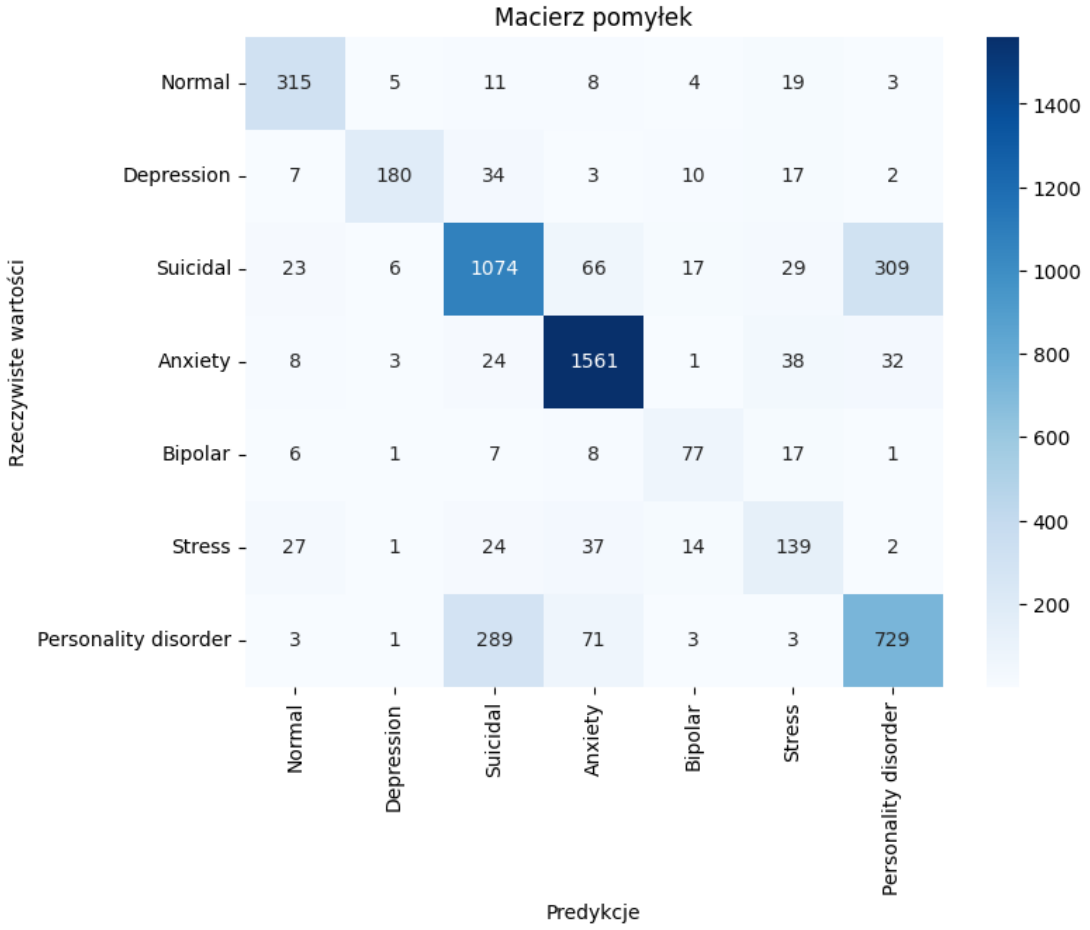
- Wykorzystanie mocniejszego sprzętu (GPU/TPU).

- Użycie transfer learning zamiast trenowania modelu od zera.

- Sprawdzenie, czy model nie jest zbyt skomplikowany w stosunku do problemu, co mogłoby prowadzić do nadmiernego dopasowania.

3.3 Wizualizacja wyników

W celu dokładnego zrozumienia wyników poniżej zostaną przedstawione wizualizacje. Najpierw przeanalizowany został wykres macierzy pomyłek, który przedstawia, jak model klasyfikował rzeczywiste etykiety w stosunku do przewidywanych. Odczytując wiersze macierzy, widzimy, ile przykładów danej klasy zostało poprawnie sklasyfikowanych oraz ile zostało błędnie przypisanych do innych kategorii.



Wykres 1 Macierz pomyłek

Analizując wykres najlepsza klasyfikacja została osiągnięta przez:

- Klasa Anxiety (Lęk) – 1561 przykładów zostało poprawnie sklasyfikowanych, co sugeruje, że model dobrze rozpoznaje ten stan emocjonalny.

- Klasa Suicidal (Myśli samobójcze)- poprawnie sklasyfikowane przypadki wynoszą 1074, ale zauważalne są również błędne przypisania do innych kategorii.

- Klasa Personality disorder (Zaburzenia osobowości) – 729 poprawnych klasyfikacji, jednak aż 289 przypadków błędnie przypisano do "Suicidal".

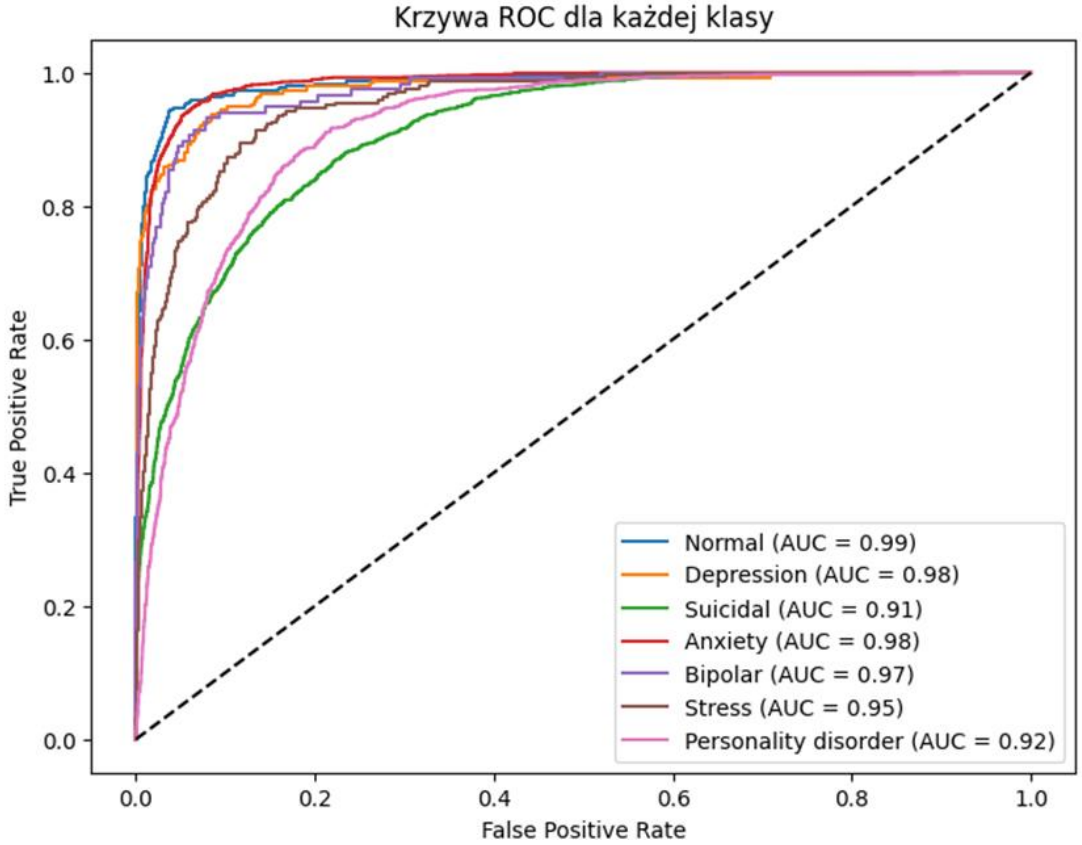
Natomiast, najczęściej zauważone błędy modelu to:

- Suicidal często mylone z Personality disorder – 309 przypadków zostało błędnie przypisanych do Personality disorder, co może wynikać z podobieństwa językowego między tymi kategoriami, również Personality disorder błędnie klasyfikowane jako Suicidal – 289 błędów w przeciwnym kierunku sugeruje, że model ma trudności z różnicowaniem tych dwóch klas.

- Stress i Anxiety są mylone – 38 przypadków stresu zostało zaklasyfikowanych jako lęk i odwrotnie, co jest zrozumiałe, ponieważ te stany są do siebie bliskie.

- Depression mylone z Suicidal – 34 przypadki depresji zostały sklasyfikowane jako Suicidal, co sugeruje, że model czasami nie rozróżnia tych dwóch stanów emocjonalnych.

Poniżej zaprezentowano wykres krzywej ROC dla każdej klasy. Krzywa ROC (Receiver Operating Characteristic) przedstawia zdolność modelu do rozróżniania między klasami na podstawie różnych progów decyzyjnych.



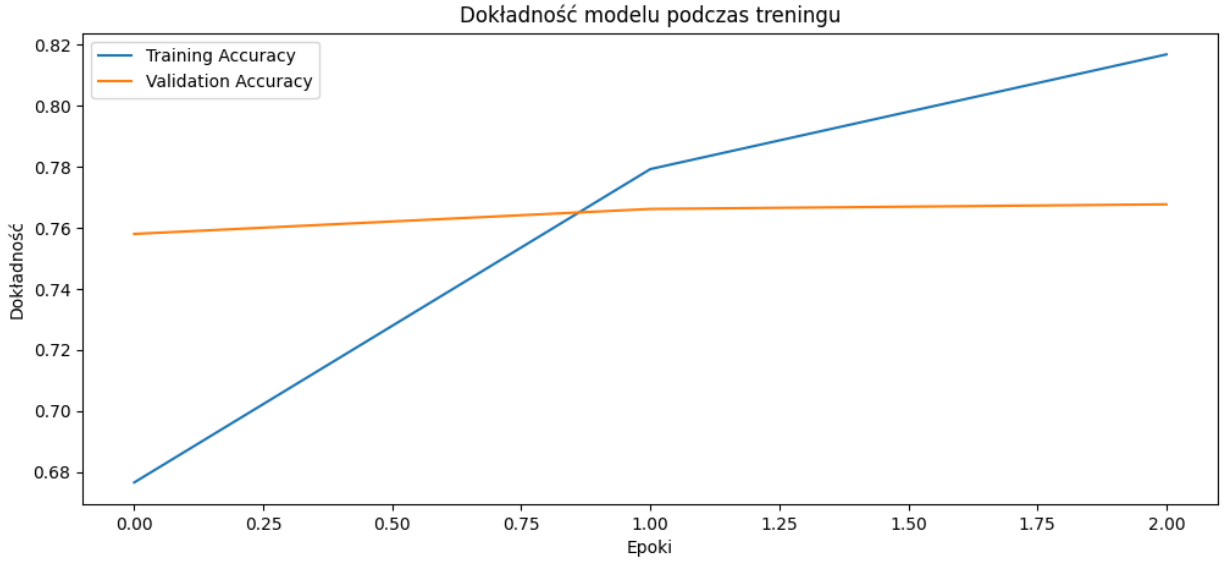
Wykres 2 Krzywa ROC dla każdej klasy

Wnioski:

Model generalnie dobrze klasyfikuje większość klas. Wartości AUC dla większości klas są wysokie, co oznacza, że model dobrze odróżnia te klasy od innych. Klasy "Normal" (AUC = 0.99), "Depression" (AUC = 0.98), "Anxiety" (AUC = 0.98) oraz "Bipolar" (AUC = 0.97) są dobrze rozpoznawane przez model.

Natomiast problematyczne klasy jak "Suicidal" ma AUC = 0.91, co wskazuje na pewne trudności w poprawnym rozróżnianiu tej klasy od innych. "Personality disorder" (AUC = 0.92) również ma niższą skuteczność, co może wynikać z podobieństwa tej klasy do innych zaburzeń psychicznych. Klasa "Stress" jest średnio rozpoznawana AUC = 0.95 dla klasy "Stress" wskazuje, że model radzi sobie z nią dobrze, ale istnieją przypadki błędnej klasyfikacji. Możliwe, że klasa "Stress" jest często mylony z klasami "Anxiety" lub "Depression", co jest naturalne, ponieważ objawy mogą się pokrywać.

Następnie wykonano wykres prezentujący dokładność modelu podczas treningu, który pokazuję jak zmieniała się dokładność (accuracy) modelu w trakcie kolejnych epok dla zbioru treningowego i walidacyjnego.

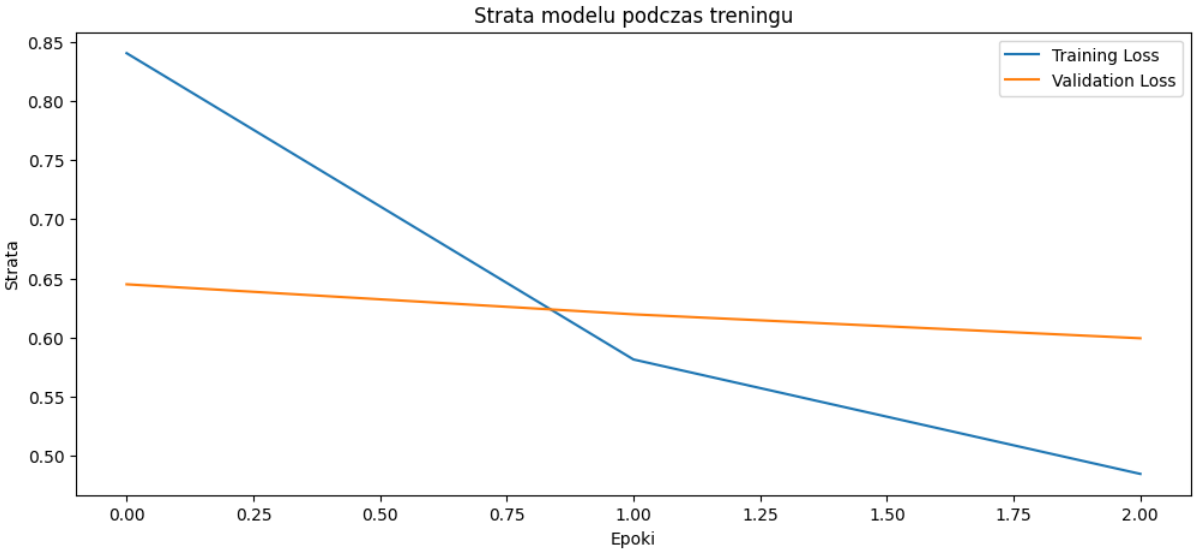


Wykres 3 Dokładność modelu podczas treningu

Wnioski:

Można zauważyć, że dokładność modelu na zbiorze treningowym systematycznie rośnie, co wskazuje na to, że model uczy się coraz lepiej klasyfikować dane treningowe. Dokładność walidacyjna również wzrasta, ale osiąga pewien stabilny poziom i przestaje rosnąć. Może to sugerować, że model osiąga już swój maksymalny potencjał na zbiorze walidacyjnym. Mała różnica między dokładnością treningową a walidacyjną sugeruje, że model nie jest nadmiernie dopasowany do danych treningowych, co jest pozytywnym wynikiem.

Kolejny wykres prezentuję wartość funkcji straty (loss), która mierzy, jak bardzo przewidywania modelu różnią się od rzeczywistych etykiet. Wykres straty modelu podczas treningu:



Wykres 4 Strata modelu podczas treningu

Wnioski:

Strata treningowa spada, co jest oczekiwanym zjawiskiem, ponieważ model coraz lepiej dopasowuje się do danych. Strata walidacyjna również nieznacznie maleje, ale jej spadek jest mniej dynamiczny niż w przypadku treningowej.

Podsumowując, model wykazuje poprawę zarówno w dokładności, jak i zmniejszaniu straty, co świadczy o skuteczności treningu. Model nie wydaje się być nadmiernie dopasowany, ale można również sprawdzić jego skuteczność na zupełnie nowych, nieznanych danych testowych.

4. Wnioski

Model oparty na dwukierunkowym LSTM (Bidirectional LSTM) osiągnął dokładność 77% w zadaniu analizy stanu psychicznego tekstów. Dobra skuteczność modelu wynika z jego zdolności do uwzględniania zarówno wcześniejszych, jak i późniejszych kontekstów w danych sekwencyjnych. Eksperymenty ujawniły kilka kluczowych czynników wpływających na wydajność modelu.

4.1.Wpływ parametrów na model

Zauważono, że większa liczba jednostek LSTM poprawiła zdolność modelu do uchwycenia złożonych wzorców w danych. Dropout na poziomie 0.2–0.3 okazał się skuteczny w zapobieganiu przeuczeniu, zapewniając równowagę między dokładnością a generalizacją modelu, a niższa wartość learning_rate może prowadzić do niedouczenia modelu. Natomiast

wpływ Batch size wskazuje na to, że mniejsze partie mogą powodować lepsze dostosowaniu modelu. W przypadku testowanie epok wyniki sugerują lekkie przeuczenie modelu przy większej liczbie epok.

4.2. Potencjalne usprawnienia

Model wykazał trudności w klasyfikacji niedoreprezentowanych klas, takich jak Bipolar czy Suicidal. W celu uzyskania lepszych wyników warto rozważyć:

- Wprowadzenie transformatorów. Transformery (np. BERT) to modele te, które dzięki mechanizmowi uwagi, mogą lepiej uchwycić złożone zależności w danych i poprawić skuteczność klasyfikacji.

- Rozszerzenie zbioru danych i integrację z innymi źródłami.

- Zastosowanie technik augmentacji tekstu, takich jak parafrazowanie, losowe usuwanie słów czy tłumaczenie maszynowe, aby sztucznie zwiększyć różnorodność danych.

- Fine-tuning transformatorów, czyli dostosowanie pretrenowanych modeli do specyficznego zadania analizy stanu psychicznego może przynieść jeszcze lepsze rezultaty.

- Eksperymenty z architekturą modelu, np. testowanie wariantów RNN, takich jak GRU, które są bardziej wydajne obliczeniowo lub dodanie warstw CNN w połączeniu z LSTM, aby uchwycić lokalne wzorce w danych sekwencyjnych. Modele sekwencyjne, takie jak LSTM, wymagają większej mocy obliczeniowej w porównaniu do prostszych modeli.

W celu rozwinięcia projektu można połączyć analizę tekstu z innymi źródłami danych, takimi jak analiza dźwięku (intonacji głosu), obrazów (ekspresji twarzy) czy danych biomedycznych, co może dostarczyć bardziej kompleksowej oceny stanu psychicznego.

Bibliografia:

https://bulldogjob.pl/readme/3-typy-rekurencyjnych-sieci-neuronowych

https://www.qtravel.ai/pl/blog/czym-sa-i-jakie-zastosowanie-maja-sieci-neuronowe/

https://www.unite.ai/pl/what-are-rnns-and-lstms-in-deep-learning/

https://bulldogjob.pl/readme/3-typy-rekurencyjnych-sieci-neuronowych

https://boringowl.io/blog/zagadnienia-typow-rekurencyjnych-sieci-neuronowych-podstawy-i-przyklady-stosowania

https://home.agh.edu.pl/~vlsi/AI/wstep1/sieci.html

https://ichi.pro/pl/20-popularnych-wskaznikow-uczenia-maszynowego-czesc-1-metryki-klasyfikacji-i-oceny-regresji-31490205459150

https://boringowl.io/tag/tensorflow

https://kpbc.umk.pl/Content/30079/PDF/00bc-psych.pdf

https://arxiv.org/abs/2207.01012

https://www.sas.com/pl_pl/insights/analytics/neural-networks.html