

Demo: monitoring modelu

Cel

Celem naszego programu jest ewaluacja już wytrenowanego modelu na kolejnych partiach danych i zapis metryk jakości do osobnego pliku.

W module wykorzystamy:

- wytrenowany w poprzednim ćwiczeniu model regresji
- syntetyczne zbiory danych, zbliżone do zbioru treningowego, niemniej na tyle różne, by zidentyfikować potencjalny dryf modelu.

Stworzony dzięki temu program będziemy mogli później wykorzystać jako komponent procesu identyfikacji dryfu modelu.

Lista kontrolna

Skrypt, który stworzymy, będzie realizował następujące zadania:

- ☐ Wczytanie wytrenowanego modelu `model/model_1.0.pkl`
- ☐ Wczytanie danych testowych:
 - ☐ wybór paczki danych testowych: `batch_no` w zakresie od `1` do `6`
 - ☐ wczytanie odpowiedniej paczki
- ☐ Wygenerowanie predykcji modelu
- ☐ Obliczenie miar jakości:
 - ☐ RMSE
 - ☐ r^2
- ☐ Zapisanie w pliku `evaluation/model_eval.csv`:
 - ☐ stempla czasowego
 - ☐ nr paczki danych
 - ☐ wartości miary RMSE
 - ☐ wartości miary r^2
 - ☐ UWAGA:
 - ☐ jeśli pliku nie istnieje: utworzenie go
 - ☐ w przeciwnym przypadku: uzupełnienie pliku (dodanie aktualnych rekordów).

Architektura

Artefakty

1. Wejście

1. Model: `model/model_1.0.pkl`
2. Pliki testowe: `data/batch_n.csv`, z `n` w zakresie od `1` do `6`

2. Wyjście

1. Plik ewaluacyjny: `evaluation/model_eval.csv`

Komponenty

Jeden program `1. Evaluate.py` realizujący zadania z listy kontrolnej.

Decyzje

Projektując to rozwiązanie, musimy podjąć następujące decyzje:

1. Wybór miary jakości modelu

1. RMSE
2. R2