Jaka motywacja stoi za AutoML?

Podsumowanie PD1 i PD2

- Jaki preprocessing wykonaliście i czym różniły się Wasze rozwiązania?
  - Bazujcie na tabelce z podsumowaniem PD1

- Kto zbudował najdokładniejszy model z domyślnymi hiperparametrami? Czy optymalizacja hiperparametrów zmieniła wyniki?
  - Zastanówcie się co może być przyczyną różnic w wynikach.

Preprocessing

- Label encoding
- usuwanie kolumn (braki danych, małe zróżnicowanie wartości)
- kategoryzowanie zmiennych
- dodawanie nowej kategorii zmaiast NA
-

Najlepsze wyniki - domyślne hiperparametry

- Random Forest
- LightGBM
- Gradient Boosting
-

Metody optymalizacji

- Która metoda optymalizacji zajęła najwięcej czasu?
- Która metoda optymalizacji najlepiej zadziałała?

Czas wykonywania optymalizacji

Random search

Grid search - najdłużej

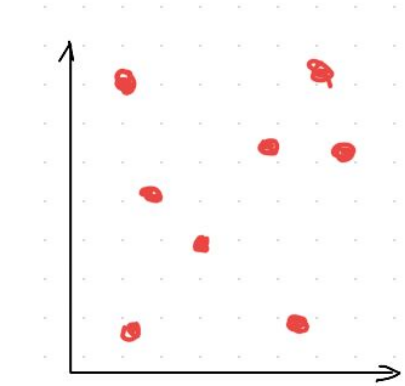
Bayesian optimization

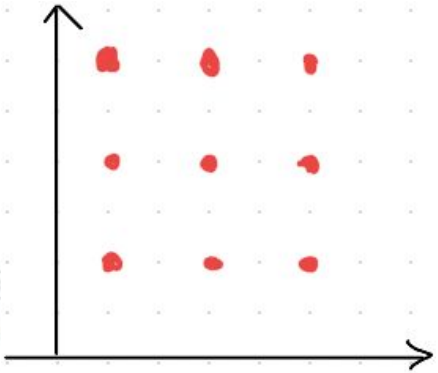
Jak właściwie porównywać wyniki?

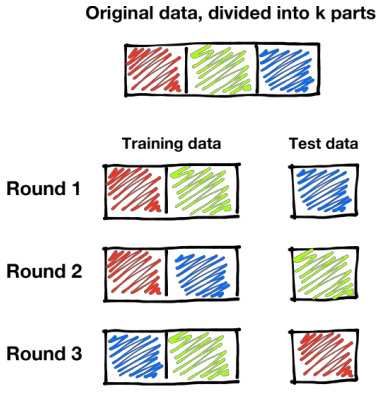
Czy różnice są istotne?

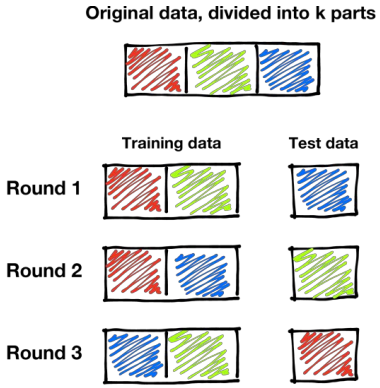
AUC (default)	AUC (GS)	AUC (RS)	AUC (BO)
0.8441008043128087	0.8508267230021713	0.8511484391511555	0.850817036655483

x CV

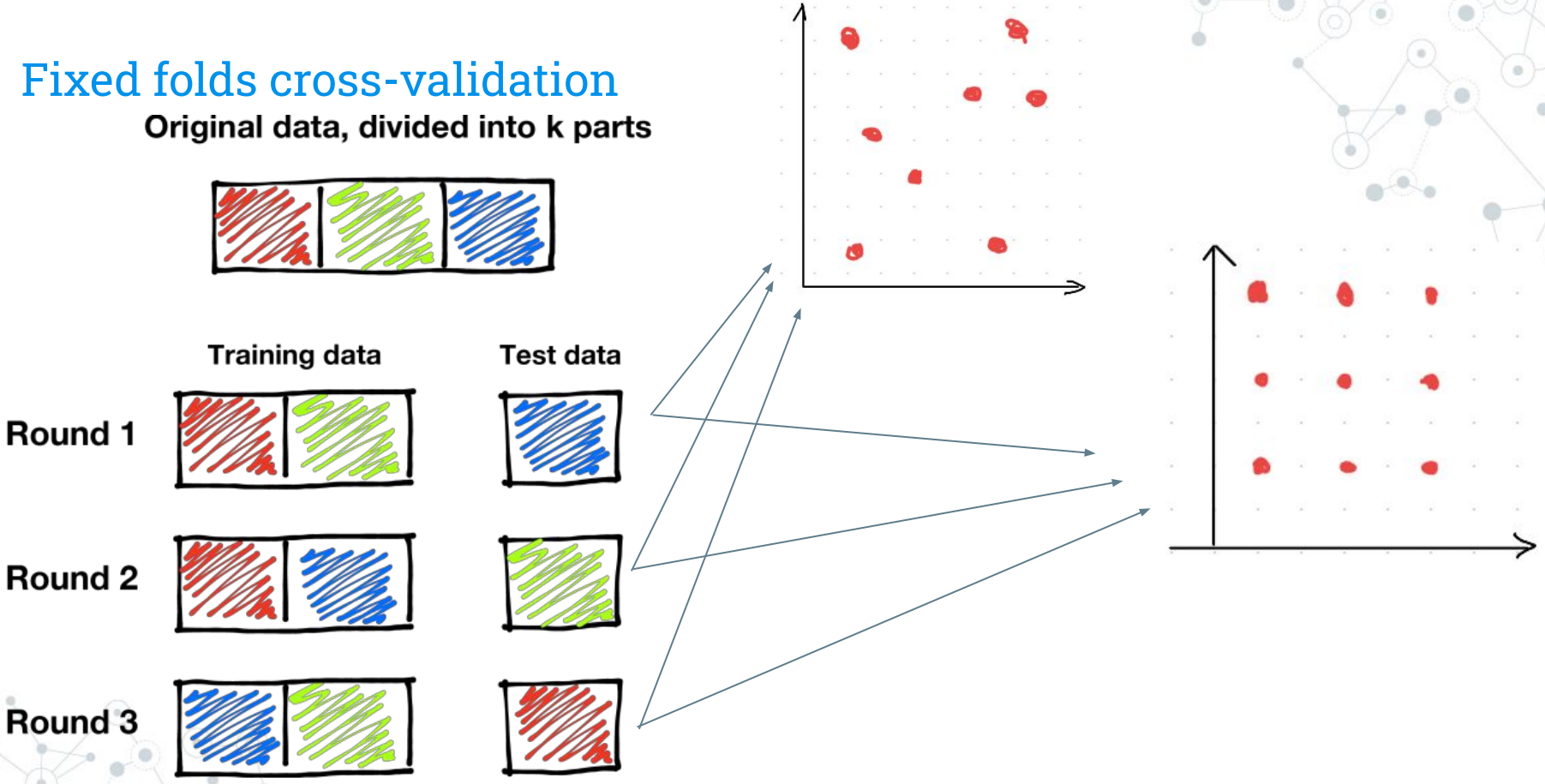





Original data, divided into k parts
Training data
Test data
Round 1
Round 2
Round 3

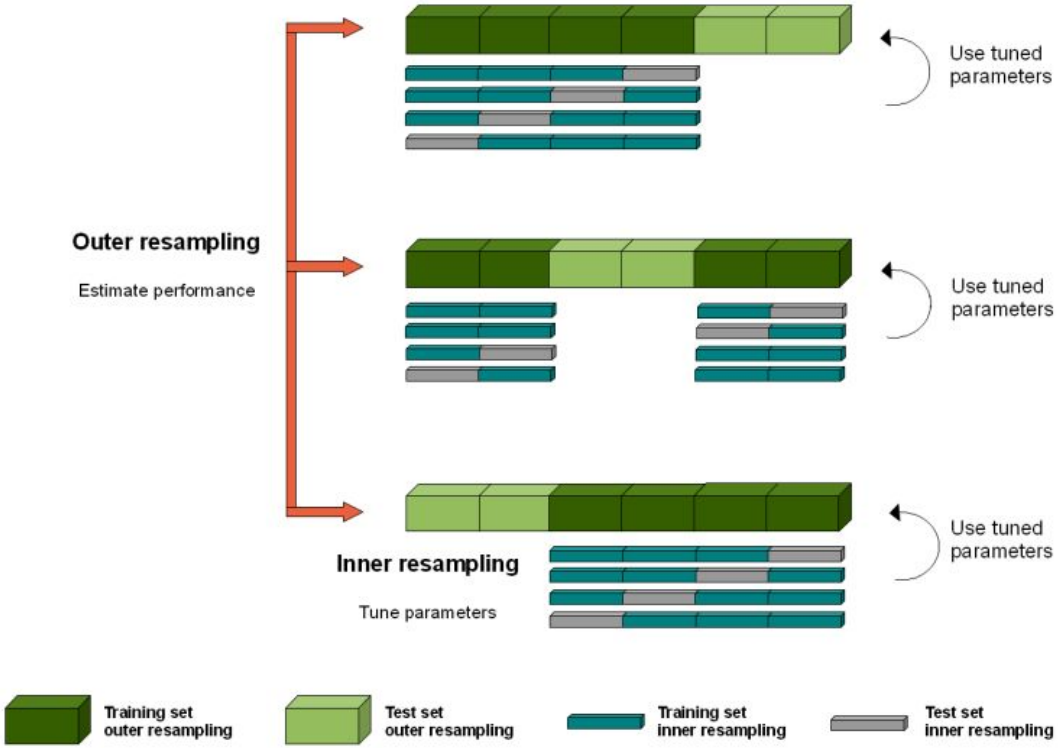

Original data, divided into k parts
Training data
Test data
Round 1
Round 2
Round 3

Fixed folds cross-validation

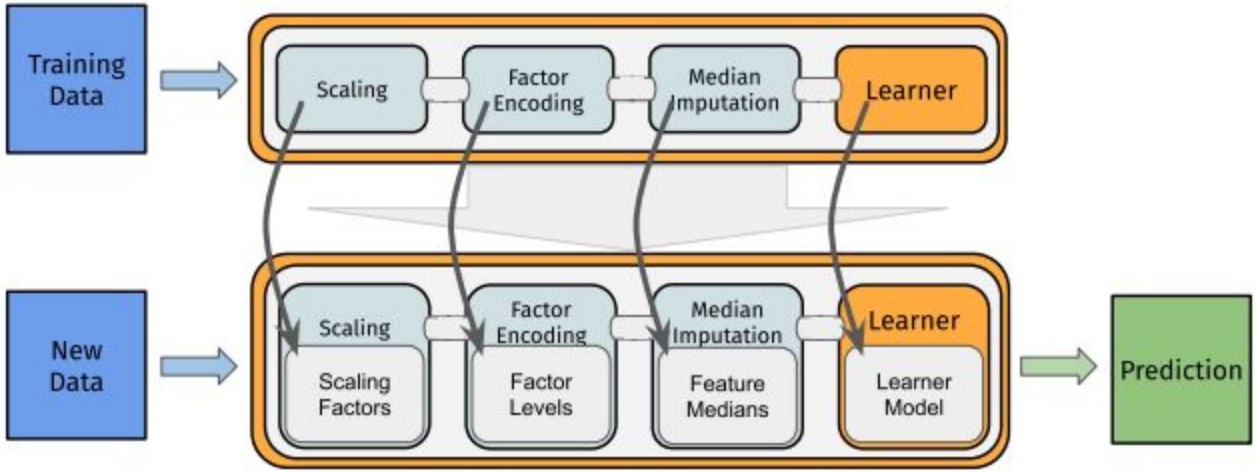


Czy używając kroswalidacji estymujemy właściwy błąd?

Nested resampling



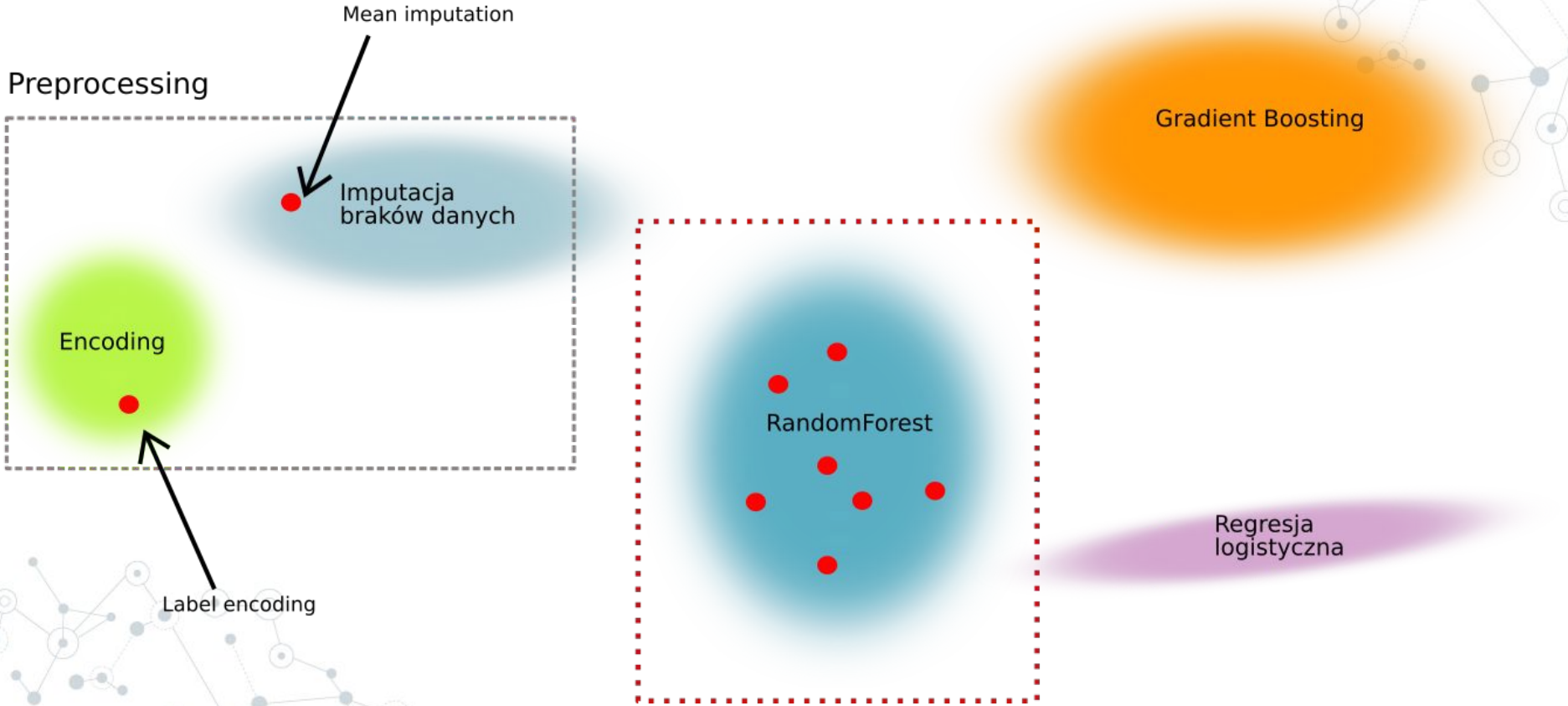
Preprocessing - dane treningowe (pipeline)



Oczekiwania wobec AutoML

- wyręczy nas w żmudnej pracy testowania wielu metod (ulga dla DS)
- przeprowadzamy eksperymenty w sposób systematyczny z uwzględnieniem bardziej zaawansowanych technik (nawet jeśli przeprowadza je osoba bez doświadczenia w ML)
- w systematyczny sposób można zbierać informacje o tym, które metody dają najlepsze wyniki - wsparcie kolejnych metod AutoML

Hyperparameter optimization



Definition

Let

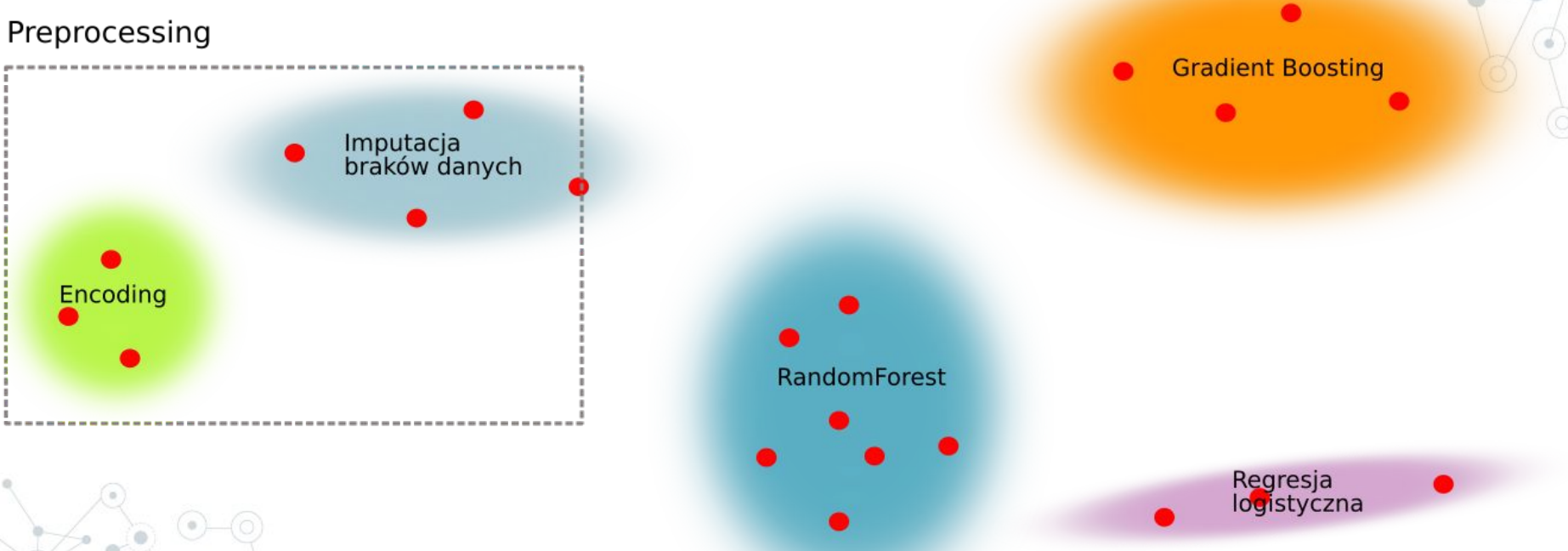
- $\boldsymbol{\lambda}$ be the hyperparameters of an ML algorithm $\mathcal{A}$ with domain $\boldsymbol{\Lambda}$,
- $\mathcal{D}_{opt}$ be a dataset which is split into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$
- $c(\mathcal{A}_{\boldsymbol{\lambda}}, \mathcal{D}_{train}, \mathcal{D}_{valid})$ denote the cost of $\mathcal{A}_{\boldsymbol{\lambda}}$ trained on $\mathcal{D}_{\text{train}}$ and evaluated on $\mathcal{D}_{\text{val}}$.

The *hyper-parameter optimization (HPO)* problem is to find a hyper-parameter configuration that minimizes this cost:

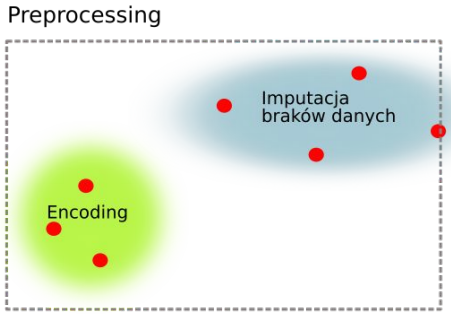
$$\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} c(\mathcal{A}_{\boldsymbol{\lambda}}, \mathcal{D}_{train}, \mathcal{D}_{valid})$$

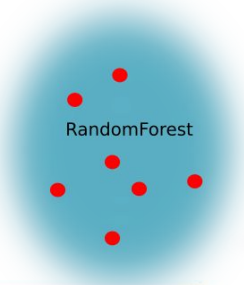
https://learn.ki-campus.org/courses/automl-luh2021/items/2Hir3f8YnNbuLZTu0nxYMw

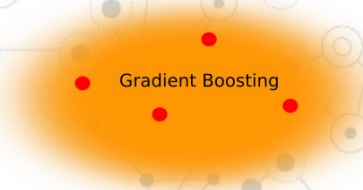
CASH: Combined Algorithm Selection and Hyperparameter Optimization



CASH







Definition

Let

- $\mathbf{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k\}$ be a set of algorithms (a.k.a. portfolio)
- $\mathbf{\Lambda}$ be a set of hyperparameters of each machine learning algorithm $\mathcal{A}_i$
- $\mathcal{D}_{opt}$ be a dataset which is split into $\mathcal{D}_{train}$ and $\mathcal{D}_{valid}$
- $c(\mathcal{A}_{\boldsymbol{\lambda}}, \mathcal{D}_{train}, \mathcal{D}_{valid})$ denote the cost of $\mathcal{A}_{\boldsymbol{\lambda}}$ trained on $\mathcal{D}_{train}$ and evaluated on $\mathcal{D}_{valid}$.

we want to find the best combination of algorithm $\mathcal{A} \in \mathbf{A}$ and its hyperparameter configuration $\boldsymbol{\lambda} \in \mathbf{\Lambda}$ minimizing:

$$(\mathcal{A}^*, \boldsymbol{\lambda}^*) \in \underset{\mathcal{A} \in \mathbf{A}, \boldsymbol{\lambda} \in \mathbf{\Lambda}}{\arg\min}\, c(\mathcal{A}_{\boldsymbol{\lambda}}, \mathcal{D}_{train}, \mathcal{D}_{valid})$$

https://learn.ki-campus.org/courses/automl-luh2021/items/2Hir3f8YnNbuLZTu0nxYMw

Jakie frameworki istnieją?

- Autosklearn
- AutoWEKA artykuł
- Autogulon artykuł -
- AutoKeras artykul
- AutoPytorch
- GAMA github
- FLAML
- Hyperopt
- naiveAutoML
- ML-Plan
- TPOT artykuł
- AutoPrognosis
- H2O AutoML artykuł
- Oboe
- LightAutoML
- AutoXGBoost artykuł

Kamień milowy 1

Każdy zespół projektowy ma zadanie wybrać jeden z istniejących pakietów do AutoML (każdy zespół inny) i zapoznać się z literaturą dotyczącą tego rozwiązania. W szczególności należy przeanalizować czy pakiet oferuje różne metody preprocessingu, czy wykonywana jest optymalizacja hiperparametrów, jaka klasa algorytmów ML jest dostępna.

Na podstawie badania literaturowego zespół przygotowuje prezentację dotyczącą pakietu (7.04). Czas trwania prezentacji 25 min

Mile widzane pierwsze doświadczenia z kodem.