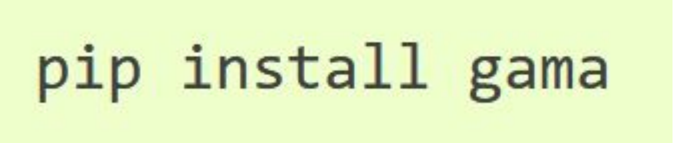
GAMA

GAMA

- narzędzie do automatycznego uczenia maszynowego
- potrafi automatycznie wykonać preprocessing, dobrać model oraz wykonać optymalizację hiperparametrów
- ma swoje repozytorium na Githubie
- prosty w obsłudze
- mało popularny: 62 gwiazdki na githubie, 0 cytowań

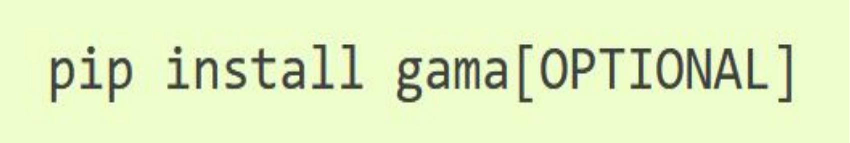
Instalacja

Zwykła wersja



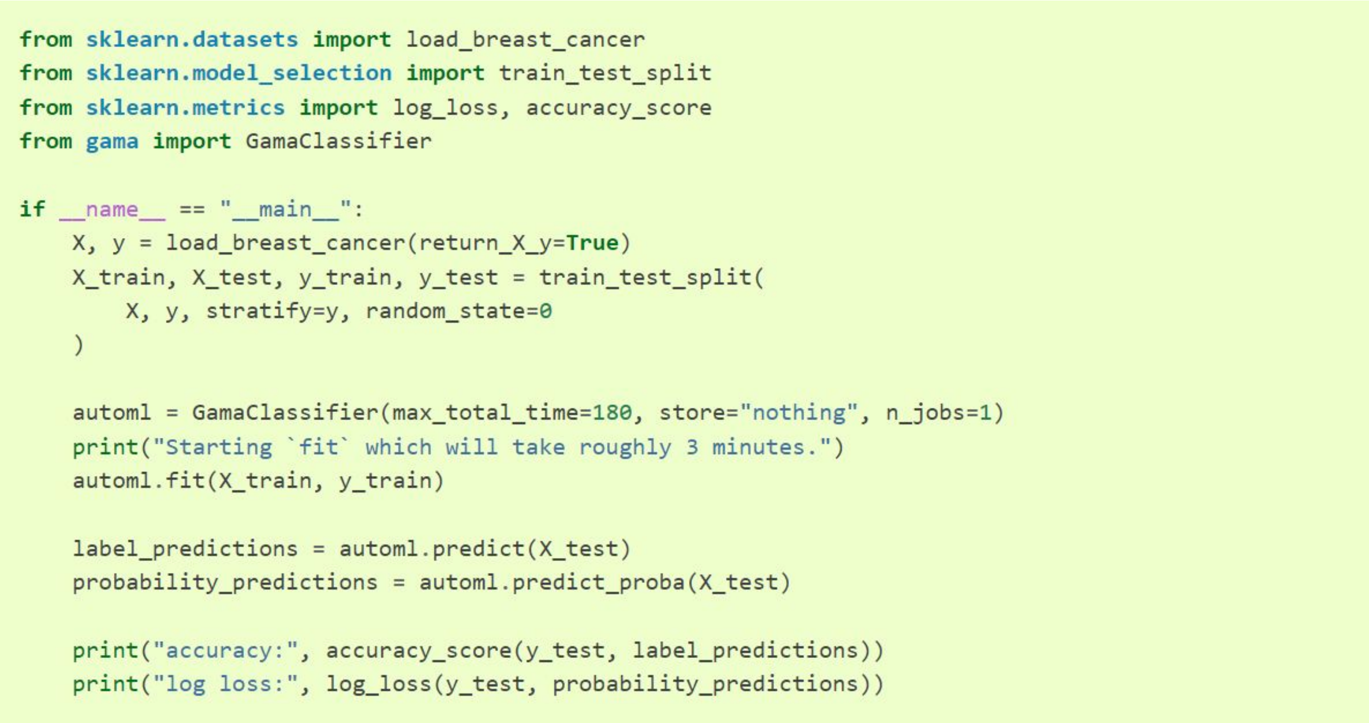
```
pip install gama
```

Wersja z dodatkami



```
pip install gama[OPTIONAL]
```

Sposób działania



```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.metrics import log_loss, accuracy_score
from gama import GamaClassifier

if __name__ == "__main__":
    X, y = load_breast_cancer(return_X_y=True)
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, stratify=y, random_state=0
    )

    automl = GamaClassifier(max_total_time=180, store="nothing", n_jobs=1)
    print("Starting `fit` which will take roughly 3 minutes.")
    automl.fit(X_train, y_train)

    label_predictions = automl.predict(X_test)
    probability_predictions = automl.predict_proba(X_test)

    print("accuracy:", accuracy_score(y_test, label_predictions))
    print("log loss:", log_loss(y_test, probability_predictions))
```

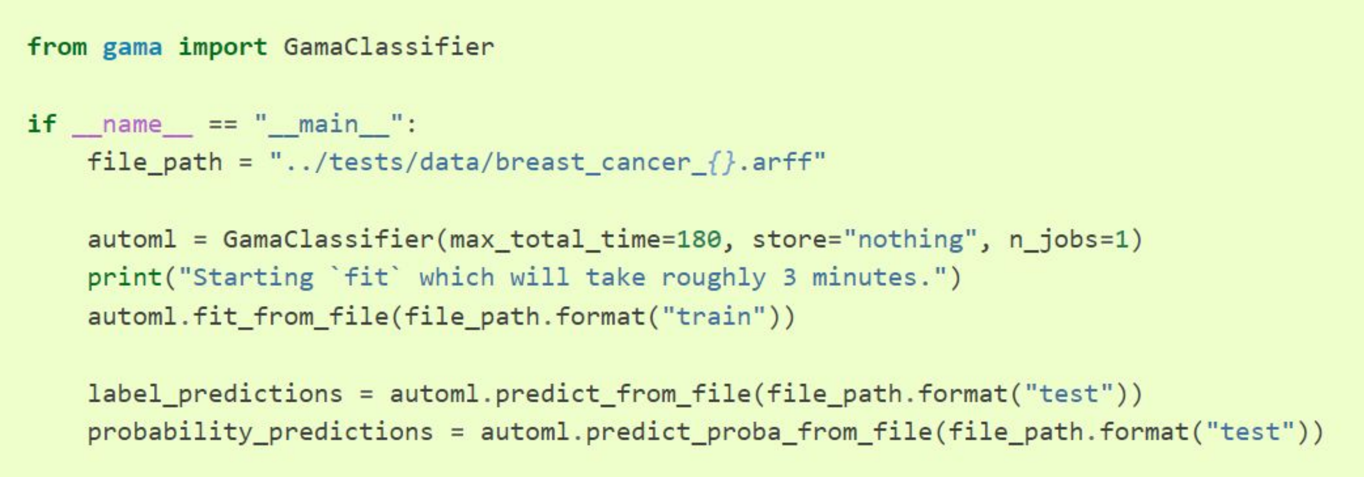
Parametry

- scoring = (str, Metric, Tuple)
- regularize_length = bool
- max_pipeline_length = (int, optional)
- config = dict
- random_state = int
- verbosity = int
- search = BaseSearch
- post_processing = BasePostProcessing
- output_directory = (str, optional)
- store = str

Parametry zasobów

- n_jobs = (int, optional)
- max_total_time = int
- max_eval _time = int
- max_memory_mb = int

Wgranie danych z pliku



```
from gama import GamaClassifier

if __name__ == "__main__":
    file_path = "../tests/data/breast_cancer_{}.arff"

    automl = GamaClassifier(max_total_time=180, store="nothing", n_jobs=1)
    print("Starting `fit` which will take roughly 3 minutes.")
    automl.fit_from_file(file_path.format("train"))

    label_predictions = automl.predict_from_file(file_path.format("test"))
    probability_predictions = automl.predict_proba_from_file(file_path.format("test"))
```

Preprocessing

Zmienne kategoryczne

- OneHotEncoder (≤ 10)
- OrdinalEncoder (> 10)
- TargetEncoder (> 10)
- Imputowane medianą

Zmienne numeryczne

- MinMaxScaler
- MaxAbsScaler
- StandardScaler
- Normalizer
- PolynomialFeatures
- Nystroem
- RBFSampler
- PCA

Zastosowane modele

Klasyfikacyjne

- modele bayesowskie
- regresja logistyczna
- SVM
- K najbliższych sąsiadów
- komitety modeli

Regresyjne

- modele liniowe
- SVM
- K najbliższych sąsiadów
- komitety modeli

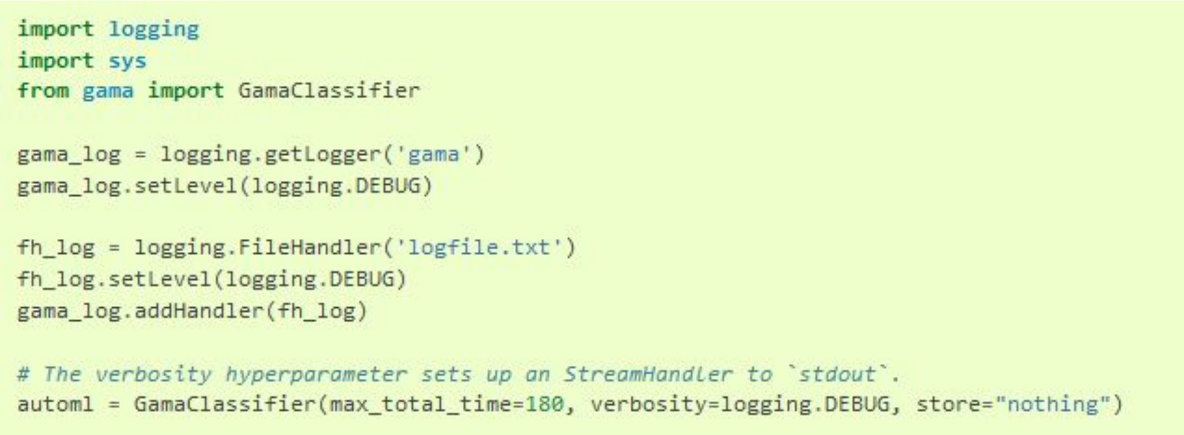
Algorytmy poszukiwań

- Random Search
- Asynchronous Evolutionary Algorithm (default)
- Asynchronous Successive Halving Algorithm

Post - processing

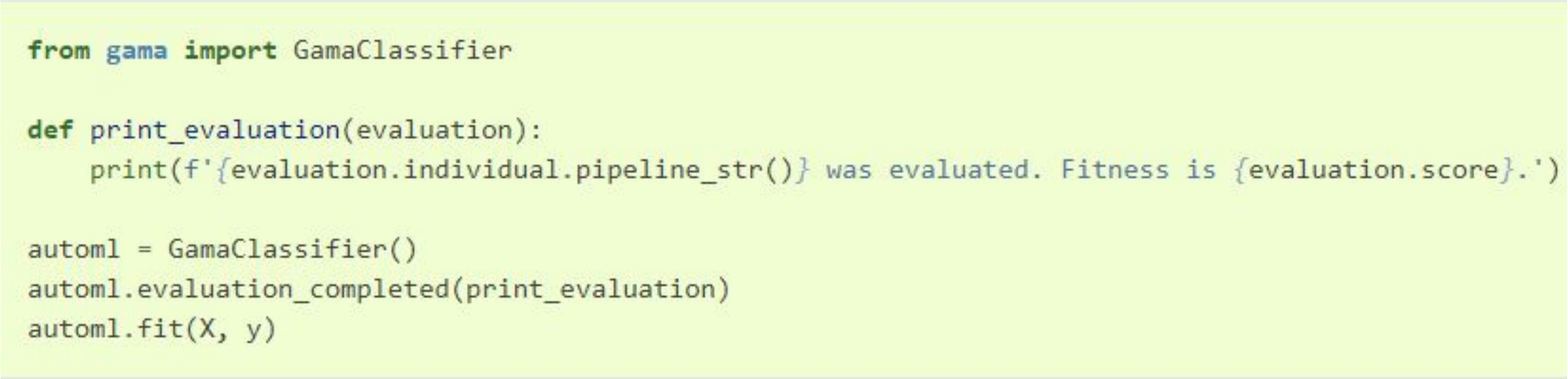
- NoPostProcessing
- BestFitPostProcessing (default)
- EnsemblePostProcessing

Logging



```
import logging
import sys
from gama import GamaClassifier

gama_log = logging.getLogger('gama')
gama_log.setLevel(logging.DEBUG)

fh_log = logging.FileHandler('logfile.txt')
fh_log.setLevel(logging.DEBUG)
gama_log.addHandler(fh_log)

# The verbosity hyperparameter sets up an StreamHandler to `stdout`.
automl = GamaClassifier(max_total_time=180, verbosity=logging.DEBUG, store="nothing")
```

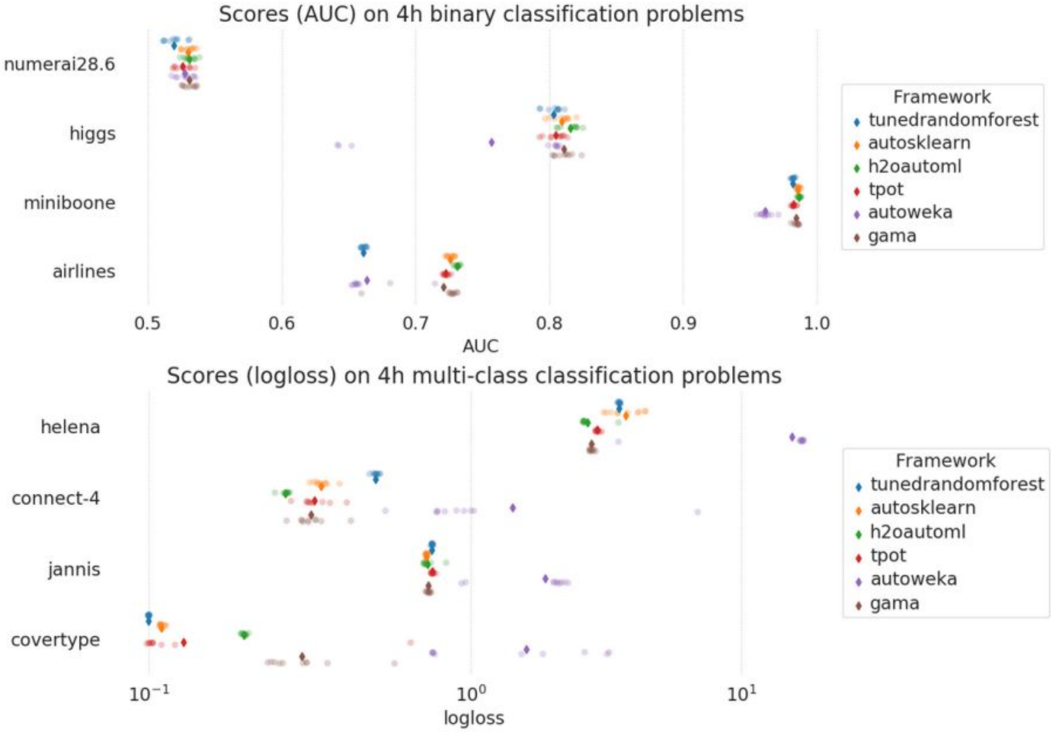
Events



```python
from gama import GamaClassifier

def print_evaluation(evaluation):
    print(f'{evaluation.individual.pipeline_str()} was evaluated. Fitness is {evaluation.score}.')

automl = GamaClassifier()
automl.evaluation_completed(print_evaluation)
automl.fit(X, y)
```

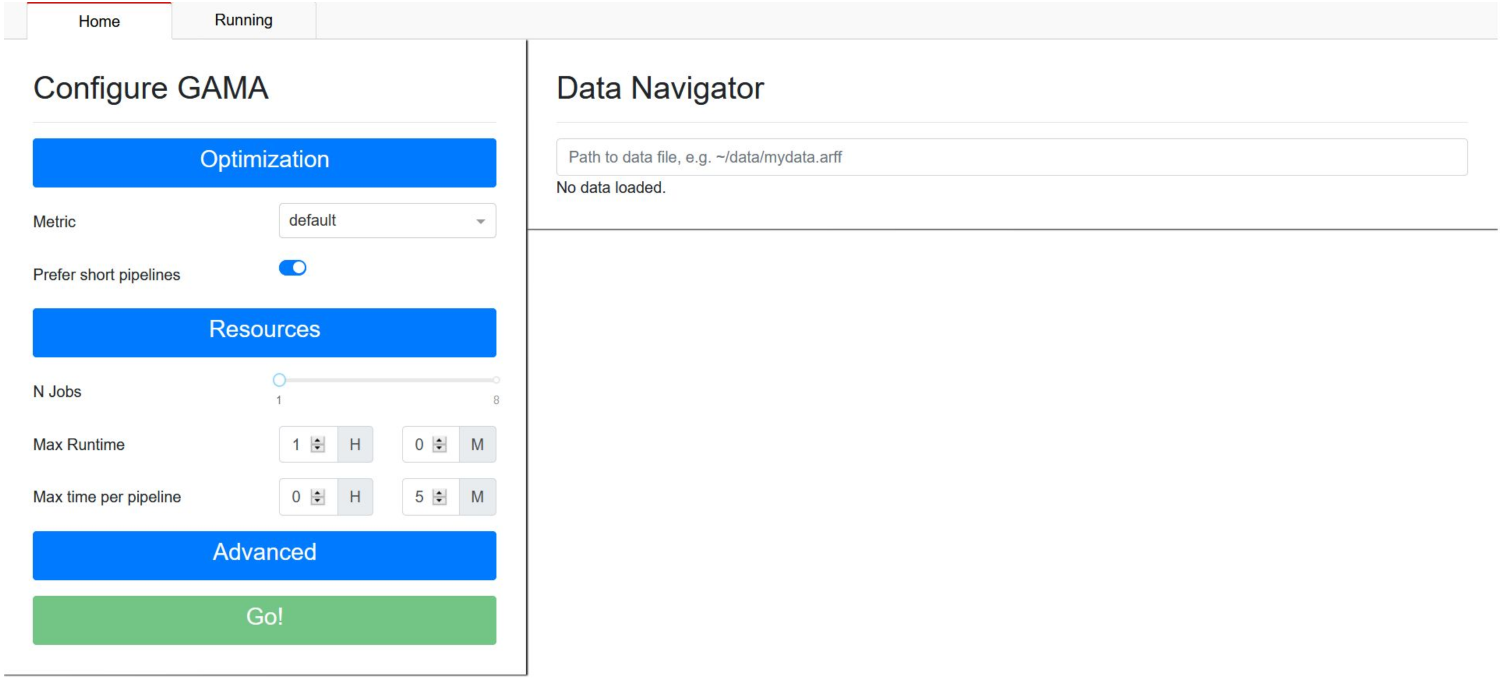
Dodawanie własnych metod wyszukiwania i post-processingu

Benchmarki



Scores (AUC) on 4h binary classification problems
numerai28.6
higgs
miniboone
airlines
0.5
0.6
0.7
0.8
0.9
1.0
AUC
Framework
tunedrandomforest
autosklearn
h2oautoml
tpot
autoweka
gama
Scores (logloss) on 4h multi-class classification problems
helena
connect-4
jannis
covertype
10^{-1}
10^{0}
10^{1}
logloss
Framework
tunedrandomforest
autosklearn
h2oautoml
tpot
autoweka
gama

Dashboard



Home | Running

Configure GAMA

Metric: default

Prefer short pipelines: ●

N Jobs: 1 (range 1–8)

Max Runtime: 1 H 0 M

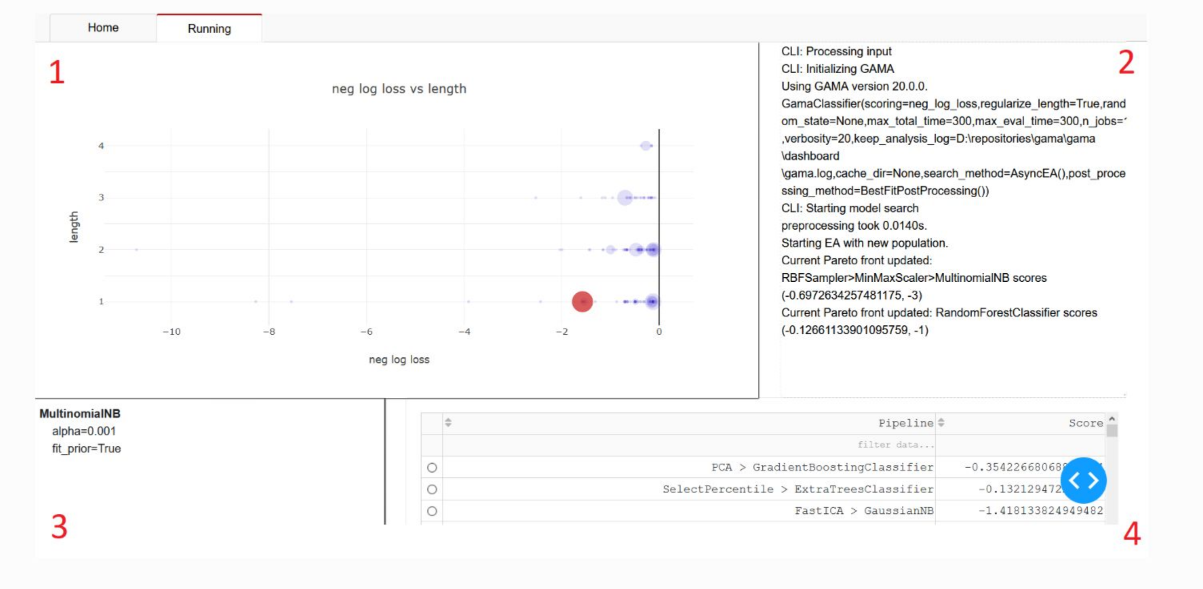
Max time per pipeline: 0 H 5 M

Data Navigator

Path to data file, e.g. ~/data/mydata.arff

No data loaded.

Running tab

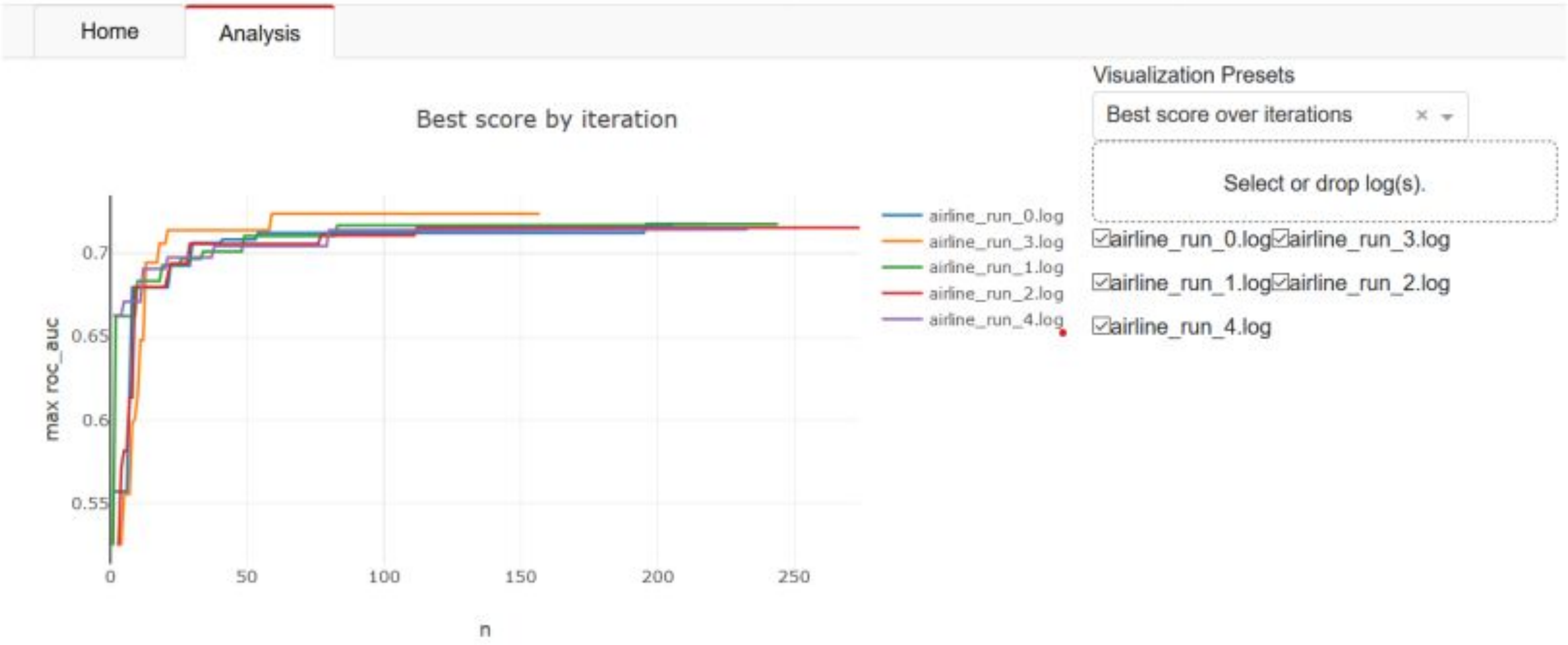


```
CLI: Processing input
CLI: Initializing GAMA
Using GAMA version 20.0.0.
GamaClassifier(scoring=neg_log_loss,regularize_length=True,rand
om_state=None,max_total_time=300,max_eval_time=300,n_jobs=
,verbosity=20,keep_analysis_log=D:\repositories\gama\gama
\dashboard
\gama.log,cache_dir=None,search_method=AsyncEA(),post_proce
ssing_method=BestFitPostProcessing())
CLI: Starting model search
preprocessing took 0.0140s.
Starting EA with new population.
Current Pareto front updated:
RBFSampler>MinMaxScaler>MultinomialNB scores
(-0.6972634257481175, -3)
Current Pareto front updated: RandomForestClassifier scores
(-0.12661133901095759, -1)
```

MultinomialNB
alpha=0.001
fit_prior=True

	Pipeline	Score
	filter data...	
○	PCA > GradientBoostingClassifier	-0.354226680688
○	SelectPercentile > ExtraTreesClassifier	-0.132129472
○	FastICA > GaussianNB	-1.418133824949482

Analysis tab



Visualization Presets

Best score over iterations

Select or drop log(s).

☑airline_run_0.log ☑airline_run_3.log

☑airline_run_1.log ☑airline_run_2.log

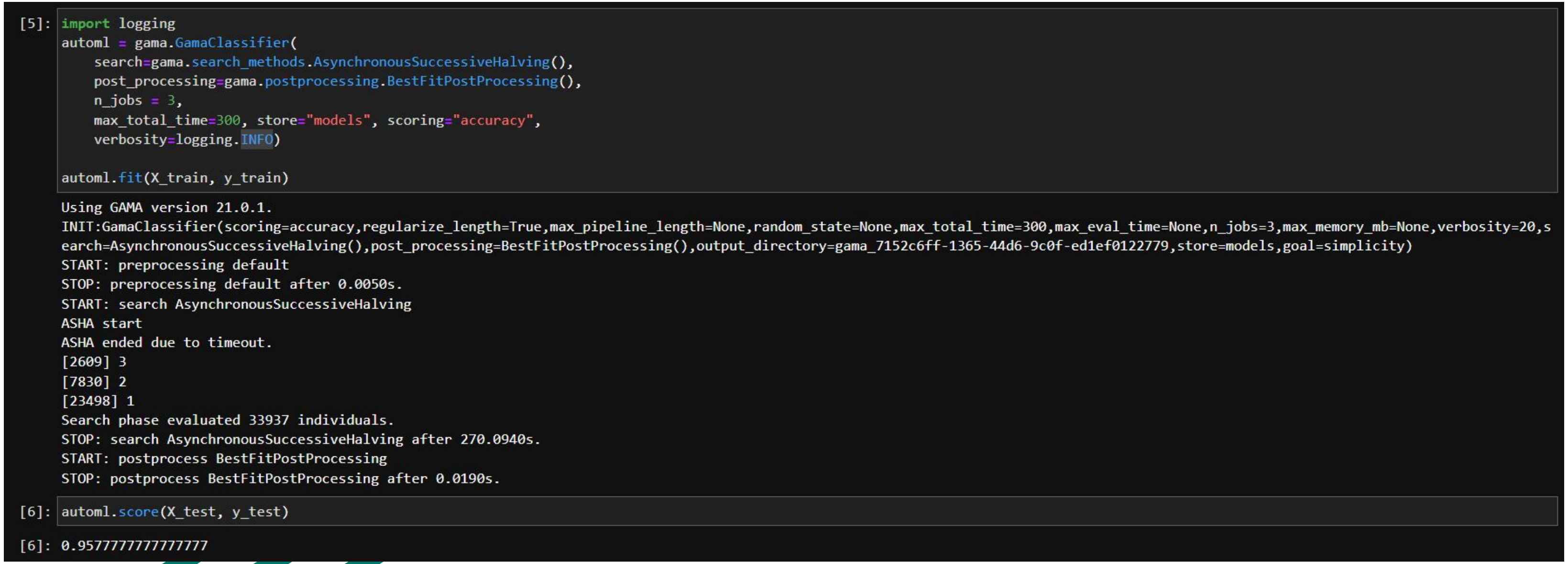
☑airline_run_4.log

Pierwsze doświadczenia

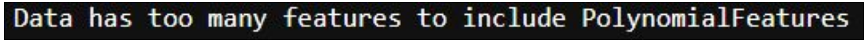
```
pip install -i category_encoders==2.3.0
```

```
pip install -i Werkzeug==2.0.0
```

Pierwsze pozytywne doświadczenia

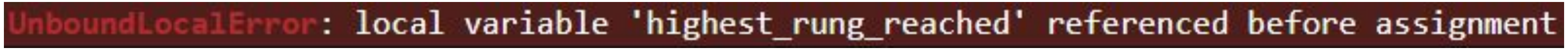


```
[5]: import logging
     automl = gama.GamaClassifier(
         search=gama.search_methods.AsynchronousSuccessiveHalving(),
         post_processing=gama.postprocessing.BestFitPostProcessing(),
         n_jobs = 3,
         max_total_time=300, store="models", scoring="accuracy",
         verbosity=logging.INFO)

     automl.fit(X_train, y_train)

     Using GAMA version 21.0.1.
     INIT:GamaClassifier(scoring=accuracy,regularize_length=True,max_pipeline_length=None,random_state=None,max_total_time=300,max_eval_time=None,n_jobs=3,max_memory_mb=None,verbosity=20,s
     earch=AsynchronousSuccessiveHalving(),post_processing=BestFitPostProcessing(),output_directory=gama_7152c6ff-1365-44d6-9c0f-ed1ef0122779,store=models,goal=simplicity)
     START: preprocessing default
     STOP: preprocessing default after 0.0050s.
     START: search AsynchronousSuccessiveHalving
     ASHA start
     ASHA ended due to timeout.
     [2609] 3
     [7830] 2
     [23498] 1
     Search phase evaluated 33937 individuals.
     STOP: search AsynchronousSuccessiveHalving after 270.0940s.
     START: postprocess BestFitPostProcessing
     STOP: postprocess BestFitPostProcessing after 0.0190s.

[6]: automl.score(X_test, y_test)

[6]: 0.9577777777777777
```

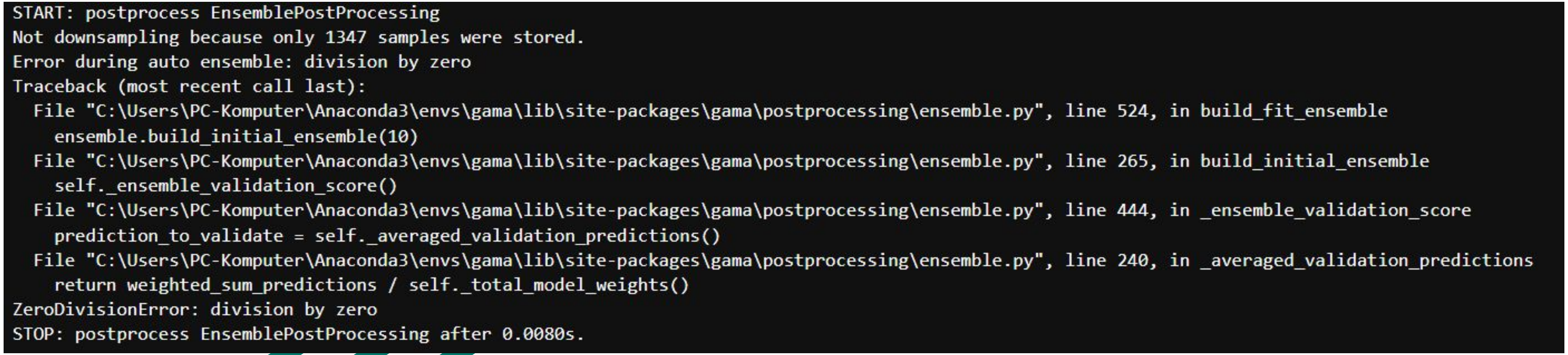


```
Data has too many features to include PolynomialFeatures
```

Kolejne mniej pozytywne niespodzianki

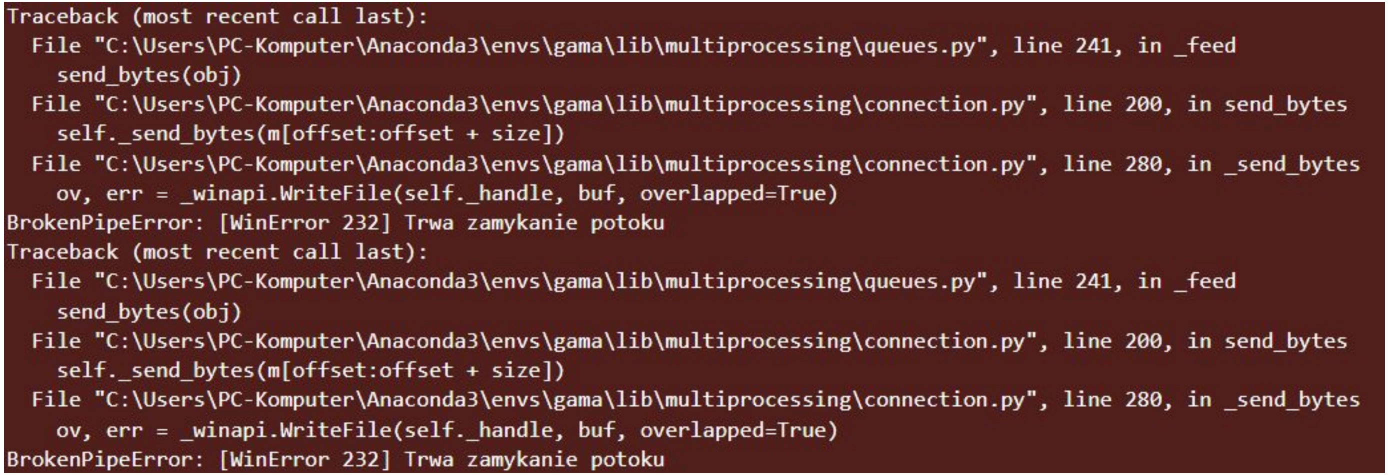


```
UnboundLocalError: local variable 'highest_rung_reached' referenced before assignment
```



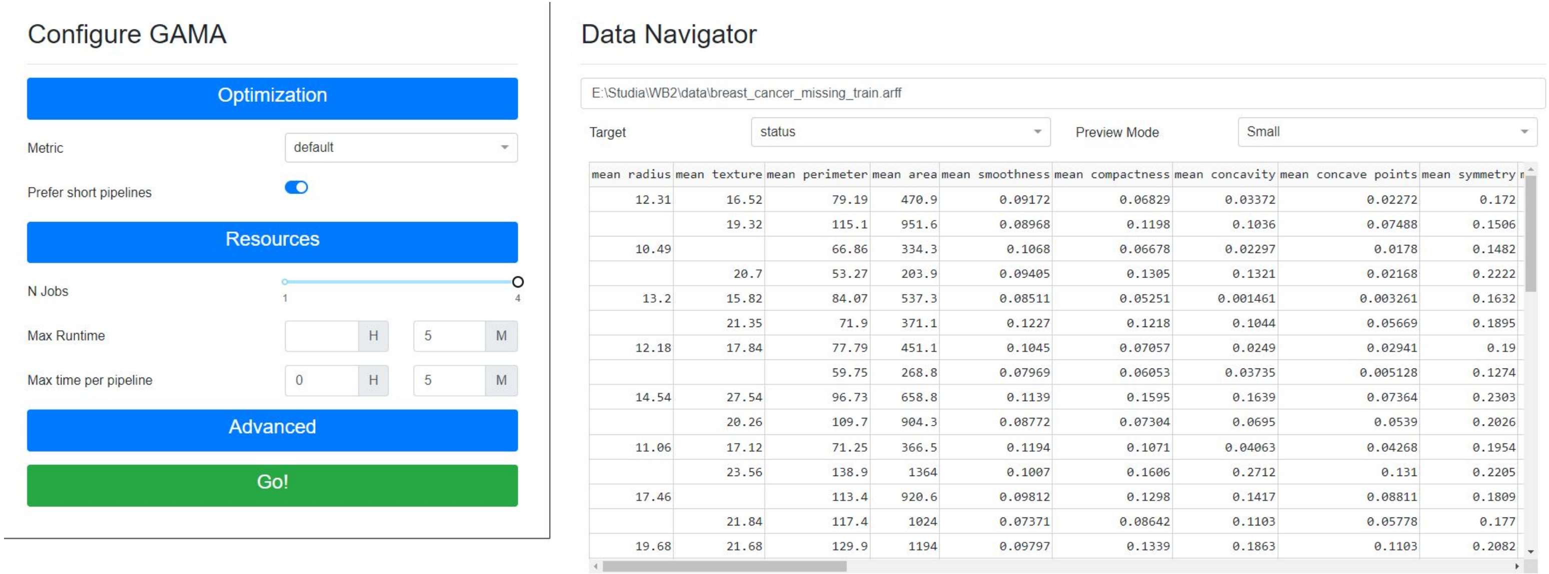
```
START: postprocess EnsemblePostProcessing
Not downsampling because only 1347 samples were stored.
Error during auto ensemble: division by zero
Traceback (most recent call last):
  File "C:\Users\PC-Komputer\Anaconda3\envs\gama\lib\site-packages\gama\postprocessing\ensemble.py", line 524, in build_fit_ensemble
    ensemble.build_initial_ensemble(10)
  File "C:\Users\PC-Komputer\Anaconda3\envs\gama\lib\site-packages\gama\postprocessing\ensemble.py", line 265, in build_initial_ensemble
    self._ensemble_validation_score()
  File "C:\Users\PC-Komputer\Anaconda3\envs\gama\lib\site-packages\gama\postprocessing\ensemble.py", line 444, in _ensemble_validation_score
    prediction_to_validate = self._averaged_validation_predictions()
  File "C:\Users\PC-Komputer\Anaconda3\envs\gama\lib\site-packages\gama\postprocessing\ensemble.py", line 240, in _averaged_validation_predictions
    return weighted_sum_predictions / self._total_model_weights()
ZeroDivisionError: division by zero
STOP: postprocess EnsemblePostProcessing after 0.0080s.
```

Jeszcze mniej pozytywne niespodzianki



```
Traceback (most recent call last):
  File "C:\Users\PC-Komputer\Anaconda3\envs\gama\lib\multiprocessing\queues.py", line 241, in _feed
    send_bytes(obj)
  File "C:\Users\PC-Komputer\Anaconda3\envs\gama\lib\multiprocessing\connection.py", line 200, in send_bytes
    self._send_bytes(m[offset:offset + size])
  File "C:\Users\PC-Komputer\Anaconda3\envs\gama\lib\multiprocessing\connection.py", line 280, in _send_bytes
    ov, err = _winapi.WriteFile(self._handle, buf, overlapped=True)
BrokenPipeError: [WinError 232] Trwa zamykanie potoku
Traceback (most recent call last):
  File "C:\Users\PC-Komputer\Anaconda3\envs\gama\lib\multiprocessing\queues.py", line 241, in _feed
    send_bytes(obj)
  File "C:\Users\PC-Komputer\Anaconda3\envs\gama\lib\multiprocessing\connection.py", line 200, in send_bytes
    self._send_bytes(m[offset:offset + size])
  File "C:\Users\PC-Komputer\Anaconda3\envs\gama\lib\multiprocessing\connection.py", line 280, in _send_bytes
    ov, err = _winapi.WriteFile(self._handle, buf, overlapped=True)
BrokenPipeError: [WinError 232] Trwa zamykanie potoku
```

Zadziwiające pozytywne doświadczenia



Configure GAMA

Metric: default

Prefer short pipelines: ●

N Jobs: 4 (1–4)

Max Runtime: H 5 M

Max time per pipeline: 0 H 5 M

Data Navigator

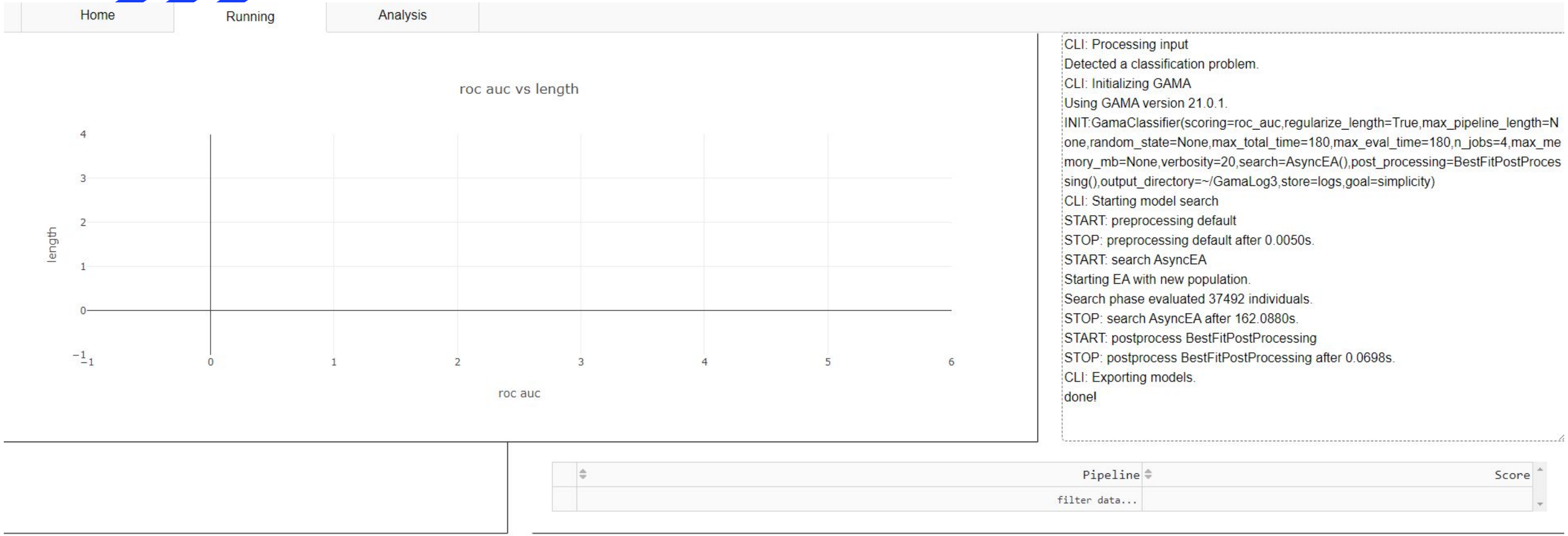
E:\Studia\WB2\data\breast_cancer_missing_train.arff

Target: status

Preview Mode: Small

mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry
12.31	16.52	79.19	470.9	0.09172	0.06829	0.03372	0.02272	0.172
	19.32	115.1	951.6	0.08968	0.1198	0.1036	0.07488	0.1506
10.49		66.86	334.3	0.1068	0.06678	0.02297	0.0178	0.1482
	20.7	53.27	203.9	0.09405	0.1305	0.1321	0.02168	0.2222
13.2	15.82	84.07	537.3	0.08511	0.05251	0.001461	0.003261	0.1632
	21.35	71.9	371.1	0.1227	0.1218	0.1044	0.05669	0.1895
12.18	17.84	77.79	451.1	0.1045	0.07057	0.0249	0.02941	0.19
		59.75	268.8	0.07969	0.06053	0.03735	0.005128	0.1274
14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303
	20.26	109.7	904.3	0.08772	0.07304	0.0695	0.0539	0.2026
11.06	17.12	71.25	366.5	0.1194	0.1071	0.04063	0.04268	0.1954
	23.56	138.9	1364	0.1007	0.1606	0.2712	0.131	0.2205
17.46		113.4	920.6	0.09812	0.1298	0.1417	0.08811	0.1809
	21.84	117.4	1024	0.07371	0.08642	0.1103	0.05778	0.177
19.68	21.68	129.9	1194	0.09797	0.1339	0.1863	0.1103	0.2082

Które okazało się złudne



```
CLI: Processing input
Detected a classification problem.
CLI: Initializing GAMA
Using GAMA version 21.0.1.
INIT:GamaClassifier(scoring=roc_auc,regularize_length=True,max_pipeline_length=None,random_state=None,max_total_time=180,max_eval_time=180,n_jobs=4,max_memory_mb=None,verbosity=20,search=AsyncEA(),post_processing=BestFitPostProcessing(),output_directory=~/GamaLog3,store=logs,goal=simplicity)
CLI: Starting model search
START: preprocessing default
STOP: preprocessing default after 0.0050s.
START: search AsyncEA
Starting EA with new population.
Search phase evaluated 37492 individuals.
STOP: search AsyncEA after 162.0880s.
START: postprocess BestFitPostProcessing
STOP: postprocess BestFitPostProcessing after 0.0698s.
CLI: Exporting models.
done!
```

Pipeline	Score
filter data...	

Dziękujemy za obejrzenie prezentacji

Tomasz Siudalski, Grzegorz Zbrzeżny, Piotr Marciniak

Źródła:

- GAMA: a General Automated Machine learning Assistant
- https://github.com/openml-labs/gama
- https://openml-labs.github.io/gama/master