Siatki hiperparametrów modeli

1 Logistic Regression

RandomizedSearch

Parametr	Zakres / Rozkład
`solver`	{lbfgs}
`penalty`	{l2}
`C`	`loguniform(1e-3, 1e3)`
`max_iter`	`randint(1000, 4000)`
`solver`	{saga}
`penalty`	{l1, l2, elasticnet}
`l1_ratio`	`uniform(0.05, 0.9)`
`C`	`loguniform(1e-3, 1e3)`
`max_iter`	`randint(1000, 4000)`

Tabela 1: Przestrzenie hiperparametrów Logistic Regression — RandomizedSearch.

BayesSearch

Pod-przestrzeń (liczba prób)	Definicja

`lbfgs + l2` (25)	`solver = Categorical([lbfgs])`
	`penalty = Categorical([l2])`
	`C = Real(1e-3, 1e3, prior='log-uniform')`
	`max_iter = Integer(1000, 4000)`

`saga + {l1,l2}` (25)	`solver = Categorical([saga])`
	`penalty = Categorical([l1, l2])`
	`C = Real(1e-3, 1e3, prior='log-uniform')`
	`max_iter = Integer(1000, 4000)`

`saga + elasticnet` (25)	`solver = Categorical([saga])`
	`penalty = Categorical([elasticnet])`
	`l1_ratio = Real(0.05, 0.9)`
	`C = Real(1e-3, 1e3, prior='log-uniform')`
	`max_iter = Integer(1000, 4000)`

Tabela 2: Przestrzenie hiperparametrów Logistic Regression — BayesSearch.

Uzasadnienie: Parametr C odpowiada za siłę regularyzacji (niższe wartości $\Rightarrow$ silniejsza regularyzacja). Solver `saga` umożliwia wykorzystanie rzadkiej regularizacji L1 oraz ElasticNet.

Przedziały iteracji zapewniają wystarczającą liczbę kroków optymalizacji nawet przy danych trudnych do konwergencji.

2 Random Forest

RandomizedSearch

Parametr	Zakres / Rozkład
`n_estimators`	`randint(10, 200)`
`max_depth`	`randint(2, 20)`
`min_samples_split`	`randint(2, 8)`
`min_samples_leaf`	`randint(1, 5)`
`max_features`	{sqrt, log2, None}
`bootstrap`	{True, False}
`criterion`	{gini, entropy, log_loss}
`class_weight`	{None, balanced}

Tabela 3: Przestrzenie hiperparametrów Random Forest — RandomizedSearch.

BayesSearch

Parametr	Przestrzeń
`n_estimators`	`Integer(10, 200)`
`max_depth`	`Integer(2, 20)`
`min_samples_split`	`Integer(2, 8)`
`min_samples_leaf`	`Integer(1, 5)`
`max_features`	`Categorical([sqrt, log2, None])`
`bootstrap`	`Categorical([True, False])`
`criterion`	`Categorical([gini, entropy, log_loss])`
`class_weight`	`Categorical([None, balanced])`

Tabela 4: Przestrzenie hiperparametrów Random Forest — BayesSearch.

Uzasadnienie: Zakres liczby drzew (10–200) zapewnia kompromis między dokładnością a czasem treningu. Zmienność głębokości i liczby próbek na liść pozwala regulować poziom przeuczenia. Użycie różnych kryteriów i wag klas umożliwia dopasowanie modelu do danych niezrównoważonych.

3 XGBoost

RandomizedSearch

Parametr	Zakres / Rozkład
n_estimators	randint(300, 1200)
max_depth	randint(3, 8)
learning_rate	loguniform(0.01, 0.1)
subsample	uniform(0.6, 0.4)
colsample_bytree	uniform(0.6, 0.4)
min_child_weight	randint(1, 8)
gamma	uniform(0.0, 3.0)
reg_lambda	uniform(0.0, 10.0)
reg_alpha	uniform(0.0, 5.0)
scale_pos_weight	uniform(1.0, 5.0)

Tabela 5: Przestrzenie hiperparametrów XGBoost — RandomizedSearch.

BayesSearch

Parametr	Przestrzeń
n_estimators	Integer(300, 1200)
max_depth	Integer(3, 8)
learning_rate	Real(0.01, 0.1, prior='log-uniform')
subsample	Real(0.6, 1.0)
colsample_bytree	Real(0.6, 1.0)
min_child_weight	Integer(1, 8)
gamma	Real(0.0, 3.0)
reg_lambda	Real(0.0, 10.0)
reg_alpha	Real(0.0, 5.0)
scale_pos_weight	Real(1.0, 5.0)

Tabela 6: Przestrzenie hiperparametrów XGBoost — BayesSearch.

Uzasadnienie: Zakresy odzwierciedlają standardowe praktyki dla XGBoost: `learning_rate` od 0.01 do 0.1 zapewnia kompromis między szybkością nauki a stabilnością. Parametry regularizacji (`reg_alpha`, `reg_lambda`) ograniczają przeuczenie, a `scale_pos_weight` kompensuje niezrównoważone klasy.