

1. Klasyfikacja kwiatów Iris z wykorzystaniem uczenia maszynowego

Opis

Zbuduj model uczenia maszynowego, który klasyfikuje gatunki kwiatów Iris na podstawie cech morfologicznych (długość i szerokość płatków oraz działek).

Zakres

- Przygotowanie danych (podział na zbiory treningowy i testowy, sprawdzenie balansu klas).
- Eksploracja danych (wykresy, korelacje, analiza zależności między cechami).
- Porównanie wybranych modeli klasyfikacji (np. KNN, SVM, Random Forest).
- Walidacja krzyżowa i tuning hiperparametrów (Grid Search, Random Search).
- Ocena wyników: accuracy, precision, recall, F1-score, ROC-AUC.
- Interpretacja wyników i analiza najważniejszych cech.

Dane

[Iris Dataset – Kaggle](#)

2. Rozpoznawanie ubrań za pomocą sieci neuronowej

Opis

Zaimplementuj sieć neuronową w Pythonie do klasyfikacji obrazów z datasetów Fashion-MNIST.

Zakres

- Przygotowanie danych (normalizacja, podział na train/test).
- Budowa modelu (DNN lub CNN w TensorFlow/Keras).
- Trenowanie modelu z różnymi hiperparametrami.
- Ocena: accuracy, precision, recall, ROC-AUC.
- Eksperymenty z dropout, batch normalization, różnymi architekturami.
- Porównanie wyników między różnymi podejściami.

Dane

[Fashion MNIST – Kaggle](#)

3. System ekspertowy oparty na logice rozmytej

Opis

Zbuduj prosty system ekspertowy w Pythonie, który podejmuje decyzje na podstawie reguł rozmytych (np. ocena jakości powietrza, ryzyka zdrowotnego lub wyboru produktu).

Zakres

- Definicja zmiennych lingwistycznych i funkcji przynależności (np. trapezoidalne, gaussowskie).

- Stworzenie zestawu reguł rozmytych.
- Implementacja wnioskowania rozmytego.
- Dodanie prostego interfejsu użytkownika (np. konsola lub GUI).
- Testowanie systemu dla różnych scenariuszy.
- Analiza poprawności i optymalizacja reguł.

Dane

Możesz stworzyć własny mały zbiór danych (np. przykładowe cechy środowiska, zdrowia, produktów).

Dokumentacja

Opis systemu, reguł i wyników testów + kod źródłowy.

4. Optymalizacja trasy dostaw za pomocą algorytmu genetycznego

Opis

Zaimplementuj algorytm genetyczny optymalizujący trasę dostaw dla firmy kurierskiej lub sieci dronów dostawczych.

Zakres

- Kodowanie trasy (permutacje klientów/punktów).
- Inicjalizacja populacji i funkcja celu (koszt/czas).
- Operatory genetyczne: selekcja, krzyżowanie, mutacja.
- Ewolucja populacji przez wiele pokoleń.
- Wizualizacja optymalnej trasy (np. matplotlib).
- Eksperymenty z parametrami (rozmiar populacji, mutacje).

Dane

Możesz użyć własnej macierzy odległości (np. 10–15 punktów).

5. Klasyfikacja danych przy użyciu regresji logistycznej

Opis

Zbuduj model klasyfikacji binarnej z użyciem regresji logistycznej (np. przewidywanie przeżycia na Titanicu albo diagnozy raka piersi).

Zakres

- Przygotowanie danych (czyszczenie, normalizacja).
- Podział na train/test.
- Trenowanie regresji logistycznej.
- Walidacja krzyżowa, regularyzacja.
- Ocena: accuracy, precision, recall, F1-score, ROC-AUC.

Dane

<https://www.kaggle.com/c/titanic>

[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

6. Klasyfikacja tekstu z użyciem NLP

Opis

Zaimplementuj klasyfikator tekstu (np. spam/ham dla e-maili albo analiza recenzji).

Zakres

- Wczytanie zbioru danych.
- Wstępne przetwarzanie (tokenizacja, usuwanie stopwords, lematyzacja).
- Reprezentacja tekstu (Bag-of-Words, TF-IDF, embeddings).
- Klasyfikacja (Naive Bayes, SVM lub inne).
- Ewaluacja wyników i porównanie algorytmów.

Dane

<https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset>

<https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>

7. Rozpoznawanie obrazów za pomocą sieci CNN

Opis

Zaimplementuj model konwolucyjnej sieci neuronowej do rozpoznawania obrazów.

Zakres

- Przygotowanie danych (normalizacja, augmentacja).
- Architektura CNN (Conv, Pooling, Dense).
- Trenowanie i walidacja modelu.
- Dodanie dropout, batch normalization.
- Porównanie wyników dla różnych architektur.

Dane

<https://www.cs.toronto.edu/~kriz/cifar.html>

8. Klasteryzacja danych przy użyciu K-Means

Opis

Wykorzystaj algorytm K-Means do podziału danych na grupy (np. segmentacja klientów).

Zakres

- Przygotowanie zbioru danych.
- Implementacja algorytmu K-Means.
- Wizualizacja wyników (np. PCA, t-SNE).
- Eksperyment z DBSCAN jako alternatywą.

Dane

<https://www.kaggle.com/datasets/shwetabh123/mall-customers>

<https://archive.ics.uci.edu/ml/datasets/Wholesale+customers>

9. System rekomendacji z filtracją kolaboracyjną

Opis

Zbuduj prosty system rekomendacji produktów lub filmów.

Zakres

- Przygotowanie danych.
- Implementacja filtracji kolaboracyjnej (user-based, item-based).
- Ocena systemu (precision, recall, RMSE).
- Eksperymenty z macierzą faktoryzacji.

Dane

<https://grouplens.org/datasets/movielens/>

<https://www.kaggle.com/datasets/bittlingmayer/amazonreviews>

10. Predykcja wartości klienta (Customer Lifetime Value)

Opis

Przewidź wartość klienta w czasie (CLV) na podstawie danych zakupowych.

Zakres

- Przygotowanie danych (transakcje, dane demograficzne).
- Wybór modelu (regresja, drzewa decyzyjne, XGBoost).
- Ewaluacja modelu.
- Analiza cech mających największy wpływ.

Dane

<https://archive.ics.uci.edu/ml/datasets/online+retail>

<https://www.kaggle.com/datasets/rodsaldanha/arketing-customer-segmentation>

11. Detekcja anomalii przy użyciu autoenkoderów

Opis

Zaimplementuj autoenkoder do detekcji anomalii w danych (np. oszustwa finansowe, ataki sieciowe).

Zakres

- Przygotowanie danych.
- Budowa i trenowanie autoenkodera.
- Detekcja anomalii na podstawie błędu rekonstrukcji.
- Porównanie z tradycyjnymi metodami (Isolation Forest).

Dane

<https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

<http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>