

## Baza danych

Użyta w projekcie baza danych pochodzi ze strony Physionet, link: Sleep-EDF Database Expanded v1.0.0 (physionet.org). Cała baza zawiera aż 197 nagrań ze snu. Zapisy zawierają zapis sygnału EEG z elektrod Fpz-Cz, EEG Pz-Oz, a także zapisy sygnałów EOG, EMG brody i znaczniki zdarzeń. Udostępnione są też pliki, które posiadają zapisy pochodzące z oddychania i temperatury ciała.

Są także dostępne hypnogramy, które są zapisami wzorców snu w danym sygnale, zintegrowane z plikami PSG. Znaczniki zidentyfikowane prawidłowo, zostały sprawdzone przez techników zgodnie z podręcznikiem Rechtschaffen i Kales.

Zbiór danych jest podzielany na dwie części. W mojej projekcie użyłam jednego z nich, a dokładniej zestawu „sleep cassette”.

Wykorzystałam 101 odpowiednich według mnie nagrań. Zapisy pochodzą z lat 1987-1991, w badaniach brały udział osoby wieku 25-101 lat. Pacjenci nie przyjmowali żadnych leków, które mogłyby w jakikolwiek sposób zmienić zapisy. Sygnały EEG zostały poddane próbkowaniu o częstotliwości 100 Hz.

## Implementacja algorytmu detekcji faz snu

Praca nad algorytmem rozpoczęła się od zaimplementowania odpowiednich bibliotek, w tym mne i pyeeg, które posiadają szeroki zakres funkcji przydatnych do analizy sygnałów EEG.

```
import os
import mne
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import warnings
import matplotlib.pyplot as plt
import seaborn as sns
```

Kolejnym krokiem było określenie katalogu, w którym znajdują się dane przeznaczone do detekcji faz snu, czyli sygnały EEG pochodzące z wcześniej opisanej bazy danych. Zmienna *data\_directory* została zdefiniowana w celu wskazania tego katalogu.

```
data_directory = "C:/Users/48511/Desktop/data_folder1"
```

Pętla *for* przegląda pliki w tym katalogu, szukając plików z rozszerzeniem "PSG.edf". Dla każdego znalezionej pliku z danymi EEG, kod wczytuje ten plik przy użyciu biblioteki mne.

```
for file_name in os.listdir(data_directory):
    if file_name.endswith("PSG.edf"):
        eeg_file_path = os.path.join(data_directory, file_name)

        raw_train = mne.io.read_raw_edf(eeg_file_path, preload=True)
```

Następnie kod szuka odpowiadającego plikowi hypnogramu, wczytuje go i przetwarza, tworząc odpowiednie zdarzenia (events).

```
if os.path.exists(hypnogram_file_path):
    try:
        hypnogram = np.genfromtxt(hypnogram_file_path, dtype=int, invalid_raise=False)
    except Exception as e:
        print(f"Błąd podczas wczytywania hypnogramu dla {file_name}: {e}")
        continue

    annot = mne.read_annotations(hypnogram_file_path)
    annot.crop(annot[1]["onset"] - 30 * 60, annot[-2]["onset"] + 30 * 60)
    raw_train.set_annotations(annot, emit_warning=False)

    annotation_desc_2_event_id = {
        'Sleep stage R': -1,
        'Sleep stage W': 1,
        'Sleep stage 1': 2,
        'Sleep stage 2': 3,
        'Sleep stage 3': 4,
        'Sleep stage 4': 4
    }
    events_train, _ = mne.events_from_annotations(raw_train, event_id=annotation_desc_2_event_id, chunk_duration=30.0)
```

Znajduje się także funkcja odpowiedzialna za filtrowanie sygnału. Został użyty filtr pasmowo – przepustowy Butterwortha, który jest używany do implementacji filtru pasmowo-przepustowego, gdzie zmienna *l\_freq* określa dolną częstotliwość graniczną, a zmienna *h\_freq* określa górną częstotliwość graniczną.

```
raw_train.filter(l_freq=0.5, h_freq=30, method='iir', iir_params={'order': 5, 'ftype': 'butter', 'output': 'sos'})
```

Wartości te ustalają pasmo częstotliwości, które pozostanie po filtracji. Struktura danych *events* przechowuje informacje o zdarzeniach lub etykietach czasowych, które są związane z danymi EEG. Te zdarzenia mogą obejmować różne wydarzenia, takie jak zmiana stanu snu, szybki ruch gałek ocznych (REM), fazy snu (NREM), okresy aktywności, itp. W kontekście analizy sygnałów EEG, *events* są mapowane na konkretne wartości numeryczne, które identyfikują określone fazy snu. Kolejno są to:

'Sleep stage R': -1, to jest faza snu REM

'Sleep stage W': 1, to okres przejściowy między snem, a czuwaniem;

'Sleep stage 1': 2, to faza snu lekkiego, w której łatwo jest osobę przebudzić, a aktywność mózgu spada;

'Sleep stage 2': 3, to główna faza snu lekkiego, w tym stadium sen staje się głębszy

'Sleep stage 3': 4,

'Sleep stage 4': 4

Faza "sleep stage 3" i faza "sleep stage 4" to fazy snu głębokiego.

```
annotation_desc_2_event_id = {  
    'Sleep stage R': -1,  
    'Sleep stage W': 1,  
    'Sleep stage 1': 2,  
    'Sleep stage 2': 3,  
    'Sleep stage 3': 4,  
    'Sleep stage 4': 4  
}
```

Etapy snu 3 i 4 są traktowane jako jedna faza, obie są określone poprzez „4”, ponieważ mają podobne wzorce aktywności mózgu i są zwykle uważane za fazę snu głębokiego.

Te numeryczne identyfikatory faz snu są wykorzystywane jako etykiety dla danych EEG. Po stworzeniu struktury danych *events*, są one wykorzystywane do przetwarzania i analizy danych EEG w dalszych etapach algorytmu.

Następnie tworzone są epoki, jedna trwa 30 s. Stworzone są one po to, aby analizować sygnał EEG w krótkich okresach czasu, które są nazywane epokami. Podział sygnału na epoki jest istotny, ponieważ pozwala na analizę sygnału w kontekście różnych faz snu lub innych zdarzeń zdefiniowanych przez plik hypnogramu.

```
tmax = 30.0 - 1.0 / raw_train.info["sfreq"]  
epochs_train = mne.Epochs(  
    raw=raw_train,  
    events=events_train,  
    event_id=annotation_desc_2_event_id,  
    tmin=0.0,  
    tmax=tmax,  
    baseline=None,
```

Następnie, po utworzeniu epok, przeprowadzane jest obliczanie widmowej gęstości mocy (PSD) przy użyciu funkcji z biblioteki mne.

```
psds_selected_channels, freqs = mne.time_frequency.psd_array_welch(  
    epochs_train.get_data(), sfreq=epochs_train.info['sfreq'], fmin=0.5, fmax=30, n_fft=1000)
```

PSD jest miarą mocy sygnału w różnych częstotliwościach i stanowi kluczowy wskaźnik do analizy sygnałów EEG. Pozwala ona zobaczyć, jak moc sygnału rozkłada się w różnych zakresach częstotliwości, co może być użyteczne do identyfikacji charakterystycznych wzorców aktywności mózgowej związanych z różnymi fazami snu. Przed wyliczeniem PSD dla epok zostały także wybrane dwa kanały zainteresowania tj. 'EEG Fpz-Cz', 'EEG Pz-Oz'.

```
channels_of_interest = ['EEG Fpz-Cz', 'EEG Pz-Oz']
```

Wyniki PSD są przechowywane w zmiennej *psds\_selected\_channels*, a odpowiadające im częstotliwości są przechowywane w zmiennej *freqs*. Dane te są używane jako cechy

wyjściowe dla klasyfikatora SVM (Support Vector Machine), który jest używany do prognozowania faz snu na podstawie wzorców aktywności mózgowej.

```
X = psds_selected_channels.reshape(len(psds_selected_channels), -1)
y = np.concatenate([epochs_train.events[:, 2]])
```

Następnie następuje podział na zbiór treningowy i testowy, przy użyciu funkcji „train\_test\_split()” z biblioteki „sklearn”.

```
X_train, X_test, y_train, y_test = train_test_split(X_data, y_data, test_size=0.2, random_state=42)
```

Kolejny etap to stworzenie klasyfikatora „svm” z jądrem „rbf”. Jest to podstawowe jądro o wartościach dodatnich, najczęściej stosowane ze względu na dobre wyniki klasyfikacji[13]. Trenowanie na danych treningowych jest możliwe dzięki funkcji „fit()”. Wykorzystując wytrenowany klasyfikator, dokonuje się prognozy faz snu na danych testowych przy użyciu funkcji predict().

```
svm_classifier = SVC(C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0,
shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False,
max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=None)

svm_classifier.fit(X_train, y_train)

y_pred = svm_classifier.predict(X_test)
```

Dokładność klasyfikacji jest oceniana za pomocą parametru „accuracy” czyli dokładność.

```
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

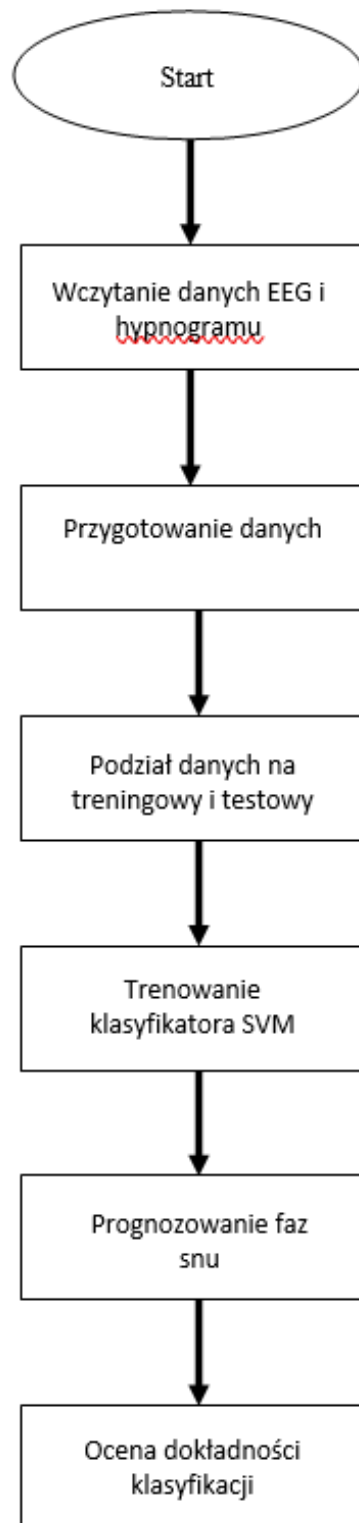
Wyniki klasyfikacji są przedstawione w następnym podrozdziale. Została stworzona również macierz pomyłek.

```
cm = confusion_matrix(y_test, y_pred)
```

Tablica pomyłek, czyli macierz pomyłek, służy do oceny skuteczności klasyfikacji binarnej, gdzie dane są oznaczone jako pozytywne lub negatywne. Klasyfikacja przypisuje im predykcję jako pozytywną lub negatywną. Występuje możliwość błędu, gdzie dane, które pierwotnie były pozytywne, zostają błędnie sklasyfikowane jako negatywne[10]. Kolejnym i ostatnim krokiem algorytmu jest wyliczenie, za pomocą funkcji *classification\_report* następujących parametrów: precyzji (ang. precision), czułości (ang. recall) i ich średniej harmonicznej czyli F1 score.

```
report = classification_report(y_test, y_pred)
print("Raport klasyfikacji:\n", report)
```

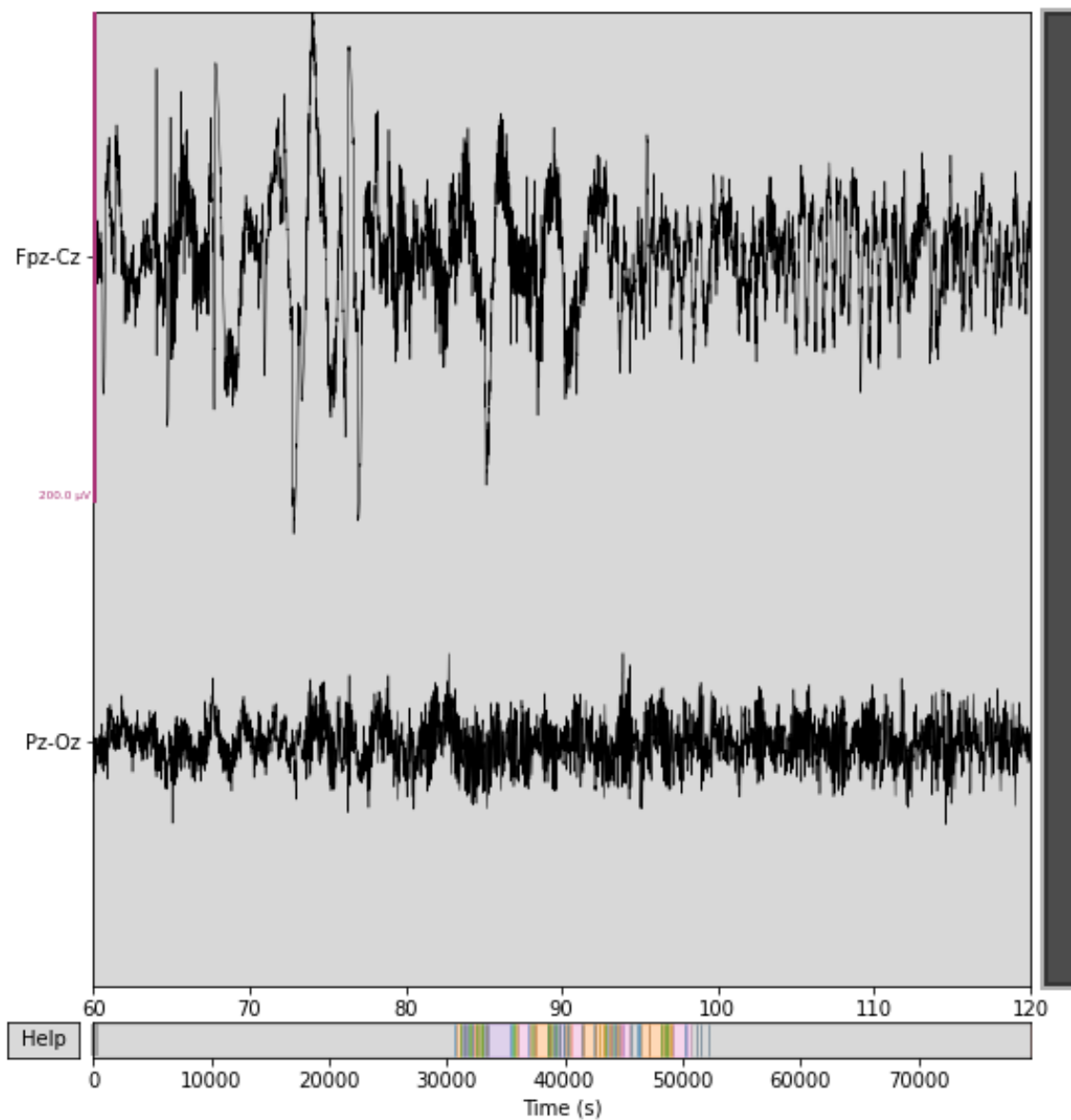




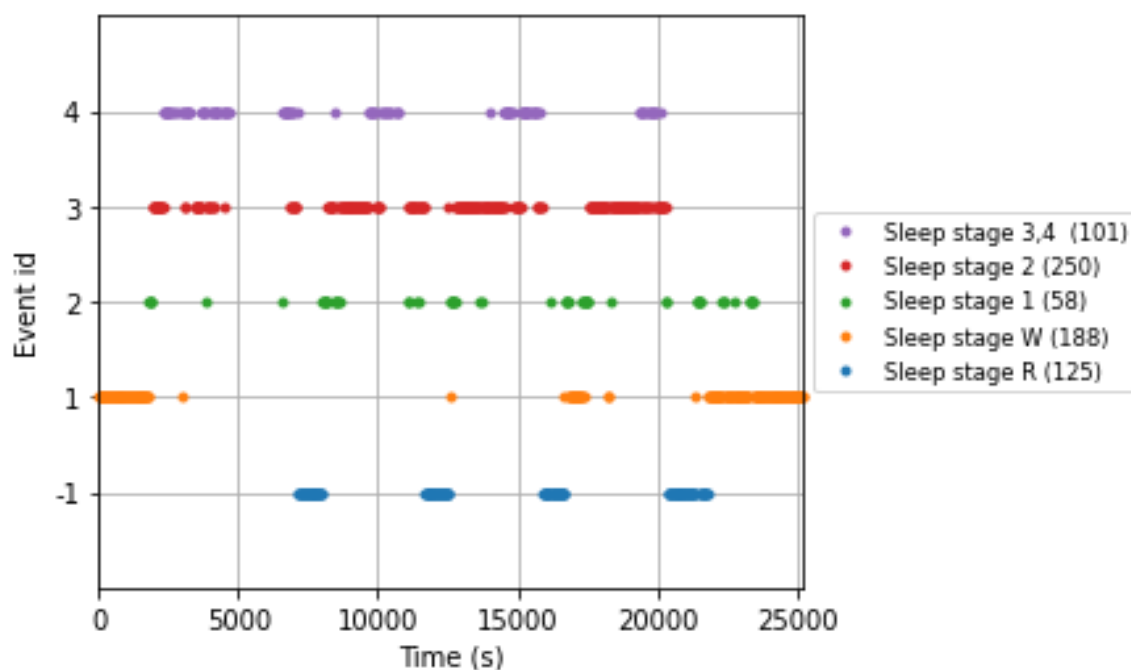
Rys.4 Schemat blokowy algorytmu detekcji faz snu.

## Wyniki

Na Rys.5 został przedstawiony wykres fal mózgowych dla jednego pacjenta

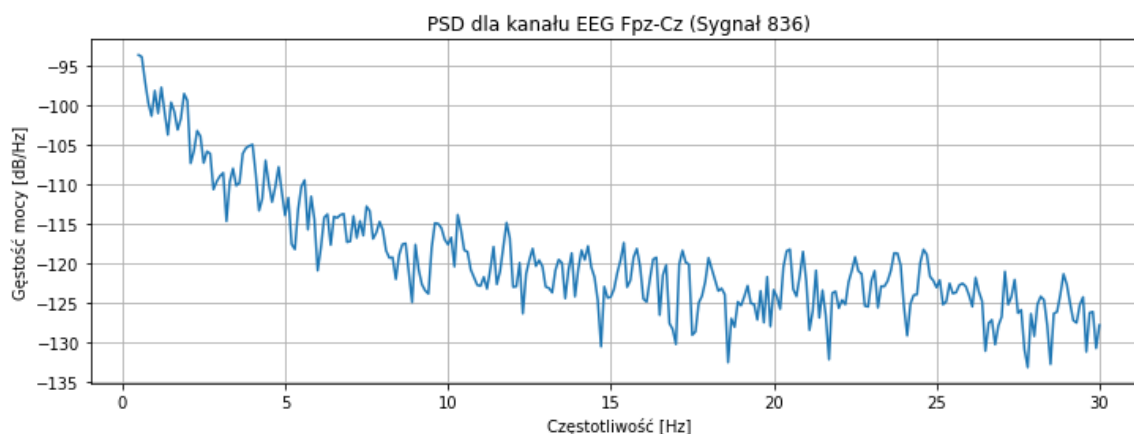


Rys.5. Wykres przedstawiający sygnał fal mózgowych dla jednego pacjenta z nagrania „SC4001E0-PSG.edf”, zostały ukazane dwa kanały EEG Fpz – Cz i EEG Pz-Oz. Jest to sygnał surowy, nieprzetworzony.



Rys.6 Wykres rozkładu etapów snu w czasie, dla pacjenta z nagrania „SC4001E0-PSG.edf”

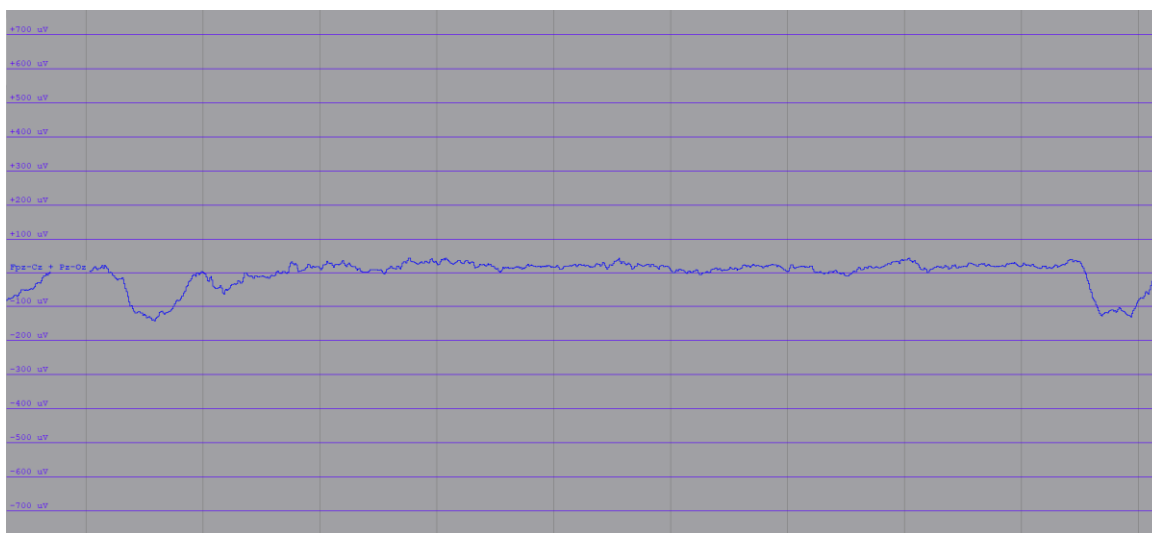
Na wykresie na Rys. 6 przedstawiono rozkład występowania różnych faz snu w czasie. Fazy snu są oznaczone różnymi kolorami i rozmieszczone wzdłuż osi czasu (oś X), co pozwala zaobserwować, kiedy i jak często występowały poszczególne etapy snu. Etapy 3 i 4 snu są połączone i przedstawione kolorem fioletowym, co jest standardową praktyką w wielu badaniach snu, gdzie te dwie fazy są łączone ze względu na ich podobieństwa. Liczby w nawiasach obok nazw faz wskazują na liczbę punktów danych odpowiadającą poszczególnym fazom, co może odzwierciedlać ich długość lub częstotliwość występowania w zarejestrowanym okresie.



Rys.7 Wykres PSD- gęstości widmowej mocy dla wybranego kanału EEG Fpz-Cz dla losowego sygnału, losowego pacjenta.



Wykres przedstawia gęstość mocy widmowej (PSD) dla kanału EEG Fpz-Cz, gdzie wartości mocy wyrażone są w decybelach na herc (dB/Hz) dla przedziału częstotliwości 0 - 30 Hz. Widoczne są fluktuacje mocy, które maleją wraz ze wzrostem częstotliwości, co jest typowe dla rozkładu mocy sygnałów EEG.



Rys.8 Wykres EEG dla dwóch kanałów, dla jednego pacjenta przedstawione w programie EDF Browser.

Na Rys. 8 przedstawiony jest wykres sygnału EEG (elektroencefalogram), który jest wizualną reprezentacją aktywności elektrycznej mózgu rejestrowanej przez specjalistyczny sprzęt. Ten konkretny wykres został wygenerowany za pomocą programu EDFbrowser.

Wynik końcowy to 70,7% dokładności klasyfikatora SVM.

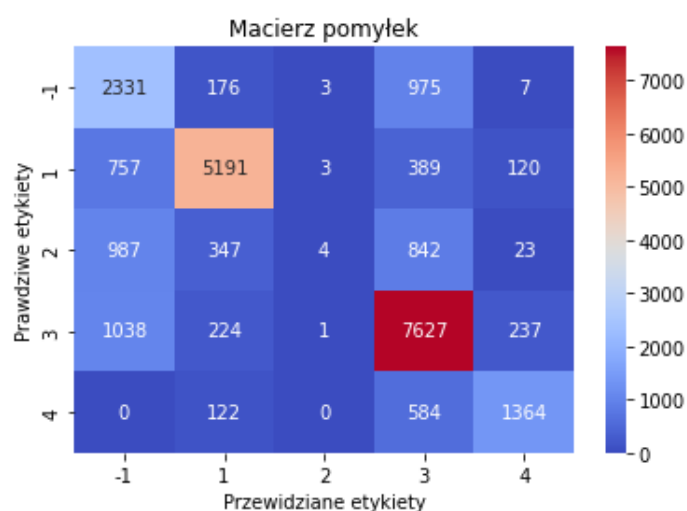
Accuracy: 0.7073055841041452

Kolejne ważne parametry, które zostały wyliczone to:

Raport klasyfikacji:				
	precision	recall	f1-score	
-1	0.46	0.67	0.54	
1	0.86	0.80	0.83	
2	0.36	0.00	0.00	
3	0.73	0.84	0.78	
4	0.78	0.66	0.71	
accuracy			0.71	
macro avg	0.64	0.59	0.57	
weighted avg	0.69	0.71	0.68	

Precision, zwana także precyzją, jest miarą, która ocenia skuteczność klasyfikatora w identyfikowaniu danej klasy. Określa ona, jaki odsetek elementów sklasyfikowanych jako należące do pewnej klasy faktycznie do niej należy. Recall, znana również jako czułość czy wskaźnik prawdziwie pozytywnych, mierzy zdolność modelu do wykrywania wszystkich rzeczywistych przypadków danej klasy. To wskaźnik, który pokazuje, jaka część wszystkich faktycznych pozytywnych przypadków została skutecznie zidentyfikowana przez klasyfikator. Miara F1 jest narzędziem wykorzystywanym do oceny wydajności klasyfikatorów, zwłaszcza gdy potrzebujemy zachować równowagę pomiędzy precision (precyzją) a recall (czułością)[20].

Na załączonym poniżej Rys. 9 jest przedstawiona macierz pomyłek. Mówi ona o stosunku przewidzianych etykiet do prawdziwych etykiet.



Rys. 9. Wyniki klasyfikacji, macierz pomyłek.

Jest to narzędzie do oceny wydajności klasyfikatora w zadaniu przewidywania etapów snu. Oś Y macierzy są prawdziwe etykiety (faktyczne etapy snu), podczas gdy oś X są etykiety przewidziane przez model klasyfikacyjny. Kolory w macierzy wskazują liczbę próbek, które zostały sklasyfikowane do danej kategorii, z cieplejszymi kolorami oznaczającymi wyższą liczbę próbek. Etykiety mają przypisane cyfry, które odpowiadają danym fazom snu. Kolejno to:

-1: 'Sleep Stage R', czyli faza snu REM

1: 'Sleep Stage W', to okres przejściowy między snem, a czuwaniem

2: 'Sleep Stage 1', odpowiada fazie N1 snu NREM, która jest etapem przejściowym między czuwaniem a snem

3: 'Sleep Stage 2', odpowiada fazie N2 snu NREM, etapowi lekkiego snu

4: 'Sleep Stage 3' oraz 'Sleep Stage 4', odpowiadają fazie N3 snu NREM, znanej również jako głęboki sen lub sen delta.

Na przekątnej macierzy, od lewego górnego rogu do prawego dolnego, znajdują się wartości odpowiadające poprawnie sklasyfikowanym próbkom. Te liczby są wyjątkowo wysokie w porównaniu do wartości poza przekątną, co wskazuje na dobrą wydajność klasyfikatora. Na przykład, największa wartość 7627 w dolnym prawym rogu przekątnej oznacza, że większość próbek etapu snu '3' została poprawnie zidentyfikowana. Jednakże, macierz pokazuje również istnienie pewnej liczby błędnych klasyfikacji, na przykład 176 przypadków faktycznego etapu '-1' zostało błędnie sklasyfikowanych jako etap '1'. Podobnie, 975 przypadków etapu '-1' zostało mylnie uznanych za etap '3', co może wskazywać na specyficzną trudność modelu w rozróżnianiu tych dwóch etapów. Wartości poza przekątną, znane jako błędy typu I (fałszywie pozytywne) i typu II (fałszywie negatywne), są stosunkowo niskie w porównaniu do wartości na przekątnej, co sugeruje, że model ma ogólnie wysoką precyzję i czułość.

Współczynniki te są istotne w kontekście monitorowania snu, ponieważ dokładna identyfikacja etapów snu jest kluczowa dla właściwej diagnozy i analizy jakości snu pacjenta. Z tablicy pomyłek można także wywnioskować, iż etap '2' był rzadko klasyfikowany. Wyniki zostaną omówione i zinterpretowane w kolejnym rozdziale pracy.

## Dyskusja wyników

Analiza sygnałów EEG w celu detekcji faz snu jest złożonym zadaniem, wymagającym skutecznych metod przetwarzania sygnałów oraz precyzyjnych klasyfikatorów. W przeprowadzonych badaniach uzyskano dokładność klasyfikatora na poziomie 70,7%. Powodem, dla którego wartość ta nie jest wyższa może być liczba wprowadzonych do klasyfikatora danych. Gdy liczba sygnałów została zredukowana o ponad połowę, dokładność spadła do zaledwie 46%. Fakt ten sugeruje, że liczba analizowanych danych ma istotne znaczenie. Na podstawie raportu klasyfikacji z miarami precyzji, czułości (recall) i wyniku F1 dla problemu klasyfikacji wieloklasowej, można wysunąć kilka wniosków. Widoczne są znaczne różnice w wynikach dla poszczególnych klas, co może sugerować nierównowagę w liczbie przykładów dla danych klas. Klasa oznaczona jako "-1" ma relatywnie niską precyzję (0.46) i wynik F1 (0.54), co wskazuje, że model ma problemy z dokładnym przewidywaniem tej klasy. Z kolei klasa 1 ma wysoką precyzję (0.86) i wynik F1 (0.83), co wskazuje na to, że model dobrze radzi sobie z przewidywaniem tej klasy. Klasa 2 wyróżnia się bardzo niskim wynikiem F1 (0.08), co jest wyjątkowo niskim wynikiem i może sugerować, że model prawie wcale nie jest w stanie poprawnie przewidzieć tej klasy. Średnia ważona wyników, która uwzględnia liczebność klas, osiąga wynik F1 na poziomie 0.68, co sugeruje, że model w sposób zrównoważony podchodzi do klasyfikacji, biorąc pod uwagę różnorodność i rozmiar danych. Wynik recall (czułości) na poziomie 0.71 w średniej ważonej wskazuje, że model stosunkowo dobrze identyfikuje wszystkie pozytywne przypadki, co jest zachęcające, zwłaszcza przy bardziej zróżnicowanych zestawach danych. Precyzja na poziomie 0.69 świadczy o umiejętności modelu w wyłapywaniu rzeczywiście pozytywnych przypadków wśród tych, które zaklasyfikował jako pozytywne, co jest obiecującym wskaźnikiem dla jego wiarygodności. W kodzie użyto domyślnych parametrów dla klasyfikatora SVM. W celu poprawienia wyniku być może byłoby skuteczne przeprowadzenie procesu optymalizacji parametrów, na przykład za pomocą walidacji krzyżowej, aby znaleźć najlepsze parametry klasyfikatora, które mogą poprawić jego wydajność. Przyczyną może być również brak mojego doświadczenia w wybieraniu i zastosowaniu filtrów stosowanych do przetwarzania wstępnego sygnałów, które mogą posiadać liczne artefakty i zakłócenia. Być może inny filtr lub ten sam, a o innych parametrach lepiej spełniałby swoje zadanie. Przeprowadzone zostało również doświadczenie z klasyfikatorem SVM i jądrem „linear”. Dokładność przy zastosowaniu tego

jądra spadła o około 7%. Finalnie zostało użyte jądro „rbf”, tak jak to było od początku zakładane. Uważam, że lepszego klasyfikatora niż Support Vector Machine do tego zadania, jakim jest detekcja faz snu w sygnałach EEG, nie znalazłabym. W wielu pozycjach naukowych właśnie ta metoda uczenia maszynowego również osiągała najlepsze wyniki, wynik z owej projekcji jest jednym z lepszych na tle innych artykułów naukowych, gdzie wyniki innych twórców oscylowały wokół niższych wartości niż 70%. Aczkolwiek korzystne byłoby również zastosowanie i porównanie innych modeli uczenia maszynowego, oprócz SVM, takich jak np. sieci neuronowe, drzewa decyzyjne czy algorytmy k-najbliższych sąsiadów. Należy także zwrócić uwagę na fakt, iż SVM najgorzej sklasyfikował etap snu „2”. W macierzy pomyłek możemy zauważyć, a dokładniej na jej przekątnej, iż dobrze sklasyfikowane zostały tylko 2 przypadki. Również bardzo rzadko przypisywana była mylnie etykieta „2” innym próbkom. Być może cechy charakteryzujące etap snu "2" są trudne do wyodrębnienia lub podobne do cech innych faz snu, a w efekcie klasyfikator może mieć trudności w ich poprawnej klasyfikacji. Moim zdaniem dobrym pomysłem byłoby również wyliczenie większej liczby cech sygnału EEG, tak aby klasyfikator miał większe możliwości w uczeniu się. Byłoby również wartościowe eksperymentowanie z różnymi zestawami cech oraz analiza ich wpływu na skuteczność klasyfikacji, aby znaleźć najbardziej znaczące cechy dla detekcji faz snu w sygnałach EEG. Początkowo algorytm zawierał również wyliczanie innych cech sygnału. Jednakże stworzenie z tych parametrów macierzy było problematyczne, a obliczenia trwały bardzo długo i wróciłam do wersji wyliczenia jedynie gęstości widmowej mocy. Warto także skupić się na zagadnieniu liczby kanałów. Klasyfikator SVM został zbudowany tylko na podstawie danych z dwóch kanałów EEG: 'EEG Fpz-Cz' i 'EEG Pz-Oz'. Wybór dwóch głównych kanałów EEG pomógł w skoncentrowaniu analizy na obszarach mózgu, które są istotne dla regulacji snu oraz zoptymalizował wydajność obliczeniową. Reasumując, analiza sygnałów EEG w celu detekcji faz snu jest wieloetapowym procesem, który wymaga uwzględnienia wielu czynników i podejść. Choć uzyskano obiecujący wynik dokładności klasyfikatora na poziomie 70,7%, istnieje wiele obszarów, które mogą być dalszym przedmiotem badań i optymalizacji. Poprzez dalsze badania i eksperymenty w tych obszarach możliwe jest poprawienie skuteczności klasyfikacji oraz lepsze zrozumienie procesów biologicznych związanych z fazami snu.