

Dokumentacja Automatycznej Analizy i Doboru Modeli

1. Opis Wybranego Narzędzia AutoML

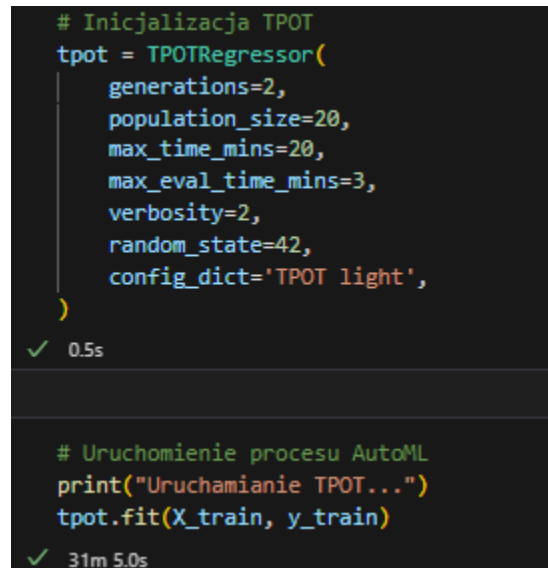
W projekcie wykorzystano narzędzie **TPOT (Tree-based Pipeline Optimization Tool)**, które automatycznie optymalizuje modele machine learningowe. TPOT używa algorytmów genetycznych do przeszukiwania przestrzeni możliwych pipeline'ów, składających się z preprocessingów, selekcji cech i modeli predykcyjnych. Dzięki temu wybiera najlepszy pipeline dopasowany do danych i celu analizy.

Kluczowe cechy TPOT:

- Automatyczne tworzenie, testowanie i wybór najlepszych pipeline'ów.
- Użycie cross-validation, co zapewnia stabilność wyników.
- Eksport kodu Python dla najlepszego modelu, co pozwala na łatwe wdrożenie.
- Wspiera zarówno klasyfikację, jak i regresję.

W projekcie zastosowano **TPOTRegressor** do problemu regresji, którego celem było przewidzenie oceny filmów (imdb_score) na podstawie dostępnych danych.

Uruchamianie TPOT trwało niezwykle długo i musiałem zmienić parametry przez co uzyskałem dużo gorsze wyniki niż przy większej ilości generacji.



```
# Inicjalizacja TPOT
tpot = TPOTRegressor(
    generations=2,
    population_size=20,
    max_time_mins=20,
    max_eval_time_mins=3,
    verbosity=2,
    random_state=42,
    config_dict='TPOT light',
)

# Uruchomienie procesu AutoML
print("Uruchamianie TPOT...")
tpot.fit(X_train, y_train)
```

The screenshot shows two code cells in a Jupyter Notebook. The first cell initializes a TPOTRegressor with parameters: generations=2, population_size=20, max_time_mins=20, max_eval_time_mins=3, verbosity=2, random_state=42, and config_dict='TPOT light'. It shows a green checkmark and a runtime of 0.5s. The second cell prints "Uruchamianie TPOT..." and calls tpot.fit(X_train, y_train). It also shows a green checkmark and a runtime of 31m 5.0s.

generations=2:

- Liczba generacji, które TPOT przeprowadzi w ramach procesu optymalizacji. Każda generacja to jeden cykl ewolucji algorytmów genetycznych. Im wyższa liczba generacji, tym bardziej zoptymalizowane mogą być modele.

population_size=20:

- Wielkość populacji modeli w każdej generacji. Większa populacja pozwala na większą różnorodność wśród testowanych modeli, ale zwiększa czas obliczeń.

`max_time_mins=20:`

- Maksymalny czas trwania całego procesu optymalizacji (w minutach). Po upływie tego czasu proces zostanie zakończony, nawet jeśli nie osiągnięto maksymalnej liczby generacji.

`max_eval_time_mins=3:`

- Maksymalny czas na ocenę pojedynczego modelu (w minutach). Ustawienie tego parametru zapobiega zbyt długiemu testowaniu pojedynczych modeli.

`verbosity=2:`

- Poziom szczegółowości komunikatów wyświetlanych podczas działania TPOT. Wartość 2 oznacza umiarkowaną ilość informacji, w tym postęp procesu i wyniki najlepszych modeli.

`random_state=42:`

- Ustawienie stałego ziarna losowości, co zapewnia powtarzalność wyników przy ponownym uruchomieniu kodu.

`config_dict='TPOT light':`

- Używana konfiguracja modeli. TPOT light ogranicza zbiór testowanych algorytmów do prostszych i szybszych w ocenie modeli, co jest przydatne przy ograniczonym czasie optymalizacji.

2. Wnioski z Raportu Automatycznego

Przed uruchomieniem TPOT przeprowadzono eksploracyjną analizę danych (EDA) z wykorzystaniem bibliotek Pandas, Matplotlib i Seaborn, a także automatycznego narzędzia **ydata-profiling**.

Overview

Brought to you by YData

Overview

Alerts22

Reproduction

Dataset statistics

Number of variables	27
Number of observations	3755
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	33
Duplicate rows (%)	0.9%
Total size in memory	3.5 MiB
Average record size in memory	969.1 B

Variable types

Categorical	4
Text	7
Numeric	16

Variables

Select Columns

color

Categorical

Imbalance

Distinct	2
Distinct (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	258.0 KiB



More details

Statystyki Zbioru Danych:

- Liczba zmiennych: 27 (4 kategoriyczne, 7 tekstowych, 16 numerycznych).
- Liczba rekordów: 3755.
- Brakujące dane: Brak brakujących wartości w całym zbiorze (0% missing).
- Duplikaty: Występują 33 powielone wiersze (0.9%), które warto usunąć.

Struktura Pamięci:

- Całkowity rozmiar w pamięci: 3.5 MB.
- Średni rozmiar rekordu: 969.1 B.

Alerts

Dataset has 33 (0.9%) duplicate rows	Duplicates
actor_1_facebook_likes is highly overall correlated with actor_2_facebook_likes and 2 other fields	High correlation
actor_2_facebook_likes is highly overall correlated with actor_1_facebook_likes and 2 other fields	High correlation
actor_3_facebook_likes is highly overall correlated with actor_1_facebook_likes and 2 other fields	High correlation
budget is highly overall correlated with gross and 1 other fields	High correlation
cast_total_facebook_likes is highly overall correlated with actor_1_facebook_likes and 2 other fields	High correlation
country is highly overall correlated with language	High correlation
gross is highly overall correlated with budget and 2 other fields	High correlation
language is highly overall correlated with budget and 1 other fields	High correlation
num_critic_for_reviews is highly overall correlated with num_user_for_reviews and 2 other fields	High correlation
num_user_for_reviews is highly overall correlated with gross and 2 other fields	High correlation
num_voted_users is highly overall correlated with gross and 2 other fields	High correlation
title_year is highly overall correlated with num_critic_for_reviews	High correlation
color is highly imbalanced (79.1%)	Imbalance
language is highly imbalanced (91.7%)	Imbalance
country is highly imbalanced (74.5%)	Imbalance
content_rating is highly imbalanced (50.4%)	Imbalance
actor_1_facebook_likes is highly skewed ($\gamma_1 = 20.33840332$)	Skewed
budget is highly skewed ($\gamma_1 = 44.16873671$)	Skewed
director_facebook_likes has 642 (17.1%) zeros	Zeros
facenumber_in_poster has 1581 (42.1%) zeros	Zeros
movie_facebook_likes has 1742 (46.4%) zeros	Zeros

Kluczowe wnioski z analizy:

- Zidentyfikowano 33 powielone wiersze (0.9% danych), które należy usunąć, aby uniknąć błędów w analizie.
- budget i gross: Silna zależność między budżetem a przychodami.
- num_voted_users, gross, num_user_for_reviews: Popularność filmu (liczba głosów i recenzji) jest mocno powiązana z dochodami.
- actor_1_facebook_likes, actor_2_facebook_likes, cast_total_facebook_likes: Polubienia aktorów i obsady są skorelowane, co wskazuje na redundancję.
- language (91.7%): Dane są silnie zdominowane przez jeden język
- color (79.1%): Większość filmów jest w kolorze, co ogranicza różnorodność tej cechy.
- content_rating (50.4%): Dominacja jednej kategorii wiekowej.

TPOT przetestował różne pipeline'y, a poniżej przedstawiono trzy najwyżej ocenione modele:

1. Pipeline 1: Random Forest Regressor z minimalnym preprocessingiem

- Algorytm: RandomForestRegressor
- Preprocessing: Skalowanie za pomocą StandardScaler.
- Uzasadnienie: Random Forest jest odporny na dane o dużej liczbie cech i nie wymaga rygorystycznej normalizacji danych. Sprawdził się dobrze w problemach regresji.

2. Pipeline 3: Gradient Boosting Regressor

- Algorytm: GradientBoostingRegressor
- Preprocessing: Skalowanie cech i usuwanie cech wysoko skorelowanych.
- Uzasadnienie: Gradient Boosting jest jednym z najwydajniejszych modeli do regresji i radzi sobie z danymi nieliniowymi. Daje lepsze wyniki niż proste modele liniowe.

3. Pipeline 2: Ridge Regression

- Algorytm: RidgeCV (z automatyczną walidacją współczynnika regularizacji)
- Preprocessing: Skalowanie cech za pomocą StandardScaler.
- Uzasadnienie: Model liniowy dobrze radzi sobie z danymi o umiarkowanej liczbie cech. Regularizacja eliminuje nadmierne dopasowanie.