# Identyfikacja Gestu

## Grammatical Facial Expressions

Okres przygotowania

20/12/2024 - 23/01/2025

Przygotowane przez

Łukasz Podoba 14d
Bartosz Paszko 14d
Bartłomiej Szołkowski 15d

# Spis treści

# Cel

Celem badania było stworzenie modelu identyfikującego gesty używane podczas komunikacji brazylijskim językiem migowym (LIBRAS). Język ten, należy do grupy języków migowych wykorzystujących zarówno powszechnie stosowane gesty dłoni jak i GFE (grammatical facial expressions) czyli wyrazy twarzy stanowiące gramatyczne dopełnienie zdania przekazywanego przez gesty dłoni. Bez GFE komunikacja przy użyciu brazylijskiego języka migowego byłaby znacząco utrudniona i wprowadzała dużo dwuznaczności, ponieważ użytkownik języka nie byłby w stanie rozróżnić np. zdania twierdzącego od zdania pytającego. W tym badaniu skupiliśmy się wyłącznie na rozpoznawaniu właśnie tych wyrazów twarzy.

Grammatical Facial Expressions

# Opis danych

Zbiór danych wykorzystanych w badaniu został stworzony na podstawie 18 kilkuminutowych nagrań wykonanych przy użyciu sensora Microsoft Kinect. Każde z nagrań przedstawia twarz osoby, która pięciokrotnie wyraża 5 różnych zdań w brazylijskim języku migowym. Każde z tych zdań wymaga użycia konkretnego gramatycznego wyrazu twarzy (GFE). W sumie, na nagraniach przedstawione jest 9 różnych GFE przedstawianych przez 2 osoby.

Nagrania zostały manualnie ocenione przez specjalistów od brazylijskiego języka migowego, którzy dostarczyli informacji, w których klatkach nagrania jest wyrażana dane GFE. Natomiast, sensor Microsoft Kinect umożliwił wydobycie z każdej klatki nagrania informacji o położeniu 100 punktów reprezentujących pozycję i kształt głowy, ust, nosa, oczu oraz brwi. Każdy z punktów opisywany jest przez 3 współrzędne (x, y, z) - x, y opisujące położenie punktu na ekranie oraz z opisujący odległość od sensora Kinect.

Powstały w ten sposób zbiór danych składa się w sumie z 36 plików tekstowych. 18 plików zawierających współrzędne punktów dla każdej klatki nagrania oraz jej timestamp i 18 plików z binarną oceną ekspertów językowych na temat czy dana klatka nagrania przedstawia podany gramatyczny wyraz twarzy.

# Wstępne przetwarzanie

W celu uzupełnienia brakujących wartości z w danych, przeprowadzono analizę różnych metod imputacji wykorzystujących algorytm K-Nearest Neighbors (KNN). Problem imputacji był kluczowy, ponieważ brakujące wartości mogły wpływać na jakość uczenia modeli klasyfikacyjnych. Rozważono dwie główne implementacje metod KNN, różniące się sposobem przetwarzania danych wejściowych :

## fill_missing_zeros

**Opis działania**:

- Dla każdej kolumny z identyfikowane są wiersze, w których wartość z jest równa zero.
- Z danych referencyjnych (dataframe_ref) wybierane są wiersze, w których dana kolumna z ma wartości różne od zera. Te wiersze stanowią zbiór uczący dla modelu KNN.
- Model KNN trenuje się na pozostałych kolumnach (x, y oraz innych z), z wyjątkiem kolumny imputowanej.
- Po trenowaniu modelu na danych referencyjnych brakujące wartości z są przewidywane w oryginalnym DataFrame.

**Główne cechy**:

- Imputacja odbywa się na poziomie całej kolumny, niezależnie od innych kolumn.
- Wykorzystuje dane ze wszystkich klatek, w których z nie jest równe zero.

## fill_missing_z

**Opis działania**:

- Funkcja działa iteracyjnie, analizując każdy wiersz (klatkę filmu) osobno.
- Brakujące wartości z (z = 0) są uzupełniane na podstawie punktów x, y, z dostępnych w tej samej klatce.
- Jeśli funkcja napotka kolejny brakujący punkt Z, wykorzystuje już wcześniej imputowane wartości.

**Główne cechy**:

- Imputacja odbywa się na poziomie pojedynczego wiersza, co pozwala uwzględnić lokalny kontekst klatki.
- Wartości Z są uzupełniane iteracyjnie, co umożliwia dynamiczne wykorzystanie nowych danych do dalszej predykcji.

# Metodologia

**Przygotowanie danych testowych:**

- Aby przetestować skuteczność metod, wybrano wiersze, w których wszystkie wartości z były różne od zera. Na tej podstawie symulacyjnie wprowadzano braki, zastępując do 20% wartości z zerami w 85% wierszy.
- Rzeczywiste wartości z, które zostały zamienione na zera, zapisano, aby umożliwić późniejszą ocenę skuteczności metod.

**Ocena skuteczności:**

- Skuteczność metod oceniano na podstawie średniego błędu kwadratowego (MSE), porównując przewidywane wartości z z rzeczywistymi

**First Method** for k = 3 **1834.8**

**Second Method** for k = 3 **578.17**

**Skuteczność metod:**

- Metoda fill_missing_z osiągała znacznie niższe wartości MSE w porównaniu do fill_missing_zeros.
- Wyższa precyzja fill_missing_z wynika z uwzględnienia lokalnego kontekstu klatki i iteracyjnego uzupełniania braków.

# Metoda oceniania jakości modeli

W celu oceny jakości klasyfikatorów uwzględniono accuracy oraz analizę przy pomocy miary F1 wyliczanej osobno dla każdej z klas (reprezentujących poszczególne wyrazy twarzy/emocje).

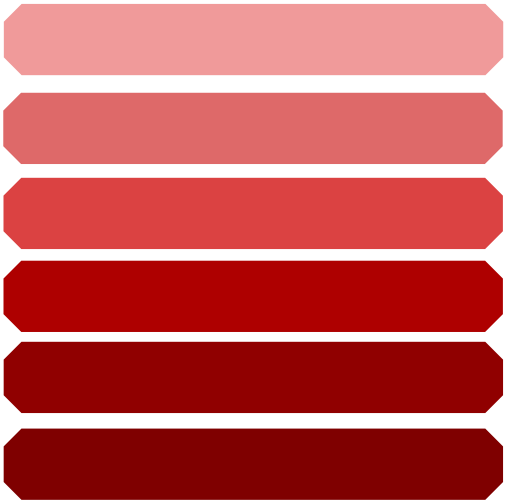
## Accuracy

Accuracy pozwala szybko sprawdzić, jaki procent próbek został prawidłowo sklasyfikowany w skali całego zbioru.

## F1-score

F1-score równoważy precision i recall w jednej statystyce, co jest szczególnie przydatne, gdy liczebność poszczególnych klas jest nierówna.

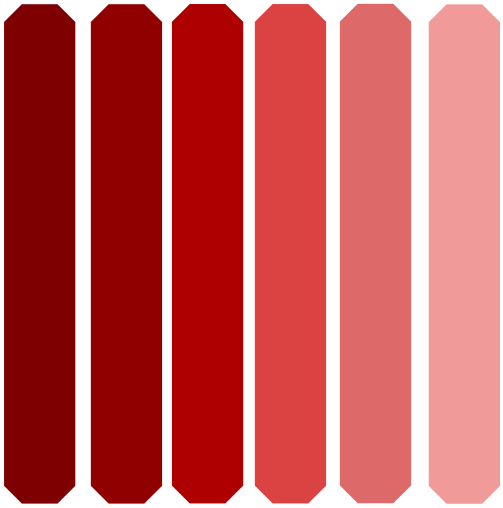
# Rekordy i Atrybuty



## LICZBA REKORDÓW

Zbiór obejmuje łącznie 27 936 wierszy (obserwacji). Każda obserwacja odpowiada jednej klatce nagrania.

**27 936**



## LICZBA ATRYBUTÓW

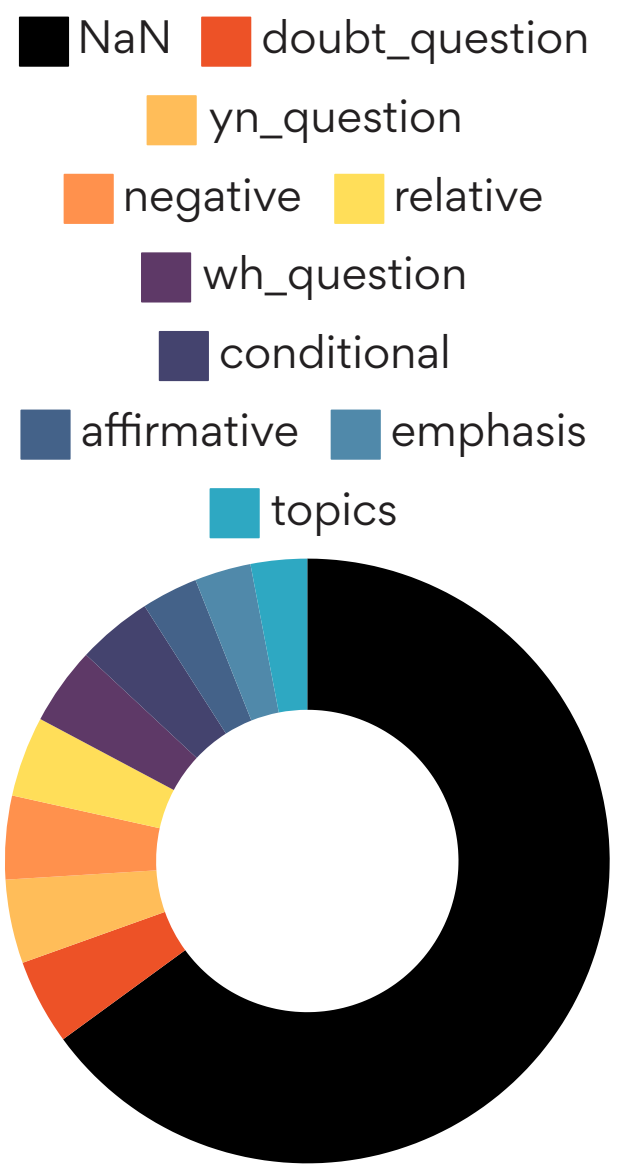
Każdy wiersz zawiera 300 kolumn reprezentujących współrzędne (x, y, z) różnych punktów na twarzy (w sumie 100 punktów). Współrzędne x i y opisują pozycję punktu na obrazie, a z wskazuje odległość tego punktu od sensora Kinect.

**300**

# Rozkład klas decyzyjnych

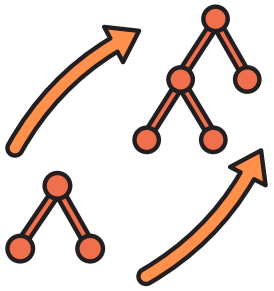
W ramach projektu analizowano różne wyrazy twarzy. Docelowo rozpoznaje się dziesięć możliwych stanów, z czego jedna klasa (“NaN”) oznacza brak konkretnego wyrazu twarzy. Rozkład poszczególnych etykiet w zbiorze wygląda następująco:

- NaN: 18 059 rekordów
- doubt_question: 1271
- yn_question: 1247
- negative: 1240
- relative: 1194
- wh_question: 1158
- conditional: 1137
- affirmative: 942
- emphasis: 861
- topics: 827




Z powyższych wartości wynika, że klasa “NaN” istotnie dominuje liczebnie względem pozostałych, co należy uwzględnić przy trenowaniu modeli.

# XGBoost



## Rozwiązanie problemu niezbalansowanych klas

- Stratyfikacja: Przy wczytywaniu danych zastosowano podział typu stratified split, aby równomiernie rozłożyć wszystkie klasy, zwłaszcza dominującą “NaN”.
- SMOTE: Przetestowano też metodę syntetycznego oversamplingu (SMOTE), ale uzyskane wyniki okazały się gorsze niż przy samej stratyfikacji.

## Analiza możliwości redukcji wymiarowości

- Sprawdzono, czy dane cechują się wystarczającą liniowością (co mogłoby pozwolić na zastosowanie PCA).
- Po wstępnej redukcji wymiarów wyniki modelu XGBoost pogorszyły się, dlatego zrezygnowano z PCA.

## Optymalizacja hiperparametrów

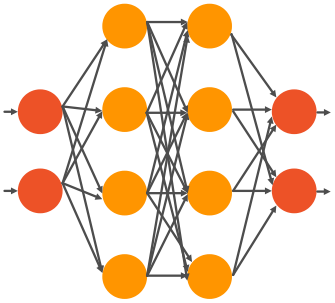
- Wykorzystano bibliotekę Optuna do przeprowadzenia 20 prób (triali) poszukiwania najlepszych ustawień hiperparametrów XGBoost.
- Najlepszy zestaw parametrów uzyskał accuracy na poziomie 0,92125 w trakcie walidacji.

## Ostateczny trening i wyniki

- Po zakończeniu optymalizacji dane ze zbioru treningowego i walidacyjnego zostały scalone, a model został ponownie wytrenowany na rozszerzonym zbiorze
- XGBoost osiągnął accuracy wynoszące 0,93, co potwierdziły szczegółowe miary f1-score dla poszczególnych klas. Wskaźnik f1-score kształtował się średnio na poziomie 0,9 dla wszystkich klas. Średnia ważona (weighted avg) wyniosła 0,93, co świadczy o stabilnej jakości klasyfikacji we wszystkich klasach. Osiągnięcie takich rezultatów było możliwe dzięki odpowiednim hiperparametrom, wyznaczonym dzięki Optunie.

Uzyskane wyniki potwierdzają skuteczność modelu XGBoost w rozpoznawaniu wyrazów twarzy w kontekście wieloklasowej klasyfikacji, przy czym stratyfikacja danych okazała się istotnym elementem zapewniającym wysoką jakość klasyfikacji.

# Sieć neuronowa



## Architektura

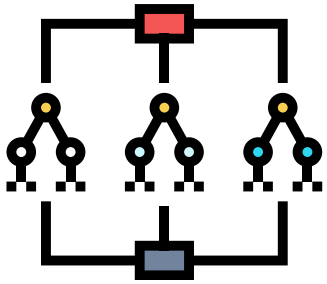
- Zaimplementowana sieć neuronowa składa się z 3 etapów. Etap pierwszy polega na połączeniu 3 ustandaryzowanych współrzędnych każdego punktu do jednego neuronu reprezentującego dany punkt w ukrytej przestrzeni. Następnie, w drugim etapie, neurony grupowane są na podstawie reprezentowanego fragmentu twarzy (lewe oko, prawe oko, nos, usta, kontur twarzy, itd.) i w zakresie tych grup przechodzą przez dwie ukryte warstwy liniowe. W ostatnim etapie, wyniki z poszczególnych grup są gęsto łączone ze sobą nawzajem i przechodzą przez jeszcze dwie warstwy liniowe.
- W celu uzyskania nieliniowości modelu po każdej ukrytej warstwie liniowej wyniki przechodzą przez funkcję aktywacji leaky relu z wyłączeniem ostatniej warstwy dla której stosowana jest funkcja softmax w celu uzyskania rozkładu prawdopodobieństwa dla każdej z klas. Pomiędzy warstwami liniowymi zastosowane także normalizację warstwową w celu poprawy stabilności uczenia modelu oraz warstwy dropout o parametrze 0,1 w celu regularyzacji i zmniejszenia nadmiernego dopasowywania się modelu do danych treningowych (overfitting).

## Trening

- Trening modelu zaplanowany był na 400 iteracji na zbiorze danych treningowych z możliwością wcześniejszego zakończenia w przypadku braku dalszej poprawy modelu (trening zakończył się po 217 iteracjach). Funkcja błędu zastosowana do uczenia modelu to entropia krzyżowa (cross entropy loss) wybrana ze względu na problem wieloklasowej klasyfikacji. Do optymalizacji parametrów użyta została metoda AdamW wykorzystująca momentum funkcji oraz wprowadzająca dodatkową regularyzację poprzez stosowanie zaniku wag (weight decay). Parametry zastosowane dla AdamW to początkowe tempo uczenia (learning rate) równe 0,001, wartości β odpowiednio 0,9 i 0,999 oraz wartość weight decay równą 0,01. Tempo uczenia jest zmniejszane ze współczynnikiem równym 0,2 w przypadku braku poprawy w minimalizacji funkcji błędu na danych walidacyjnych w ciągu 8 iteracji.

Sieć osiąga wynik dokładności na danych testowych równy 0,9302 oraz f-miarę o wartości przynajmniej 0,86 dla każdej z przewidywanych klas.

# Random Forest



## Rozwiązanie problemu niezbalansowanych klas

- SMOTE: W celu zbalansowania danych zastosowano technikę SMOTE (Synthetic Minority Over-sampling Technique). Dzięki niej liczebność każdej klasy została wyrównana w zbiorze treningowym, co pozwoliło modelowi na lepsze radzenie sobie z klasyfikacją rzadziej występujących klas i poprawiło miary jakości dla tych klas.

## Przygotowanie danych

- Początkowe podejście: Model został początkowo przetestowany na surowych danych, bez dodatkowych cech i skalowania. Choć osiągnięto przyzwoite wyniki, F1-miara dla niektórych klas, szczególnie mniej licznych, była niezadowalająca.
- Rozszerzenie cech: Aby poprawić wyniki, wprowadzono dodatkowe cechy, takie jak:
  - Długości (np. brwi, usta),
  - Odległości między kluczowymi punktami twarzy (np. nos-oczy, nos-usta),
  - Kąty (np. kąt między brwiami, kąt między ustami). Te dodatkowe cechy zwiększyły zdolność modelu do uchwycenia relacji przestrzennych na twarzy, co znacznie poprawiło jakość klasyfikacji.
- Skalowanie danych: Dane wejściowe zostały ustandaryzowane za pomocą StandardScaler, co przyczyniło się do zwiększenia stabilności i skuteczności modelu.

## Modelowanie

- Optymalizacja hiperparametrów:
  - Przeprowadzono random search w celu znalezienia najlepszych ustawień hiperparametrów dla Random Forest.
  - Testowano wpływ redukcji wymiarowości za pomocą PCA, jednak najlepsze wyniki osiągnięto bez jej stosowania. PCA w niektórych konfiguracjach obniżało wyniki.
- Feature Importance: Analiza ważności cech wskazała, że istotne były dodatkowe atrybuty, takie jak odległości (np. nos-oko), czy długość brwi. Pokazuje to, że rozszerzenie cech miało znaczenie dla sukcesu modelu.

Model osiągnął accuracy na poziomie 93.49% na danych testowych.
Klasy takie jak "None" i "doubt_question" osiągnęły najwyższą skuteczność z F1-score powyżej 0.94, a pozostałe również dały zadowalający wyniki na poziomie conajmniej 0.86.

# Podsumowanie

## Summary

Badanie dotyczyło rozpoznawania gramatycznych wyrazów twarzy (GFE) w brazylijskim języku migowym (LIBRAS) z wykorzystaniem 27 936 klatek nagrań z sensora Kinect, zawierających współrzędne stu punktów na twarzy. Zastosowano różne metody imputacji wartości brakujących (m.in. KNN iteracyjny) oraz techniki równoważenia klas (stratyfikacja, SMOTE). Najwyższą skuteczność – z dokładnością ok. 93% i f1-score przekraczającym 0,86 – uzyskały trzy modele: XGBoost (strojenie Optuną, bez SMOTE), sieć neuronowa (architektura wieloetapowa z warstwami liniowymi i leaky ReLU) oraz Random Forest (SMOTE, rozszerzone cechy i random search).

Brocelle

# Dziękujemy!

Dziękujemy za poświęcony czas i zapoznanie się z naszym raportem. Mamy nadzieję, że przedstawione metody i wyniki okażą się inspirujące oraz przydatne w dalszych pracach i badaniach.


s27478@pjwstk.edu.pl - Łukasz Podoba

s27471@pjwstk.edu.pl - Bartosz Paszko

s27541@pjwstk.edu.pl - Bartłomiej Szołkowski