

Deep Learning, Text Mining i XAI w Pythonie

Piotr Ćwiakowski

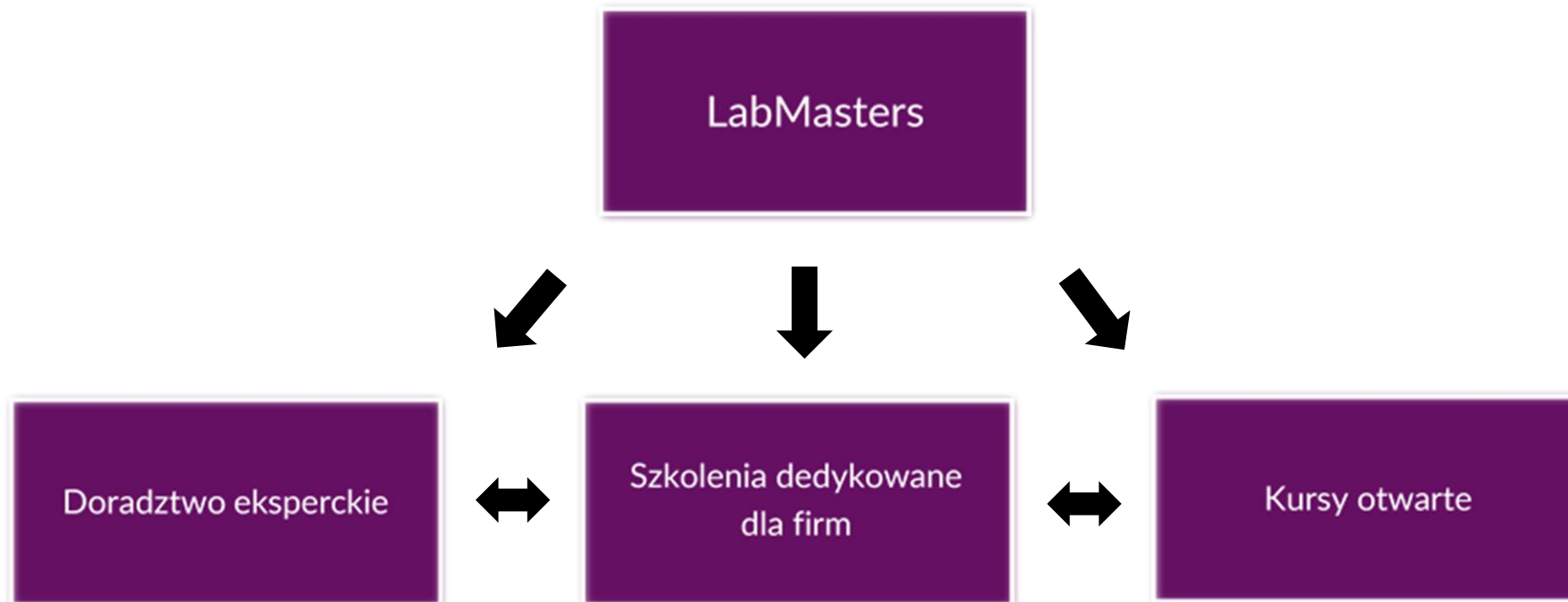
14 listopada 2023 r.



Program zajęć

- **Explainable Machine Learning (XAI).** Przegląd dostępnych pakietów (ze szczególnym uwzględnieniem Dalex). Omówienie i porównanie technik wyjaśniania modeli dla obserwacji (m. in. break-down plot, liczby Shapleya, wykres Ceteris-Paribus), dla zmiennych (m.in. rankingi ważności zmiennych, Partial Dependence Profile Plot, Local-dependence i ALE plot), diagnostyka modelu i analiza rozkładu reszt. Przykłady biznesowej ewaluacji produkcyjnych wyników modelu w formie case study.
- **Modelowanie danych niezbalansowanych.** Metody próbkowania danych (sampling): under-sampling, over-sampling, SMOTE, ROSE, Tomek links, NearMiss) wykorzystanie odpowiednich metryk przy walidacji i wyborze modelu, modyfikacja i strojenie hiperparametrów w problemach modelowania klas niezbalansowanych (m. in. XGBoost, Penalized-SVM).
- **Detekcja anomalii.** Teoria detekcji anomalii (definicje, typy obserwacji odstających, klasyfikacja podejść do wykrywania outlierów). Różnica między obserwacją nietypową a wpływową. Przegląd statystyk i modeli wykrywających anomalie, m. in.: test Tukeya, test Grubba, kryterium z-score, odległość Cooka, Isolation Forest, One-class SVM, DBSCAN, KNN, Histogram-based Outlier Detection (HBOS), Cluster-based Local Outlier Factor (CBLOF) i Autoencodery.
- **Text Mining/Natural Language Processing.** Funkcje tekstowe, wyrażenia regularne, czyszczenie i obróbka tekstu, lematyzacja, stemizacja, stop-words, analiza sentymentu, segmentacja danych, klasyfikacja dokumentów (uczenie nadzorowane). Algorytm word2vec (word embedding) i jego rozwinięcia. Latent Semantic Analysis, Latent Dirichlet Allocation, metryki odległości dla danych tekstowych.
- **Sieci neuronowe.** Typowe struktury sieci neuronowych i ich konstrukcja (MLP, CNN), przegląd i znaczenie funkcji aktywujących, porównanie algorytmów optymalizacyjnych (m. in. SGD, Adam), techniki ograniczania ryzyka przetrenowania modelu (warstwa dropout, L1, L2), dobre praktyki tuningu hiperparametrów. Praktyka tworzenia sieci neuronowych w pakietach tensorflow oraz keras. Przykłady i wprowadzenie do zaawansowanych sieci neuronowych (RNN, LSTM).

0 LabMasters



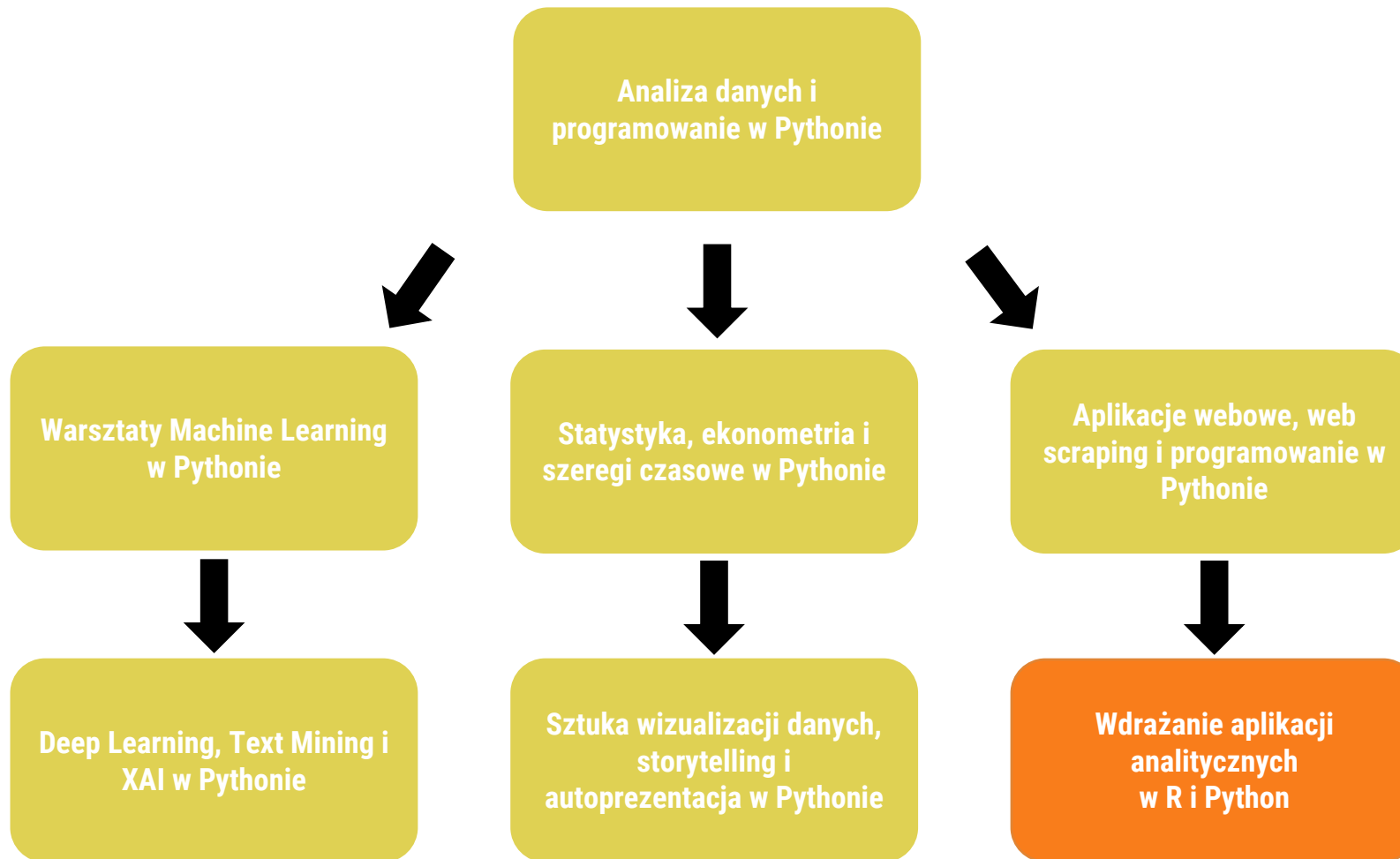
Informacje kontaktowe:

Labmasters.pl
Labmasters.pl/facebook
biuro@labmasters.pl

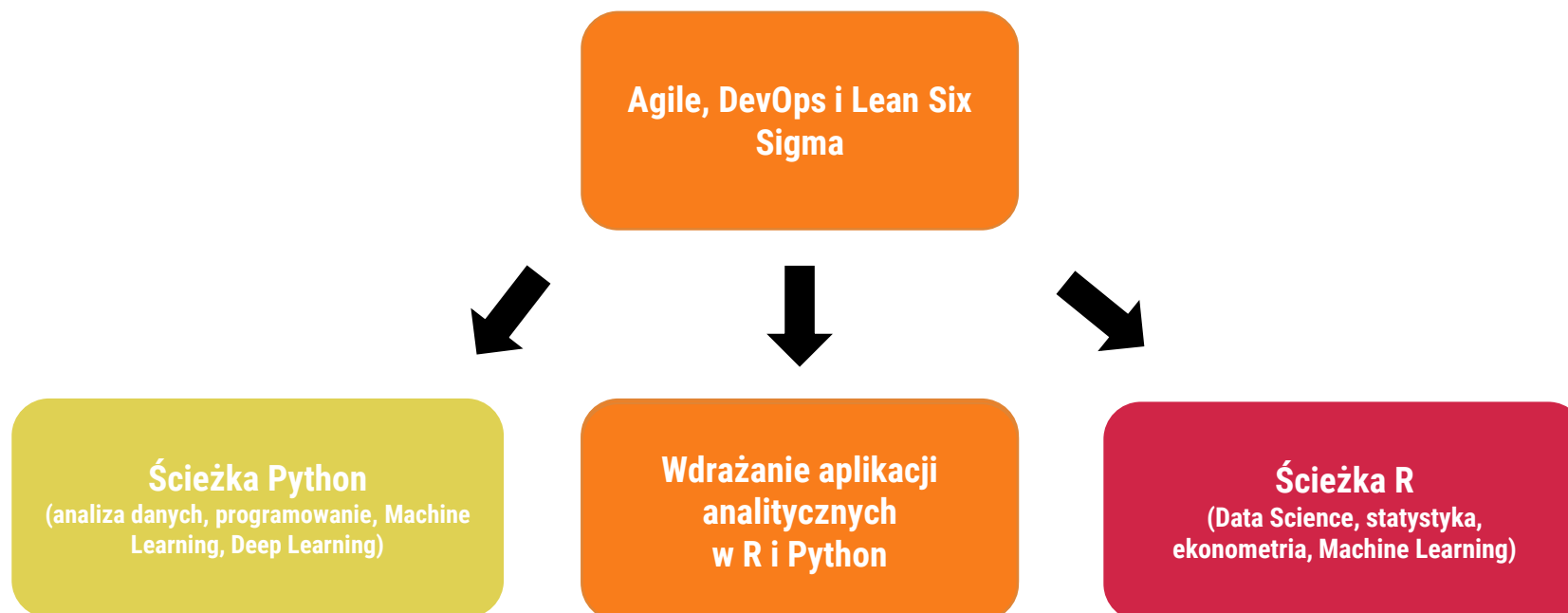
Kontakt do prowadzącego:

Piotr Ćwiakowski
Koordynator kursów R, Python, Data Science
pcwiakowski@labmasters.pl

Ścieżka Python



Ścieżka Agile



Organizacja zajęć

Tryb pracy na zajęciach

- wykład (część teoretyczna)
- ćwiczenia (praktyczna implementacja w języku Python)
- warsztat (praca samodzielna pod kierunkiem prowadzącego)
- praca samodzielna w domu (powtarzanie materiału z zajęć)

Materiały elektroniczne dostępne są:

- w pracowni komputerowej (na dysku X w folderze "pcwiakowski")
- na platformie edukacyjnej Moodle (moodle.labmasters.pl)

W skład materiałów elektronicznych wchodzi:

- notebooki pythonowe (rozszerzenie .ipynb)
- plik html z prezentacją wyników
- ćwiczenia do samodzielnego wykonania wraz z rozwiązaniami zadań

Kontakt mailowy

- w sprawach organizacyjnych oraz merytorycznych
- w czasie trwania kursu oraz po jego zakończeniu
- adres: pcwiakowski@labmasters.pl

Środowisko pracy

- Manager pakietów Anaconda/Miniconda
- Jupyter Notebook

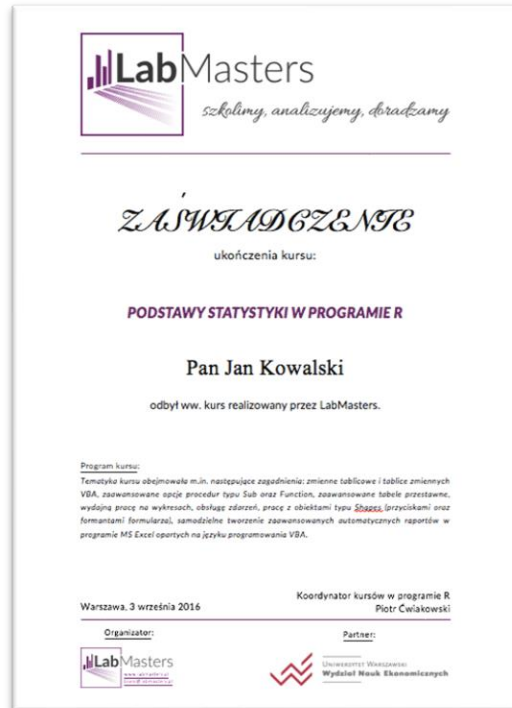


Kiedy przerwy?



16.30 – 18:30	I Blok
18:30 – 18:45	Przerwa
18:45 – 20:30	II Blok

Zaliczenie zajęć



Zaświadczenie

Każdy uczestnik kursu posiadający ponad 50% obecności na zajęciach (min. 4 zajęcia) ma prawo do odbioru zaświadczenia potwierdzającego uczestnictwo w kursie.

Zaświadczenie uczestnik otrzymuje na ostatnich zajęciach kursu, po spełnieniu wyżej wymienionych warunków.

Certyfikat LabMasters

Każdy uczestnik kursu po jego zakończeniu ma prawo do przystąpienia do egzaminu.

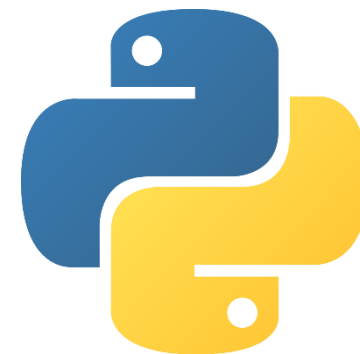
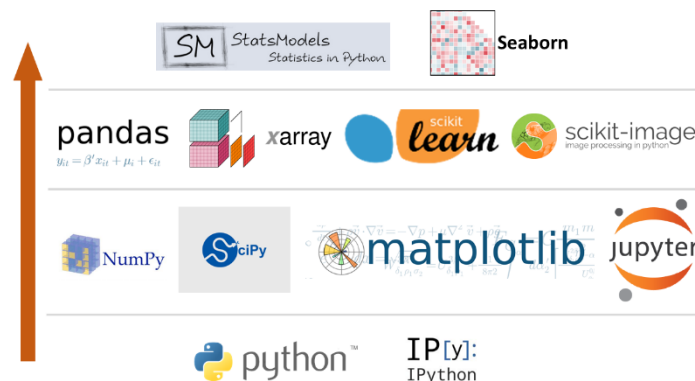
Egzamin składa się z dwóch części:

- samodzielne rozwiązanie zadanego problemu w Pythonie (w domu),
- ustne przedyskutowanie rozwiązania z prowadzącym.



0 Pythonie

- Licencja freeware
- Język ogólnego przeznaczenia
- Python v. IPython
- Wsparcie społeczności użytkowników
- Wiele tutoriali i materiałów w Internecie
- Uporządkowane uniwersum pakietów
- Intuicyjna składnia



0 Jupyter Notebook

- Licencja freeware
- Wygodny w prezentowaniu wyników
- Możliwość programowania zarówno w R jak i Python
- Możliwość eksportu m. in. do html, tex, pdf, docx (dzięki Markdown)
- Możliwość rozbudowy minimalistycznego interfejsu poprzez instalację wtyczek
- Dla bardziej wymagających Jupyter Lab



0 Anacondzie

- Manager pakietów do Pythona (zapewniona kompatybilność wersji pakietów)
- Domyślnie wykorzystuje repozytorium pakietów dostarczanych przez Continuum Analytics (<https://www.continuum.io/>)
- Możliwość zarządzania wirtualnymi środowiskami
- Instalacja wszystkich potrzebnych komponentów jednym kreatorem
- Dostępne są różne wersje: Anaconda, Miniconda, Miniforge (conda-forge)

