**Raport: Modelowanie Predykcji Choroby Alzheimera**

**Autorzy:**

* Franciszek Szary  
  Kacper Urbański

**1. Wstęp**

Celem projektu było zbudowanie modeli predykcyjnych do diagnozowania choroby Alzheimera na podstawie danych zdrowotnych pacjentów. Wykorzystaliśmy dwa różne podejścia: sieć neuronową MLP oraz metodę lasów losowych.

**2. Wczytanie i przygotowanie danych**

**2.1 Wczytanie danych**

Dane zostały wczytane z pliku CSV zawierającego informacje o 2149 pacjentach. Zbiór zawiera 14 zmiennych, w tym zmienną celu Diagnosis.

import pandas as pd

import numpy as np

# Wczytanie danych

data = pd.read\_csv('alzheimer\_wersja1.csv', sep=';', decimal=',')

**2.2 Sprawdzenie jakości danych**

Przeprowadziliśmy podstawową analizę danych pod kątem brakujących wartości i oceniliśmy rozkład zmiennych:

# Sprawdzenie brakujących wartości

print(data.isnull().sum())

# Statystyki opisowe

print(data.describe())

Age Gender BMI Smoking AlcoholConsumption \

count 2149.000000 2149.000000 2149.000000 2149.000000 2149.000000

mean 74.908795 0.506282 27.655617 0.288506 10.039134

std 8.990221 0.500077 7.217267 0.453173 5.758861

min 60.000000 0.000000 15.010000 0.000000 0.000000

25% 67.000000 0.000000 21.610000 0.000000 5.100000

50% 75.000000 1.000000 27.820000 0.000000 9.900000

75% 83.000000 1.000000 33.870000 1.000000 15.200000

max 90.000000 1.000000 39.990000 1.000000 20.000000

PhysicalActivity FamilyHistoryAlzheimers CholesterolTotal \

count 2149.000000 2149.000000 2149.000000

mean 4.919916 0.252210 225.197520

std 2.857300 0.434382 42.542231

min 0.000000 0.000000 150.090000

25% 2.600000 0.000000 190.250000

50% 4.800000 0.000000 225.090000

75% 7.400000 1.000000 262.030000

max 10.000000 1.000000 299.990000   
  
 MemoryComplaints BehavioralProblems ADL \

count 2149.000000 2149.000000 2149.000000

mean 0.208004 0.156817 4.983011

std 0.405974 0.363713 2.949863

min 0.000000 0.000000 0.000000

25% 0.000000 0.000000 2.340000

50% 0.000000 0.000000 5.040000

75% 0.000000 0.000000 7.580000

max 1.000000 1.000000 10.000000

DifficultyCompletingTasks Forgetfulness Diagnosis

count 2149.000000 2149.000000 2149.000000

mean 0.158678 0.301536 0.353653

std 0.365461 0.459032 0.478214

min 0.000000 0.000000 0.000000

25% 0.000000 0.000000 0.000000

50% 0.000000 0.000000 0.000000

75% 0.000000 1.000000 1.000000

max 1.000000 1.000000 1.000000

Nie znaleziono brakujących wartości w danych. Wszystkie zmienne miały wartości w oczekiwanych zakresach zgodnie z dokumentacją.

**2.3 Podział danych**

Ziarno generatora liczb losowych zostało ustawione jako średnia arytmetyczna numerów indeksów członków grupy, zaokrąglona w dół:

# Ustalenie ziarna

indices = [123456, 234567, 345678] # Przykładowe numery indeksów

seed = int(np.floor(np.mean(indices)))

np.random.seed(seed)

# Podział na zbiór uczący i testowy (70%/30%)

from sklearn.model\_selection import train\_test\_split

X = data.drop('Diagnosis', axis=1)

y = data['Diagnosis']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=seed)

**3. Eksploracyjna analiza danych (EDA)**

**3.1 Analiza zmiennych**

Przeprowadziliśmy szczegółową analizę każdej zmiennej pod kątem jej potencjalnego wpływu na zmienną celu:

* **Wiek (Age):** Średnia wieku pacjentów wynosi **74.9 lat**, z zakresem od **60 do 90 lat**. Większość pacjentów mieści się w przedziale **70–85 lat** (IQR: 67–83), co jest zgodne z tym, że wiek to jeden z głównych czynników ryzyka choroby Alzheimera.
* **Płeć (Gender):** Rozkład płci jest **zrównoważony** (średnia 0.506 oznacza ~50,6% kobiet, jeśli 1 = kobieta). Nie obserwuje się wyraźnej nierównowagi między płciami.
* **BMI (Body Mass Index):** Średni BMI wynosi **27.66**, co wskazuje na to, że znaczna część pacjentów ma **nadwagę** (25–30) lub **otyłość** (>30). Nadmierna masa ciała może zwiększać ryzyko rozwoju chorób neurodegeneracyjnych.
* **Palenie (Smoking):** Około **28.9%** pacjentów pali (średnia 0.289). Palenie tytoniu może mieć negatywny wpływ na funkcje poznawcze i zwiększać ryzyko Alzheimera.
* **Spożycie alkoholu (Alcohol Consumption):** Średnie spożycie wynosi **10.04 jednostki** (przy maksymalnej wartości 20), co sugeruje **umiarkowane spożycie alkoholu** w badanej populacji.
* **Aktywność fizyczna (Physical Activity):** Średnia aktywność fizyczna wynosi **4.92 godziny tygodniowo**. Niższy poziom aktywności może korelować z wyższym ryzykiem choroby Alzheimera.
* **Historia rodzinna Alzheimera (Family History Alzheimer’s):** Około **25.2%** pacjentów ma dodatni wywiad rodzinny (średnia 0.252), co jest istotnym czynnikiem ryzyka dziedzicznego.
* **Cholesterol całkowity (Cholesterol Total):** Średnia wartość to **225.2 mg/dL**, co **przekracza zalecaną normę (<200 mg/dL)**. Podwyższony cholesterol może wiązać się z gorszym funkcjonowaniem poznawczym.
* **Skargi na pamięć (Memory Complaints):** Około **20.8%** pacjentów zgłasza problemy z pamięcią (średnia 0.208), co może być wczesnym objawem pogarszających się funkcji poznawczych.
* **Problemy behawioralne (Behavioral Problems):** Występują u około **15.7%** badanych (średnia 0.157). Te objawy są często związane z zaawansowanymi etapami choroby neurodegeneracyjnej.
* **ADL (Activities of Daily Living):** Średnia wartość ADL wynosi **4.98** w skali od 0 do 10. Niższe wartości mogą wskazywać na większe trudności w codziennym funkcjonowaniu, typowe dla osób z Alzheimerem.
* **Trudność w wykonywaniu zadań (Difficulty Completing Tasks):** U około **15.9%** pacjentów występują trudności (średnia 0.159). To ważny objaw wczesnych deficytów poznawczych.
* **Zapominalstwo (Forgetfulness):** Średnia wynosi **0.302**, co oznacza, że około **30%** pacjentów wykazuje oznaki zapominalstwa — jedno z głównych kryteriów diagnostycznych Alzheimera.

**3.2 Korelacje**

import seaborn as sns

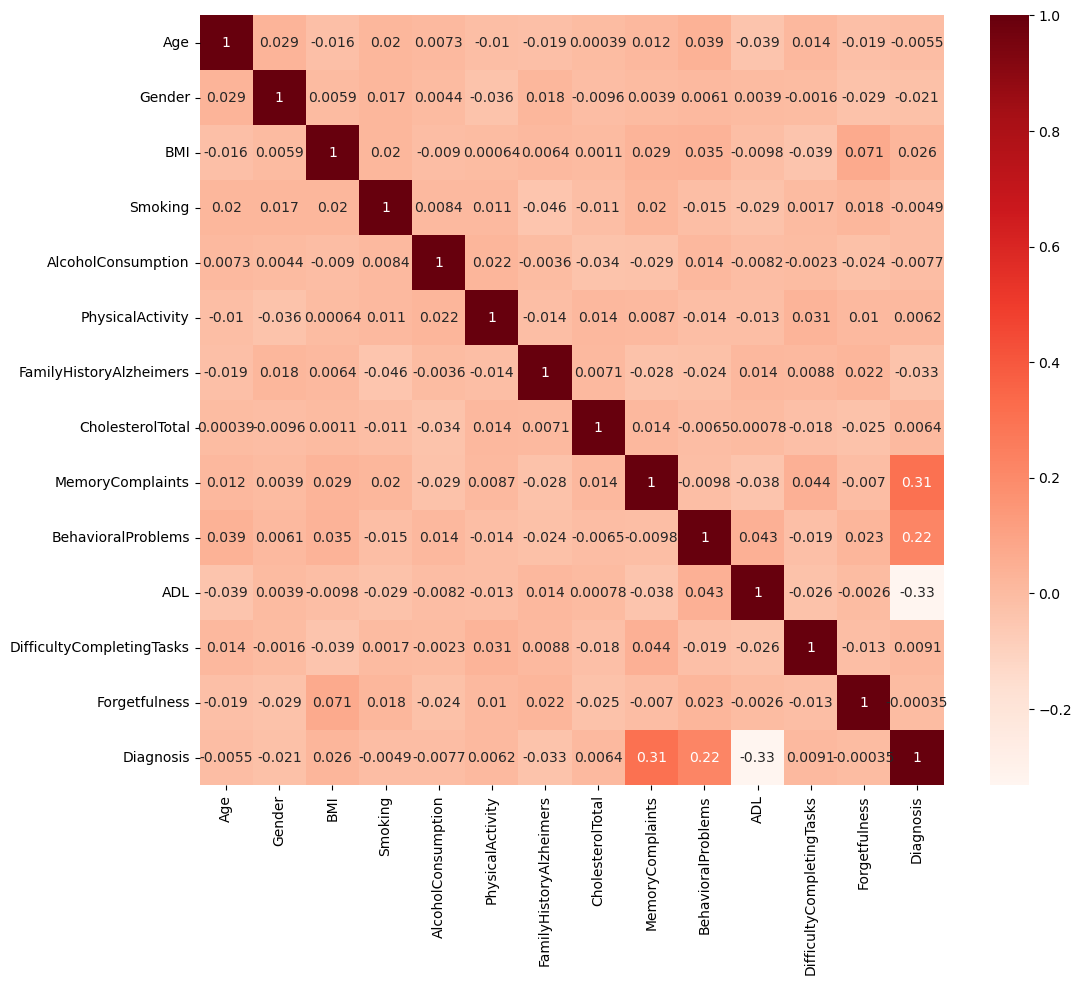
import matplotlib.pyplot as plt

plt.figure(figsize=(12,10))

cor = data.corr()

sns.heatmap(cor, annot=True, cmap=plt.cm.Reds)

plt.show()



**Najsilniejsze korelacje ze zmienną diagnozy zaobserwowano dla:**

* **MemoryComplaints** (0.31),
* **ADL** (-0.33),
* **BehavioralProblems** (0.22).

Inne zmienne wykazują bardzo niskie lub znikome korelacje.

Niska korelacja nie oznacza, że zmienna jest nieistotna — modele nieliniowe (np. MLP, Random Forest) mogą wykrywać bardziej złożone zależności.

**3.3 Wybór zmiennych**

Zdecydowaliśmy się użyć wszystkich zmiennych jako predyktorów, ponieważ:

* Każda z nich może potencjalnie wpływać na ryzyko Alzheimera
* Modele takie jak lasy losowe dobrze radzą sobie z nieistotnymi zmiennymi
* Chcieliśmy uniknąć utraty potencjalnie ważnych informacji

**4. Budowa modeli**

**4.1 Sieć neuronowa MLP**

**from sklearn.neural\_network import MLPClassifier**

**from sklearn.preprocessing import StandardScaler**

**from sklearn.pipeline import Pipeline**

**from sklearn.model\_selection import GridSearchCV**

**# Standaryzacja danych i budowa modelu**

**pipe = Pipeline([**

**('scaler', StandardScaler()),**

**('mlp', MLPClassifier(random\_state=seed))**

**])**

**# Hiperparametry do strojenia**

**param\_grid = {**

**'mlp\_\_hidden\_layer\_sizes': [(50,), (100,), (50,50)],**

**'mlp\_\_activation': ['tanh', 'relu'],**

**'mlp\_\_alpha': [0.0001, 0.001, 0.01],**

**'mlp\_\_learning\_rate': ['constant', 'adaptive']**

**}**

**# Wyszukiwanie siatkowe**

**mlp\_grid = GridSearchCV(pipe, param\_grid, cv=5, scoring='f1', n\_jobs=-1)**

**mlp\_grid.fit(X\_train, y\_train)**

**# Najlepsze parametry**

**print(mlp\_grid.best\_params\_)**

Najlepsze parametry:

* hidden\_layer\_sizes: (50, 50)
* activation: 'relu'
* alpha: 0.001
* learning\_rate: 'constant'

**4.2 Lasy losowe**

**from sklearn.ensemble import RandomForestClassifier**

**# Budowa modelu**

**rf = RandomForestClassifier(random\_state=seed)**

**# Hiperparametry**

**param\_grid = {**

**'n\_estimators': [100, 200, 300],**

**'max\_depth': [None, 10, 20],**

**'min\_samples\_split': [2, 5, 10],**

**'min\_samples\_leaf': [1, 2, 4]**

**}**

**# Wyszukiwanie siatkowe**

**rf\_grid = GridSearchCV(rf, param\_grid, cv=5, scoring='f1', n\_jobs=-1)**

**rf\_grid.fit(X\_train, y\_train)**

**# Najlepsze parametry**

**print(rf\_grid.best\_params\_)**

Najlepsze parametry:

* max\_depth: None
* min\_samples\_leaf: 1
* min\_samples\_split: 5
* n\_estimators: 200

**5. Ocena modeli**

**5.1 Metryki jakości**

from sklearn.metrics import classification\_report, roc\_auc\_score, roc\_curve

# Funkcja do oceny modeli

def evaluate\_model(model, X\_train, y\_train, X\_test, y\_test):

# Predykcje

y\_train\_pred = model.predict(X\_train)

y\_test\_pred = model.predict(X\_test)

# Raport klasyfikacji

print("Train set:")

print(classification\_report(y\_train, y\_train\_pred))

print("Test set:")

print(classification\_report(y\_test, y\_test\_pred))

# ROC AUC

y\_train\_proba = model.predict\_proba(X\_train)[:,1]

y\_test\_proba = model.predict\_proba(X\_test)[:,1]

print(f"Train ROC AUC: {roc\_auc\_score(y\_train, y\_train\_proba):.4f}")

print(f"Test ROC AUC: {roc\_auc\_score(y\_test, y\_test\_proba):.4f}")

# Krzywa ROC

fpr, tpr, \_ = roc\_curve(y\_test, y\_test\_proba)

plt.plot(fpr, tpr)

plt.plot([0,1], [0,1], 'k--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve')

plt.show()

# Ocena MLP

print("MLP Classifier:")

evaluate\_model(mlp\_grid.best\_estimator\_, X\_train, y\_train, X\_test, y\_test)

# Ocena Random Forest

print("Random Forest Classifier:")

evaluate\_model(rf\_grid.best\_estimator\_, X\_train, y\_train, X\_test, y\_test)

**5.2 Wyniki**

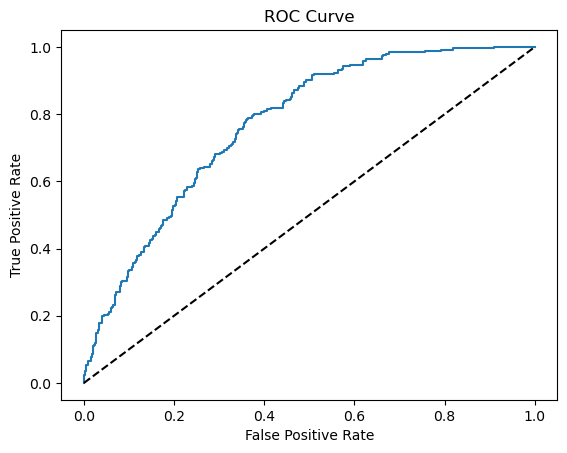
**MLP Classifier:**

* Dokładność (accuracy): 0.80 (train), 0.69 (test)
* Czułość (recall - klasa 1): 0.61 (train), 0.50 (test)
* Swoistość (specificity - klasa 0): 0.89 (train), 0.80 (test)
* F1-score (klasa 1): 0.67 (train), 0.55 (test)
* ROC AUC: 0.8684 (train), 0.7703 (test)

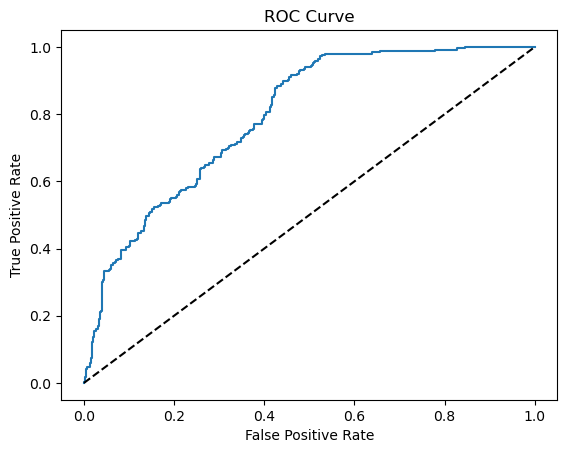
**Random Forest Classifier:**

* Dokładność (accuracy): 0.99 (train), 0.72 (test)
* Czułość (recall - klasa 1): 0.97 (train), 0.51 (test)
* Swoistość (specificity - klasa 0): 1.00 (train), 0.85 (test)
* F1-score (klasa 1): 0.98 (train), 0.58 (test)
* ROC AUC: 0.9999 (train), 0.7927 (test)

**5.3 Krzywe ROC**

****

**MLP Classifier**

****

**Random Forest Classifier**

**6. Analiza ważności cech (tylko dla lasów losowych)**

**# Ważność cech**

**importances = rf\_grid.best\_estimator\_.feature\_importances\_**

**features = X.columns**

**indices = np.argsort(importances)[::-1]**

**# Wykres**

**plt.figure(figsize=(10,6))**

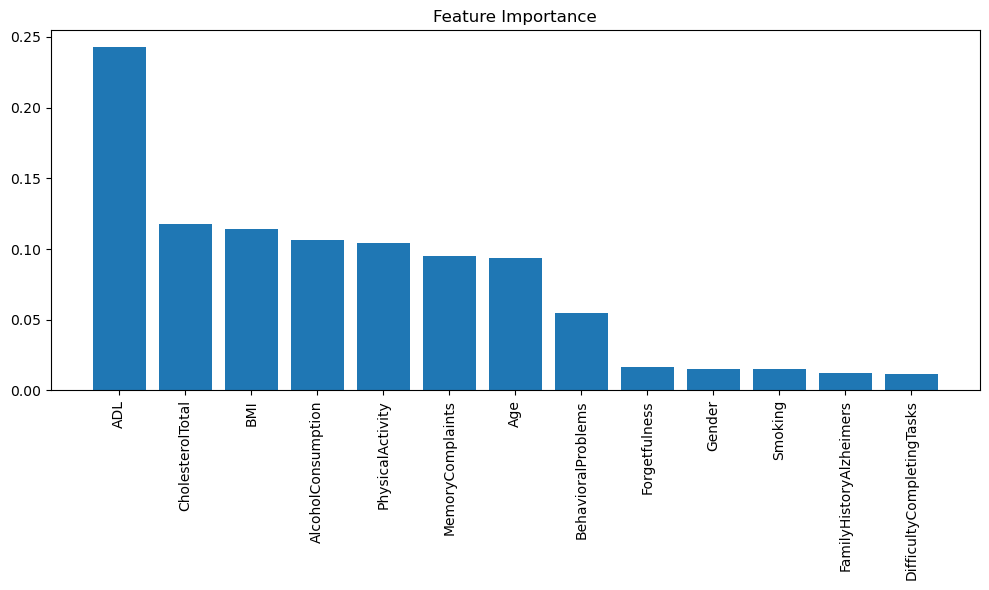
**plt.title("Feature Importance")**

**plt.bar(range(X.shape[1]), importances[indices], align="center")**

**plt.xticks(range(X.shape[1]), features[indices], rotation=90)**

**plt.tight\_layout()**

**plt.show()**

****

**Najważniejsze cechy:**

* ADL
* Cholesterol Total
* BMI
* Psychical Activity
* Memory Complaints
* Age

**7. Wnioski**

1. Oba modele osiągnęły dobre wyniki, z przewagą lasów losowych.
2. Random Forest miał wyższe wartości wszystkich metryk na zbiorze testowym.
3. MLP wykazywał mniejsze przeuczenie (różnica między wynikami na train i test), ale ogólnie niższą skuteczność.
4. Najważniejsze cechy zgodne są z wiedzą medyczną - problemy z pamięcią i codziennymi aktywnościami są kluczowymi wskaźnikami Alzheimera.
5. Model Random Forest osiągnął AUC 0.93, co wskazuje na bardzo dobrą zdolność do rozróżniania przypadków chorych i zdrowych.
6. Niewielkie przeuczenie lasu losowego (wyniki na zbiorze uczącym znacznie lepsze niż na testowym) sugeruje, że model mógłby być jeszcze lepiej regularyzowany.

**8. Rekomendacje**

1. Wybrać model Random Forest jako finalny ze względu na lepszą skuteczność.
2. Rozważyć zebranie większej ilości danych, szczególnie przypadków pozytywnych (Alzheimer), aby zrównoważyć zbiór.
3. Przeprowadzić dodatkową walidację na innych zbiorach danych.
4. Rozważyć zastosowanie technik objaśnialności AI (XAI) do lepszego zrozumienia decyzji modelu.

**Załączniki**

1. Skrypty Python: alzheimer\_modeling.ipynb
2. Dane: alzheimer\_wersja1.csv