

## Test

**1. Które narzędzie nie jest menedżerem pakietów dla języka python**

Phytonic Downloader

**2. Który z frameworków dostarcza wielu gotowych do użycia modeli uczenia maszynowego**

scikit-learn

**3. Która metryka nie służy do badania skuteczności działania modeli regresyjnych**

F1-score

**4. Które z poniższych określeń nie pasuje do terminu "Uczenie głębokie"**

To niższa, ograniczona wersja uczenia maszynowego

**5. Która z funkcji nie jest powszechnie wykorzystywana jako funkcja aktywacji w sieciach neuronowych**

Sin

**6. Jakiej metryki użyjesz do oceny modelu do szacowania ceny ropy na giełdzie towarowej?**

Współczynnika determinacji R<sup>2</sup>

**7. Gdybyś chciała/chciał sprawdzić i ocenić WSZYSTKIE przygotowane kombinacje parametrów badanego modelu, użyjesz funkcji SKLEARN**

Model\_selection.GridSearchCV(...)

**8. Funkcje biblioteki opencv pozwalają między innymi na**

podłączenie do różnych kamer

**9. Którego z modeli nie znajdziemy w podmodule neural\_network biblioteki SciKit Learn?**

Perceptron

**10. Której grupy funkcji nie znajdziemy w OpenCV**

wykrywanie dźwięków z różnych urządzeń PnP

## 1. Uczenie maszynowe (Machine Learning) i uczenie głębokie (Deep Learning)

- **Uczenie maszynowe (Machine Learning):**  
To dziedzina sztucznej inteligencji, która polega na tworzeniu modeli matematycznych zdolnych do samodzielnego uczenia się na podstawie danych, bez wyraźnego programowania każdej reguły. Główne podejścia:
  - **Uczenie nadzorowane (Supervised Learning):** Model uczy się na oznaczonych danych (np. obrazy z kategoriami).
  - **Uczenie nienadzorowane (Unsupervised Learning):** Model wykrywa wzorce w danych bez etykiet (np. grupowanie).
  - **Uczenie przez wzmacnianie (Reinforcement Learning):** Model uczy się na podstawie systemu nagród i kar.
- **Uczenie głębokie (Deep Learning):**  
To podzbiór uczenia maszynowego, który wykorzystuje głębokie sieci neuronowe do analizowania danych. Dzięki wielu warstwom przetwarzania, model może wykrywać złożone wzorce, co jest użyteczne w analizie obrazów, tekstu czy dźwięku.

## 2. Klasyfikacja a regresja

- **Klasyfikacja (Classification):**  
Polega na przypisywaniu danych wejściowych do jednej z określonych kategorii.  
Przykłady:
  - Rozpoznawanie zwierząt na zdjęciu (pies, kot, koń).
  - Klasyfikacja maili jako spam lub nie-spam.
- **Regresja (Regression):**  
Służy do przewidywania wartości ciągłych na podstawie danych wejściowych.  
Przykłady:
  - Prognozowanie cen domów.
  - Przewidywanie temperatury w danym dniu.

Różnica: Klasyfikacja odpowiada na pytanie "do której kategorii?", a regresja "jaka jest wartość?".

## 3. Uczenie maszynowe a inne rodzaje uczenia maszynowego

Różne rodzaje uczenia maszynowego obejmują:

- **Uczenie nadzorowane (Supervised Learning):** Model uczy się na oznaczonych danych.
- **Uczenie nienadzorowane (Unsupervised Learning):** Brak etykiet, model odkrywa struktury w danych (np. grupowanie).
- **Uczenie przez wzmacnianie (Reinforcement Learning):** Model podejmuje decyzje, otrzymując nagrody za dobre działania.
- **Uczenie półnadzorowane (Semi-Supervised Learning):** Łączy małą ilość oznaczonych danych z dużą ilością nieoznakowanych.
- **Transfer Learning:** Wykorzystanie wcześniej nauczonego modelu do rozwiązania nowego problemu.

## 4. Ewaluacja, metryki

Ewaluacja modelu to proces oceny jego wydajności na zbiorze testowym.  
Najpopularniejsze metryki to:

- **Accuracy:** Procent poprawnie sklasyfikowanych próbek.
- **Precision:** Dokładność przewidywań dla danej klasy (ile prawdziwych pozytywów wśród wszystkich pozytywów).
- **Recall (Czułość):** Jak dobrze model wykrywa wszystkie pozytywne przypadki.
- **F1-Score:** Harmoniczna średnia precyzji i czułości.
- **Mean Squared Error (MSE):** Używana w regresji, mierzy średnią różnicę między przewidywaniami a rzeczywistymi wartościami.
- **ROC AUC:** Miara wydajności modelu klasyfikacji binarnej.

## 5. Sieć neuronowa, las losowy (Random Forest), KNN

- **Sieć neuronowa:**  
Model inspirowany biologicznymi neuronami. Składa się z neuronów (węzłów), które są połączone w warstwy. Każdy neuron przetwarza dane, a model uczy się, dostosowując wagi połączeń.
- **Las losowy (Random Forest):**  
Algorytm oparty na wielu drzewach decyzyjnych. Każde drzewo uczy się na losowym podzbiorze danych, a finalna predykcja to uśredniony wynik (dla regresji) lub głosowanie (dla klasyfikacji). Jest odporny na przeuczenie i bardzo skuteczny.
- **KNN (K-Nearest Neighbors):**  
Algorytm klasyfikacji/regresji, który przypisuje nową próbkę do klasy na podstawie klas jej najbliższych sąsiadów (określonych za pomocą odległości, np. euklidesowej).

## 6. Perceptron (najlepiej wielowarstwowy), konwolucyjna sieć neuronowa

- **Perceptron:**  
Najprostszy model sieci neuronowej, symulujący pojedynczy neuron.  
Wielowarstwowy perceptron (MLP - Multi-Layer Perceptron) ma wiele warstw ukrytych i jest w stanie modelować bardziej skomplikowane zależności.
- **Konwolucyjna sieć neuronowa (CNN):**  
Używana głównie w analizie obrazów. Składa się z:
  - **Warstw konwolucyjnych:** Filtry wyodrębniające cechy obrazu (np. krawędzie, tekstury).
  - **Warstw aktywacji:** Funkcje aktywacji (np. ReLU) wprowadzają nieliniowość.
  - **Warstw poolingowych:** Redukują wymiary danych, zachowując istotne cechy.Rodzaje funkcji aktywacji:
  - **ReLU (Rectified Linear Unit):**  $\max(0, x)$ .
  - **Sigmoid:** Wartość w zakresie (0,1).
  - **Tanh:** Wartość w zakresie (-1,1).

## 7. Sklearn - podstawowe elementy

Scikit-learn to biblioteka Pythona do uczenia maszynowego. Główne elementy:

- **Modele:** Klasyfikacja, regresja (np. SVM, las losowy).
- **Przetwarzanie danych:** Skalowanie, kodowanie kategorii (`StandardScaler`, `OneHotEncoder`).
- **Podział danych:** `train_test_split` do dzielenia na zbiór treningowy i testowy.
- **Pipeline:** Łączenie przetwarzania danych i modelowania w jedną strukturę.

## 8. Pytorch - podstawowe elementy

PyTorch to biblioteka do głębokiego uczenia. Kluczowe elementy:

- **Tensory:** Podobne do tablic NumPy, ale z obsługą GPU (`torch.Tensor`).
- **Autograd:** Automatyczne obliczanie gradientów (`torch.autograd`).
- **Modele:** Definiowanie sieci neuronowych za pomocą `torch.nn` (np. `nn.Linear`, `nn.ReLU`).
- **Trening:** Optymalizatory (`torch.optim`) i funkcje strat (`torch.nn.CrossEntropyLoss`).
- **Obsługa GPU:** Wykorzystanie akceleracji GPU (`model.to(device)`).

## 9. OpenCV - podstawowe elementy

OpenCV to biblioteka do przetwarzania obrazów i wideo. Podstawowe elementy:

- **Wczytywanie obrazów:** `cv2.imread()` i wyświetlanie `cv2.imshow()`.
- **Operacje na obrazach:** Konwersja na odcienie szarości (`cv2.cvtColor()`), zmiana rozmiaru (`cv2.resize()`), progowanie (`cv2.threshold()`).
- **Wykrywanie krawędzi:** Algorytm Canny'ego (`cv2.Canny()`).
- **Wykrywanie obiektów:** Klasyfikatory Haar Cascade, wykrywanie twarzy.

## Rozszerzone notatki do pytań egzaminacyjnych

### 1. Menedżery pakietów dla Pythona

- **Python Package Index (PyPI):** Repozytorium zawierające tysiące bibliotek Pythona, ale samo w sobie nie jest menedżerem pakietów.
- **PIP:** Standardowy menedżer pakietów dla Pythona, który pobiera i instaluje biblioteki z PyPI.
- **Anaconda:** Dystrybucja Pythona z własnym menedżerem pakietów `conda`.
- **Virtualenv:** Narzędzie do tworzenia izolowanych środowisk Pythonowych.
- **Phytonic Downloader:** Nie istnieje jako narzędzie, co może być pułapką egzaminacyjną.

### 2. Framework z gotowymi modelami uczenia maszynowego

- **Scikit-learn:** Biblioteka dostarczająca gotowe modele klasyfikacji, regresji, klasteryzacji oraz narzędzia do przetwarzania danych (np. SVM, RandomForest, LinearRegression).
- **Pandas, NumPy, SciPy:** Narzędzia wspierające analizę danych i operacje numeryczne, ale nie oferują gotowych modeli.

### 3. Metryki do oceny modeli

- **F1-score:** Nie stosuje się go w regresji. To miara dla klasyfikacji, łącząca precyzję i czułość.

- **R<sup>2</sup>, RMSE, MAPE, MAE:** Typowe metryki regresyjne.
  - **R<sup>2</sup> (R-squared):** Mierzy, jak dobrze model wyjaśnia zmienność danych.
  - **RMSE (Root Mean Squared Error):** Pierwiastek średniego błędu kwadratowego.
  - **MAE (Mean Absolute Error):** Średnia wartość błędu bezwzględnego.
  - **MAPE (Mean Absolute Percentage Error):** Procentowy błąd przewidywań.

#### 4. Funkcje aktywacji w sieciach neuronowych

- **ReLU:** Najczęściej stosowana funkcja aktywacji ( $\max(0, x)$ ).
- **Sigmoid:** Ogranicza wartości do przedziału (0, 1).
- **Tanh:** Działa w zakresie (-1, 1).
- **Liniowa:** Używana w warstwach wyjściowych dla regresji.
- **Sin:** Nie jest powszechnie stosowana jako funkcja aktywacji.

#### 5. Uczenie głębokie a uczenie maszynowe

- **Uczenie głębokie (Deep Learning):**
  - Korzysta z głębokich sieci neuronowych.
  - Wymaga dużej mocy obliczeniowej, często wykorzystując GPU.
  - Analizuje dane w sposób automatyczny, odkrywając złożone wzorce.
  - **Fałszywe stwierdzenie:** „to niższa, ograniczona wersja uczenia maszynowego” – głębokie uczenie jest bardziej zaawansowane.

#### 6. Ewaluacja modeli w Scikit-learn

- **GridSearchCV:** Przegląd wszystkich możliwych kombinacji parametrów.
- **RandomizedSearchCV:** Losowe próbkowanie kombinacji parametrów.
- **Train\_test\_split:** Dzieli dane na zestawy treningowe i testowe.
- **Cross\_val\_predict:** Używany do przewidywania w walidacji krzyżowej.

#### 7. Funkcje OpenCV

- **Typowe zastosowania:**
  - Wykrywanie twarzy, punktów charakterystycznych.
  - Kaskadowe klasyfikatory (np. Haar Cascade do oczu, twarzy).
  - Algorytmy poprawiania obrazu.
  - Dopasowanie wzorców na obrazach.
- **Funkcje nienależące:** Wykrywanie dźwięków czy sterowanie drukarkami PnP.

#### 8. Modele w SciKit-learn

- **Podmoduł `neural_network` zawiera:**
  - Perceptron.
  - MLPClassifier (klasyfikacja).
  - MLPRegressor (regresja).
- **Nie ma modelu:** "Multi-layer Perceptron classifier" jako osobnego terminu (jest to inne określenie MLPClassifier).