



ОНЛАЙН-ОБРАЗОВАНИЕ


Онлайн-образование

Проверить, идет ли запись!





Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы

Automated Machine Learning



Пантелеев Максим Александрович

RnD TeamLead
maxim.al.panteleev@gmail.com

Правила вебинара



Активно участвуем



Задаем вопрос в чат или голосом

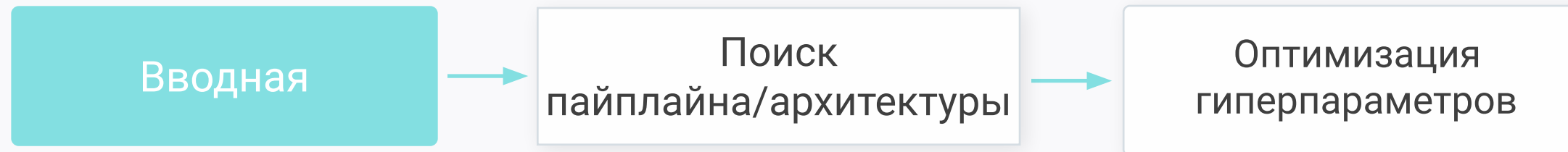


Off-topic обсуждаем в Slack #канал группы или #general



Вопросы вижу в чате, могу ответить не сразу

Маршрут вебинара



Цели вебинара | После занятия вы сможете

1 использовать automl фреймворки для решения data science задач

2 улучшить текущие решения с помощью подбора гиперпараметров

Смысл | Зачем вам это уметь

1

использовать в соревнованиях и хакатонах

2

использовать на работе

The background of the slide features an aerial view of a dense city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer that contains a white network pattern of interconnected dots and lines, suggesting a digital or technological theme. The text is centered within this blue band.

Основная информация

Определение AutoML

Let:

$\{A^{(1)}, A^{(2)}, \dots, A^{(n)}\}$ be a set of *algorithms* (operators)

$\Lambda^{(i)} = \lambda_1 \times \lambda_2 \times \dots \times \lambda_m$ be the *hyperparameter space* for $A^{(i)}$

\mathcal{A} be the space of possible *architectures* of one or more algorithms

$\Lambda_{\mathcal{A}} = \Lambda^{(1)} \times \Lambda^{(2)} \times \dots \times \Lambda^{(m)}$ its combined *configuration space*

$\lambda \in \Lambda_{\mathcal{A}}$ a specific *configuration* (of architecture and hyperparameters)

$\mathcal{L}(\lambda, D_{train}, D_{valid})$ the loss of the model created by λ , trained on data D_{train} , and validated on data D_{valid}

Find the configuration that minimizes the expected loss on a dataset \mathcal{D} :

$$\lambda^* = \operatorname{argmin}_{\lambda \in \Lambda_{\mathcal{A}}} \mathbb{E}_{(D_{train}, D_{valid}) \sim \mathcal{D}} \mathcal{L}(\lambda, D_{train}, D_{test})$$

Типы гиперпараметров

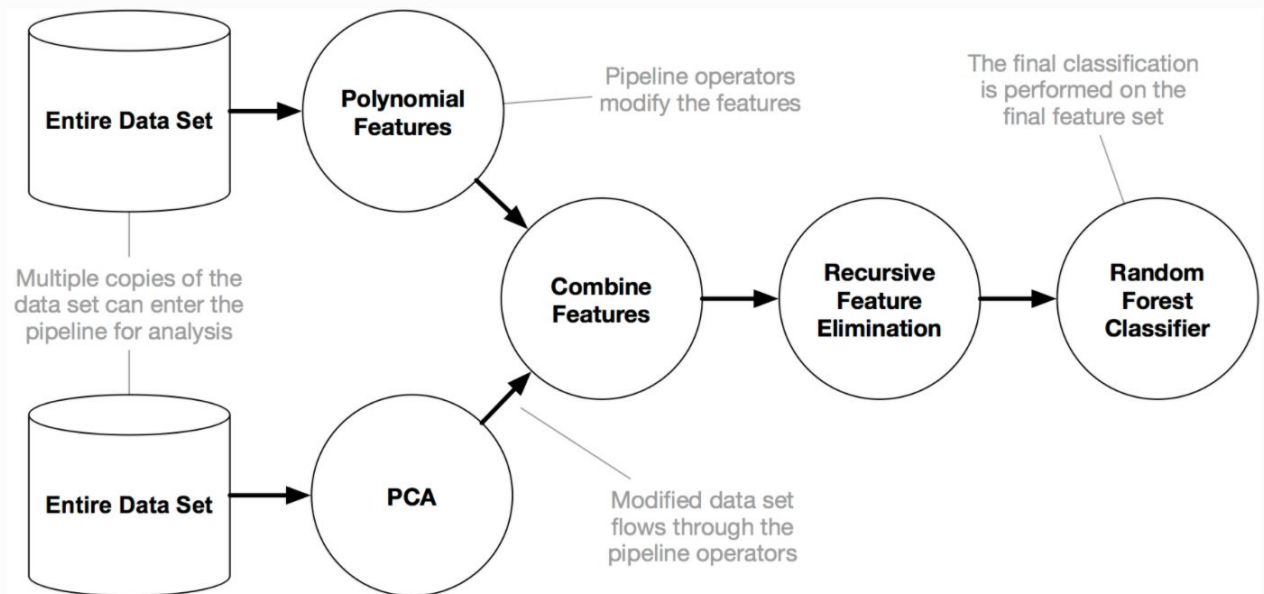
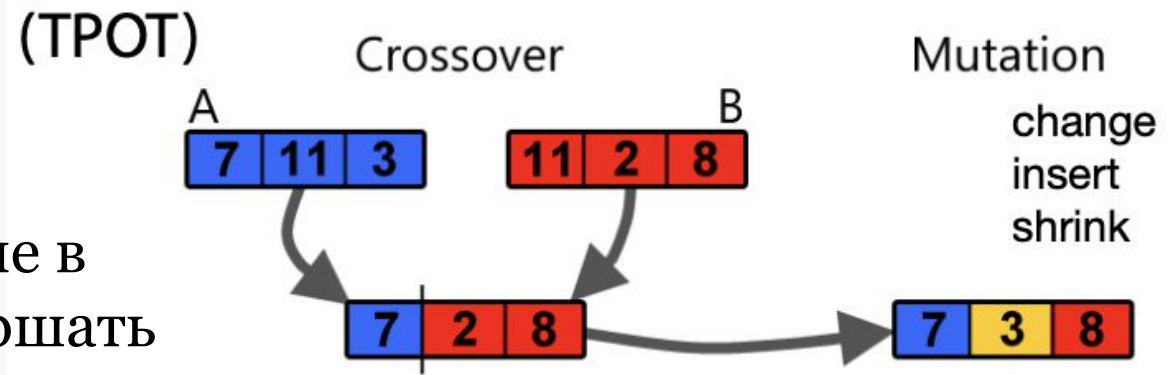
- Непрерывные (e.g. learning rate, SVM_C,...)
- Целочисленные (e.g. number of hidden units, number of boosting iterations,...)
- Категориальные
 - e.g. choice of algorithm (SVM, RandomForest, Neural Net,...) •
 - e.g. choose of operator (Convolution, MaxPooling, DropOut,...)
 - e.g. activation function (ReLU, Leaky ReLU, tanh,...)
- Conditional
 - e.g. SVM kernel if SVM is selected, kernel width if RBF kernel is selected
 - e.g. Convolution kernel size if Convolution layer is selected

Архитектура vs гиперпараметры

- Поиск архитектуры: поиска в пространстве всех возможных архитектур
 - Pipelines: Fixed predefined pipeline, grammars, genetic programming, planning, Monte-Carlo Tree Search
 - Neural architectures
- Оптимизация гиперпараметров:
 - Optimization: grid/random search, Bayesian optimization, evolution, multi-armed bandits, gradient descent (only NNs)
 - Meta-learning
- Могут решаться последовательно, одновременно или чередоваться

Pipeline search: генетическое программирование

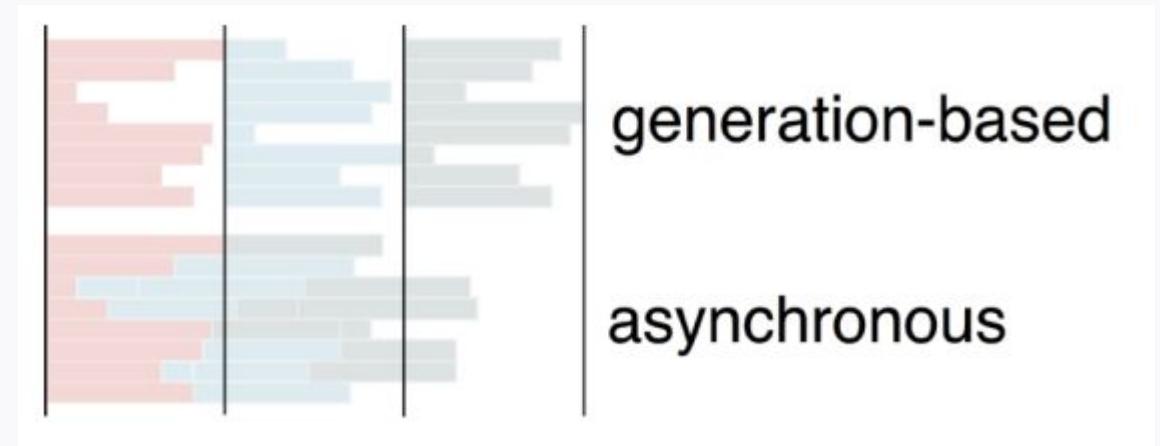
- Tree-based pipeline optimization (TPOT)
 - Инициализируем случайные пайплайны, оцениваем, лучшие в каждом поколении будут совершать cross-over или мутировать
 - GP примитивы включают копирование данных и объединение фичей
 - Multi-objective optimization: accurate but short
 - Легко распараллелить



An example TPOT pipeline

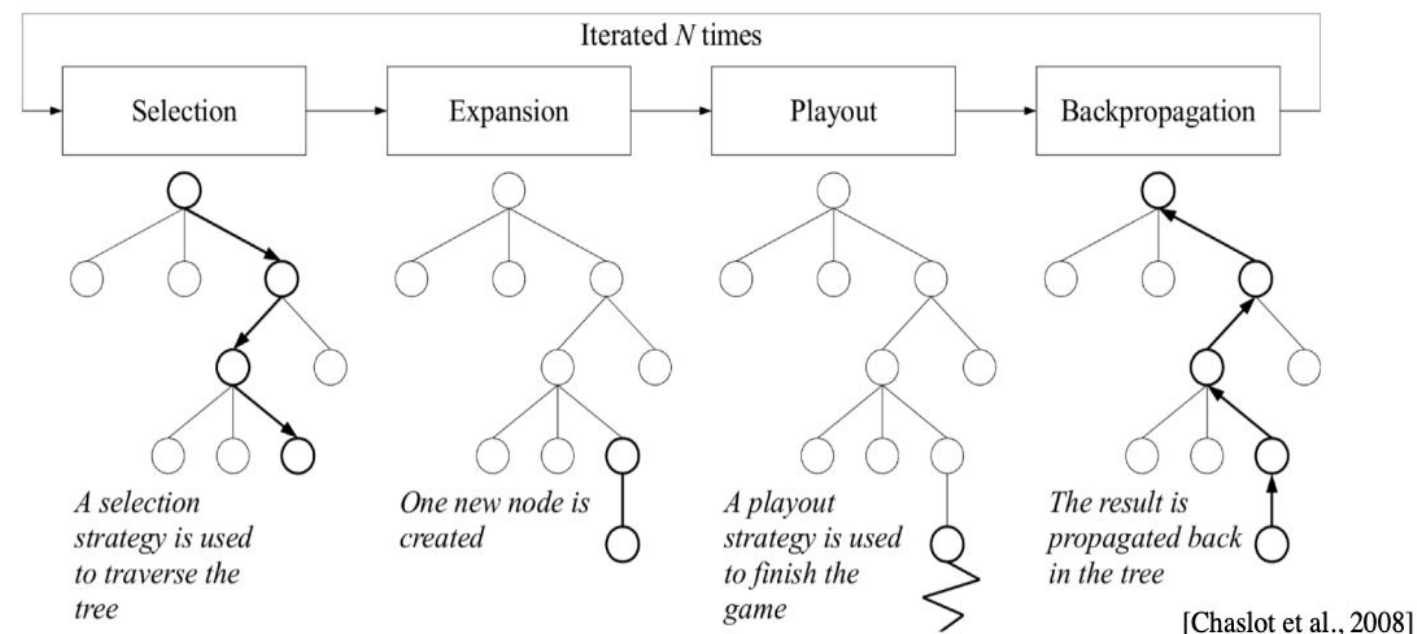
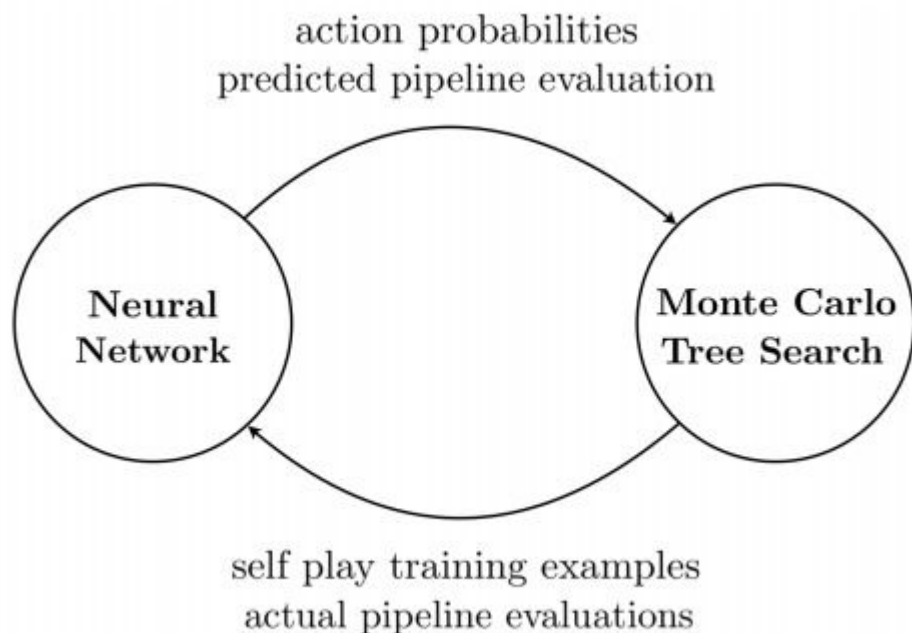
Pipeline search: генетическое программирование

- Asynchronous evolution (GAMA)
 - Не обязательно дожидаться оценивая каждого пайплайна в поколении
 - GAMA позволяет менять стратегию поиска, добавлять ассамблирование пайплайнов в качестве пост-процессинга и визуализировать процесс оптимизации



Pipeline search: Monte Carlo Tree Search

- Use MCTS to search for optimal pipelines
 - Optimize the structure and hyperparameters simultaneously by building a surrogate model to predict configuration performance
 - Bayesian surrogate model: MOSAIC
 - Neural network: [AlphaD3M](#)



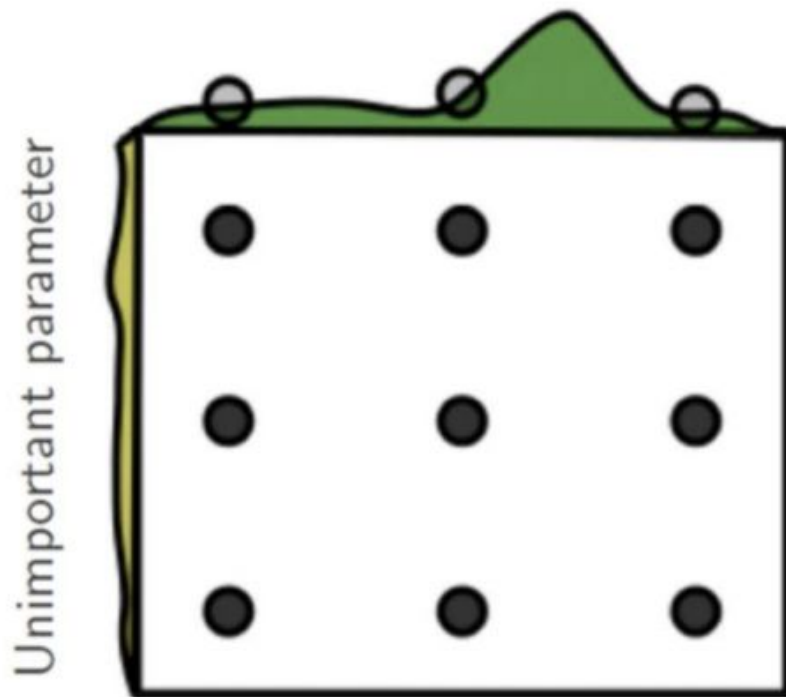
Оптимизация гиперпараметров (HPO)



$$\lambda^* = \operatorname{argmin}_{\lambda \in \Lambda_{\mathcal{A}}} \mathbb{E}_{(D_{train}, D_{valid}) \sim \mathcal{D}} \mathcal{L}(\lambda, D_{train}, D_{test})$$

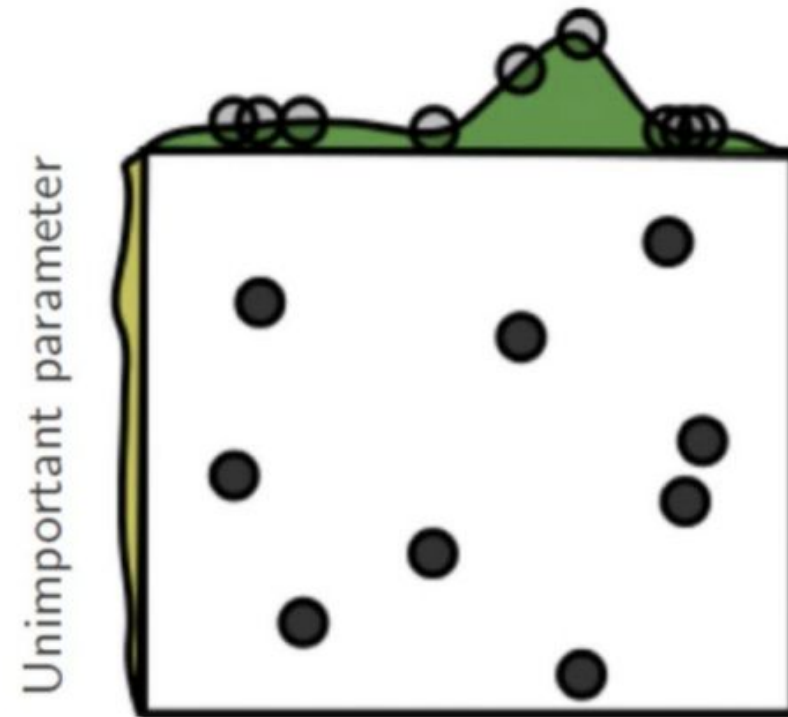
HPO: grid and random search

Grid Layout



Important parameter

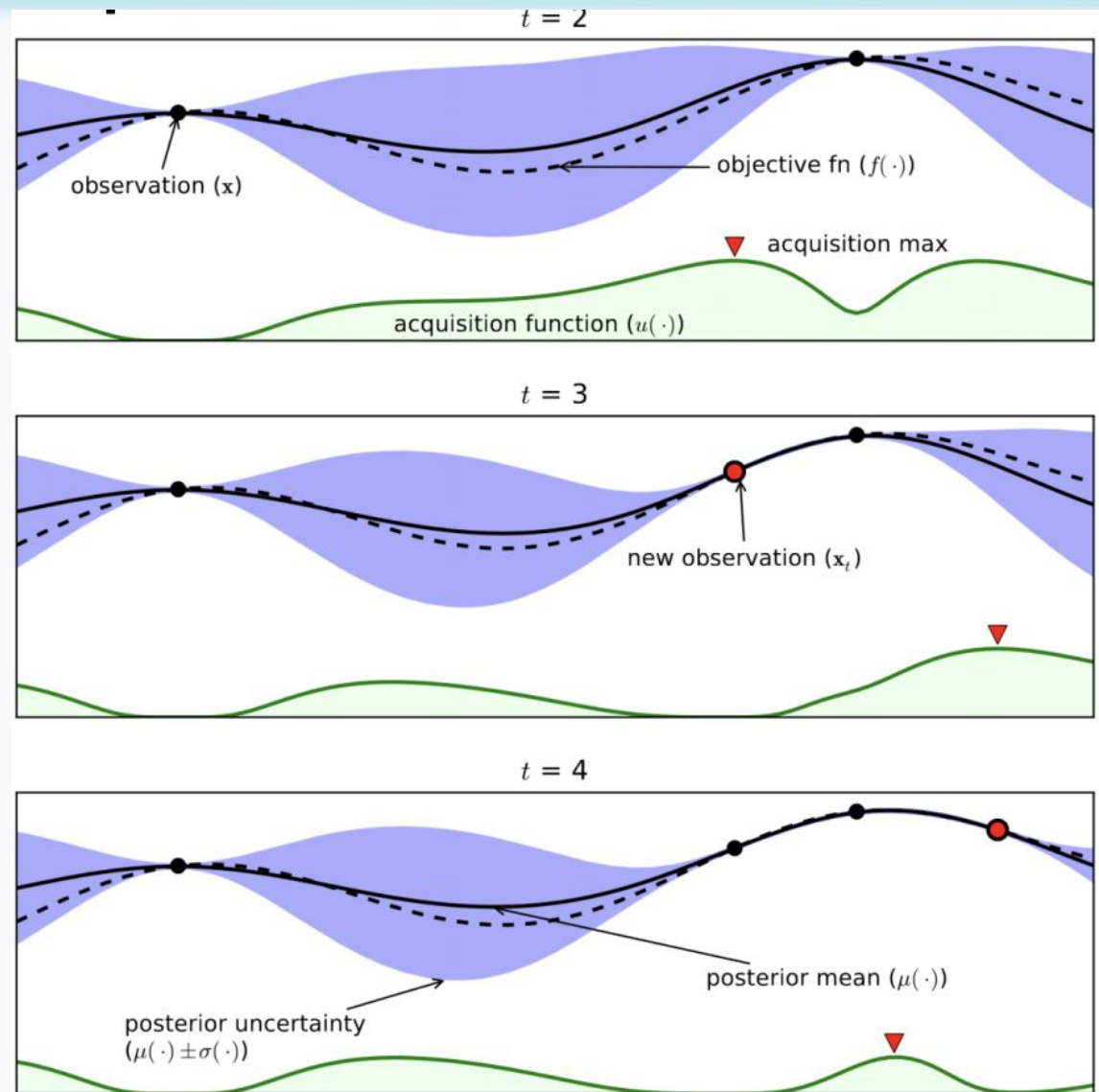
Random Layout



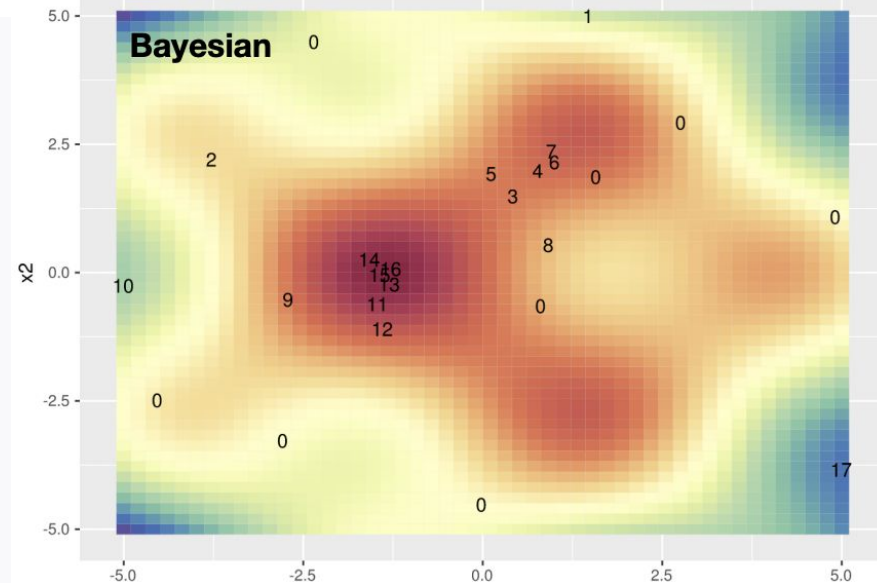
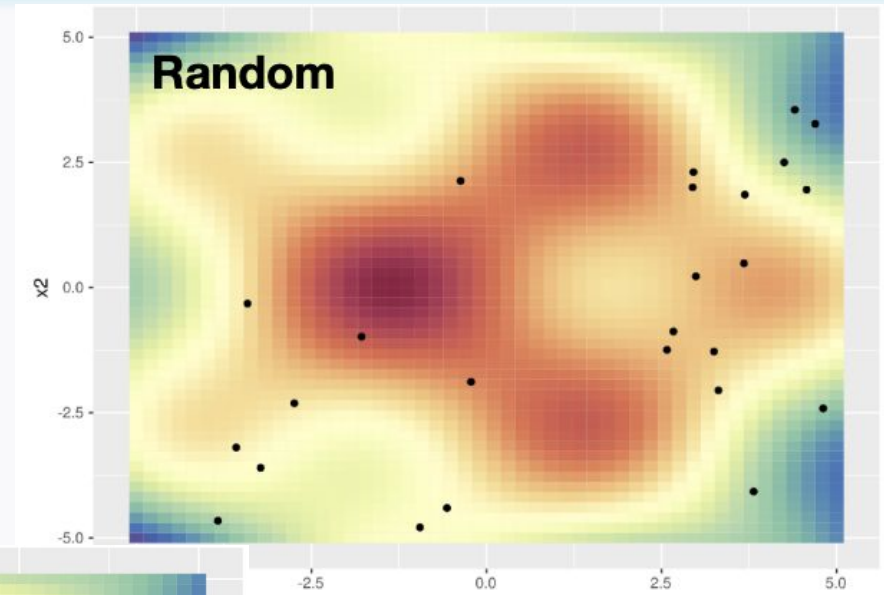
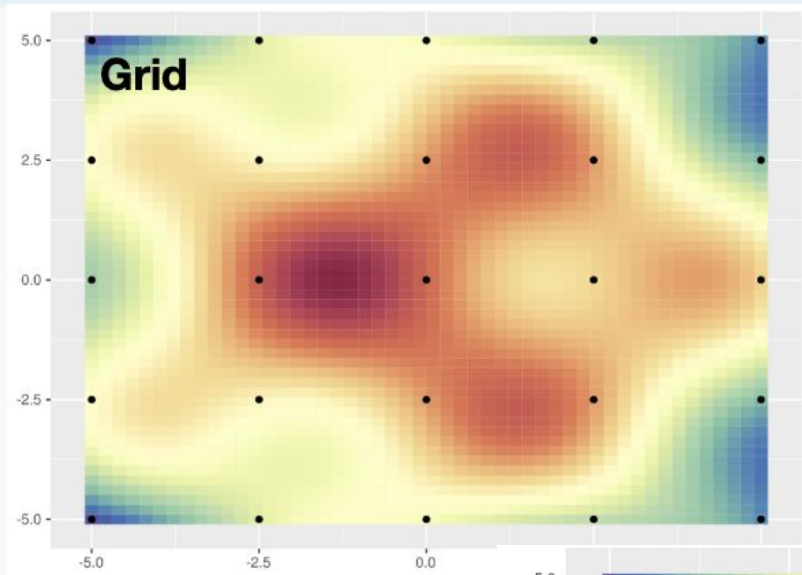
Important parameter

HPO: Bayesian Optimization

- Начинаем с нескольких случайных наборов гиперпараметров
- Строим суррогатную модель, предсказывающую качество на остальных конфигурациях: mean and standard deviation (синяя полоса)
 - Любая вероятностная регрессионная модель: e.g. Gaussian processes
- Выбор следующей точки осуществляем с помощью функции предпочтения, e.g. Expected Improvement (EI)



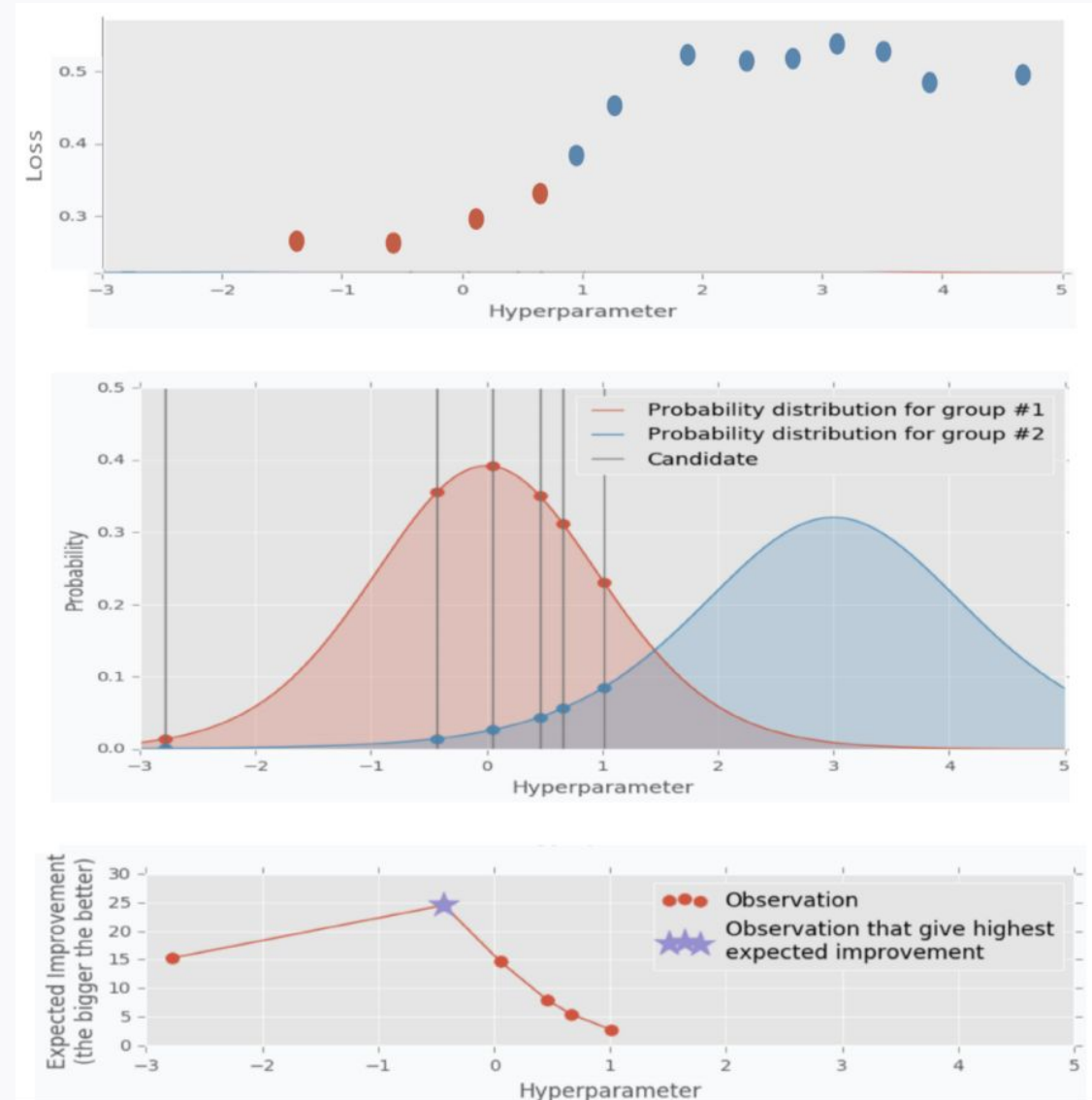
HPO: Bayesian Optimization



HPO: Tree of Parzen Estimators

1. Начинаем с нескольких случайных наборов гиперпараметров
2. Отделяем **хорошие** and **плохие** hyperparameters
3. Делаем непараметрическую оценку плотности распределений $p(\lambda = \text{good})$ и $p(\lambda = \text{bad})$
4. Сэмплируем с наибольшим отношением $p(\lambda = \text{good}) / p(\lambda = \text{bad})$

Efficient, parallelizable, robust, but less sample efficient than GPs



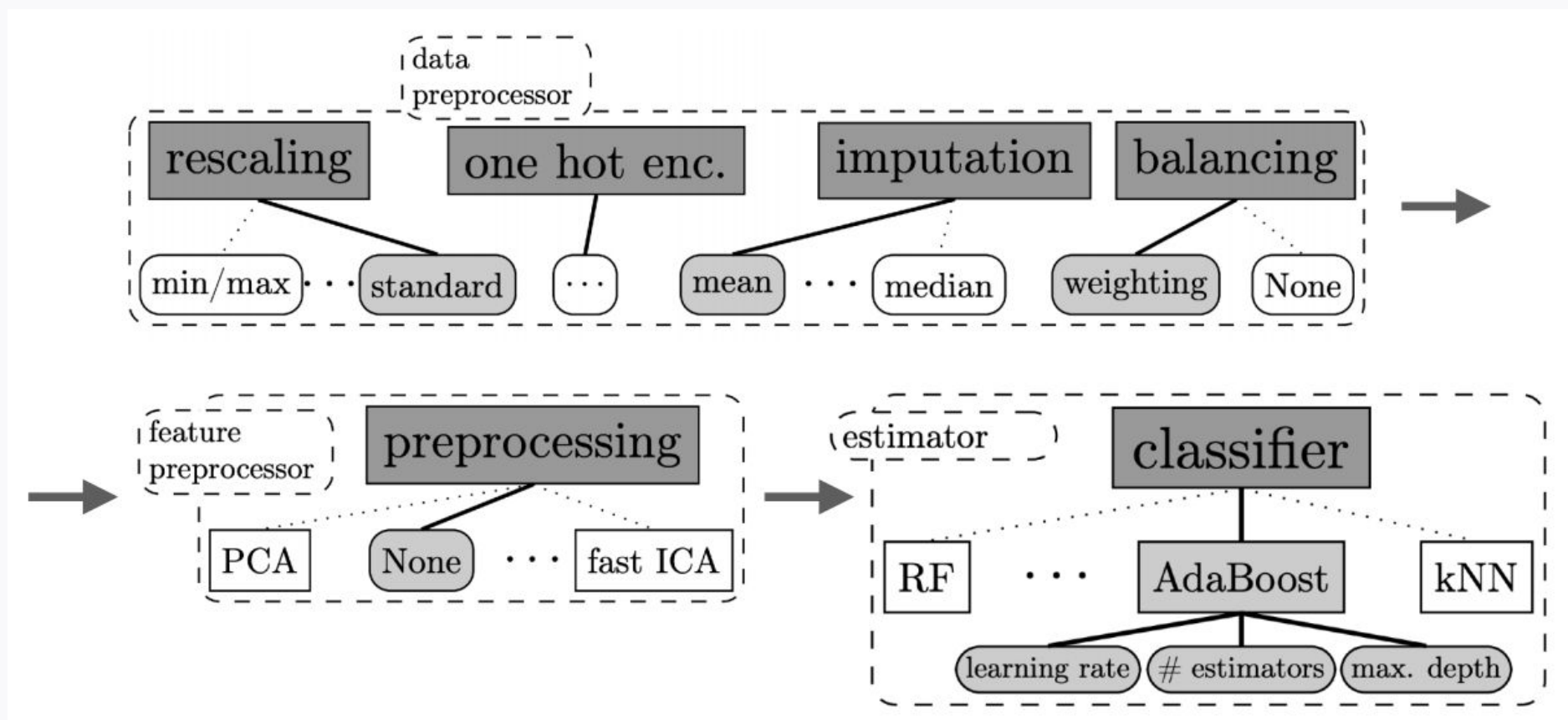
HyperBand

- Create several buckets, each having n randomly generated hyperparameter configurations, each configuration using r resources (e.g., epoch number, batch number).
- After the n configurations are finished, it chooses the top n/η configurations and runs them using increased $r*\eta$ resources.
- At last, it chooses the best configuration it has found so far.

i	$s = 4$		$s = 3$		$s = 2$	
	n_i	r_i	n_i	r_i	n_i	r_i
0	81	1	27	3	9	9
1	27	3	9	9	3	27
2	9	9	3	27	1	81
3	3	27	1	81		
4	1	81				

Pipeline search: параметризованный пайплайн

- Вводим conditional гиперпараметры
- Combined Algorithm Selection and Hyperparameter optimization (CASH):

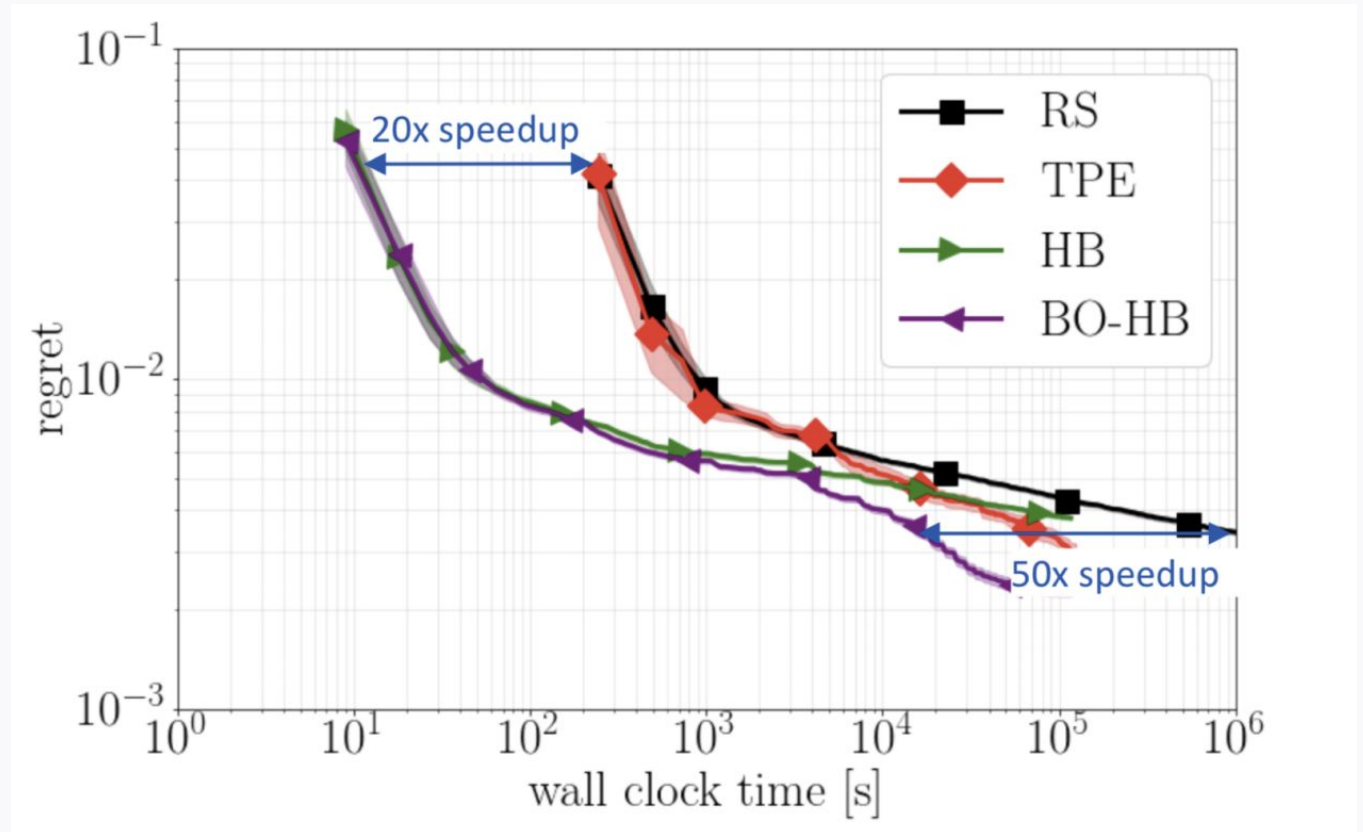


Auto-Sklearn

- Parameterized pipeline
 - 15 classifiers, 16 feature preprocessing methods and numerous data preprocessing methods
- Bayesian Optimization (RF)
- Ensembling
- warm-start
 - from k nearest datasets, where the distance between datasets is defined as the L1 distance on meta-features

BO-HB и Hyperopt-Sklearn

- Hyperopt-Sklearn
 - Parameterized pipeline
 - Tree of Parzen Estimators
- BO-HB
 - Parameterized pipeline
 - Tree of Parzen Estimators
 - Ensembling
 - Hyperband



AutoGluon (Tabular)

Data preparation:

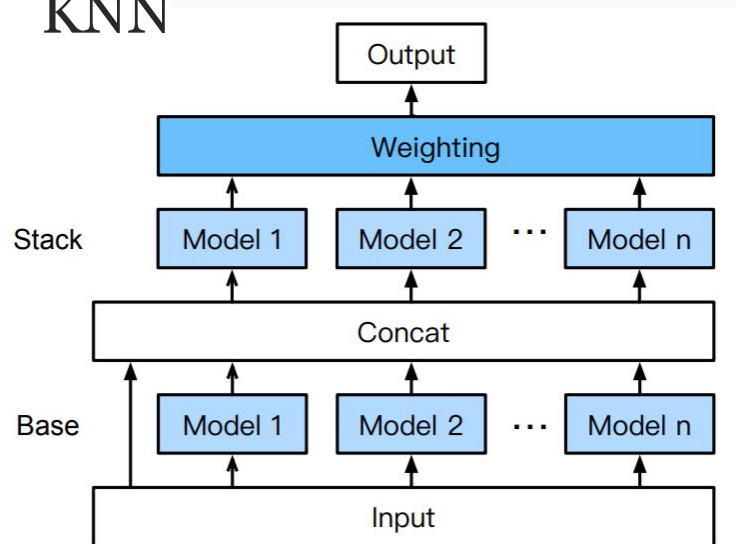
- Для категориальных признаков создаются эмбединги
- N-gram features для текстовых признаков
- Date/time features приводятся в числовой вид
- Пропущенные значения заполняются средним(числовые) или отдельной категорией
- Числовые признаки нормализуются

AutoGluon (Tabular)

Model considered:

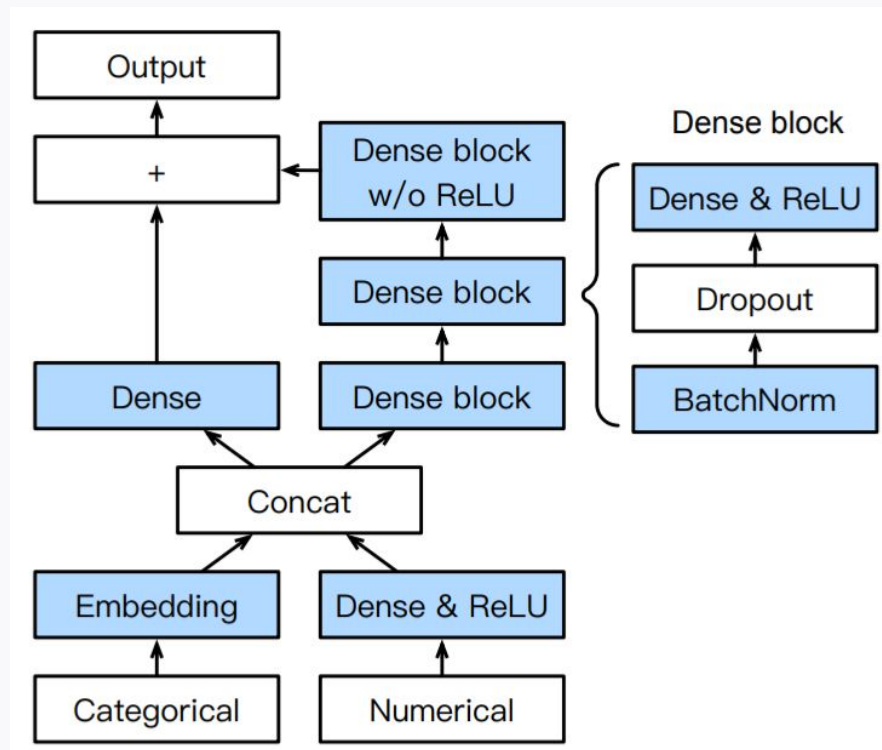
Models include

- neural networks
- LightGBM boosted trees
- CatBoost boosted trees
- Random Forests
- Extremely Randomized Trees
- KNN



Model ensemble:

multiple base models, whose outputs are concatenated and then fed into the next layer, which itself consists of multiple stacker models



H2O AutoML

Model considered:

- three pre-specified XGBoost GBM models
- fixed grid of GLMs
- default Random Forest (DRF)
- five pre-specified H2O GBMs
- near-default Deep Neural Net
- Extremely Randomized Forest (XRT)
- random grid of XGBoost GBMs
- random grid of H2O GBMs
- random grid of Deep Neural Nets.

Model ensemble:

H2O AutoML trains two **Stacked** Ensemble models:

- один ансамбль содержит все модели
- второй содержит только лучшую модель из класса/семейства

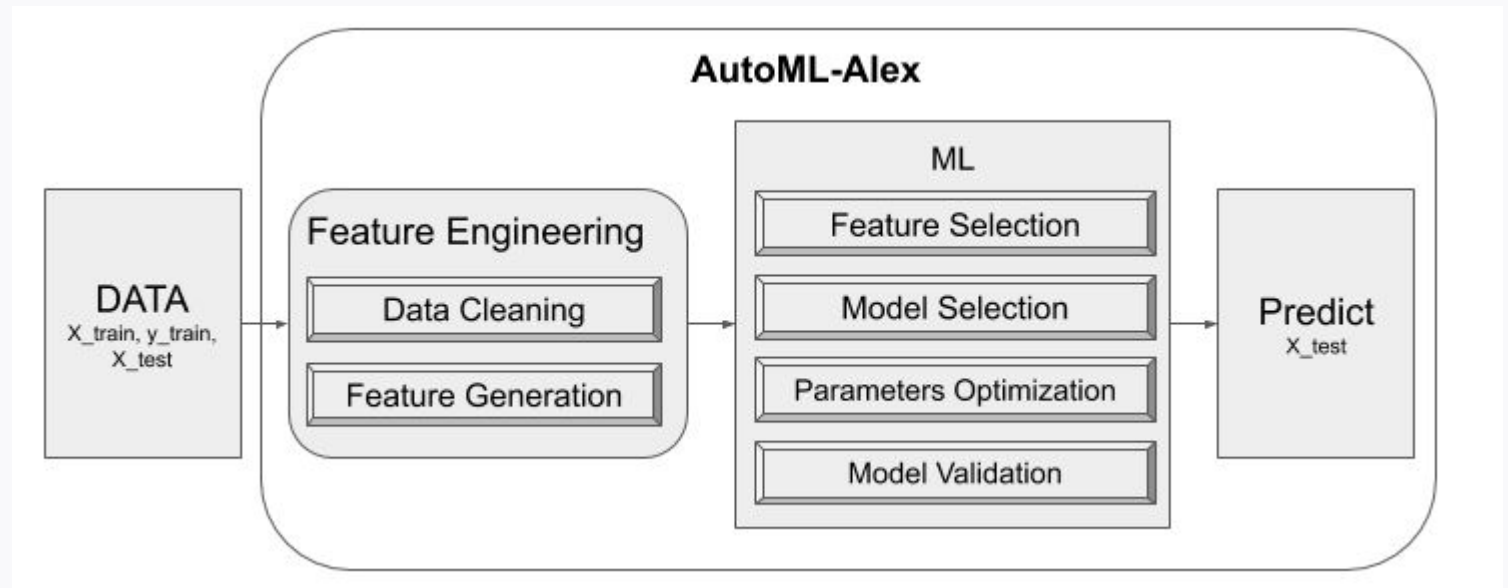
H2O AutoML

model_id	auc	logloss	aucpr	mean_per_class_error	rmse	mse	training_time_ms	predict_time_per_row_ms
StackedEnsemble_AllModels_AutoML_20191213_174603	0.789844	0.551067	0.804672	0.314665	0.432045	0.186663	924	0.05695
StackedEnsemble_BestOfFamily_AutoML_20191213_174603	0.789768	0.550906	0.805696	0.313059	0.431977	0.186604	639	0.024567
XGBoost_grid__1_AutoML_20191213_174603_model_4	0.784698	0.55681	0.80312	0.323143	0.434743	0.189002	3092	0.002083
XGBoost_3_AutoML_20191213_174603	0.784232	0.557749	0.802341	0.317933	0.434976	0.189204	2878	0.002173
XGBoost_2_AutoML_20191213_174603	0.783533	0.555997	0.803189	0.32475	0.434678	0.188945	4635	0.003292
XGBoost_grid__1_AutoML_20191213_174603_model_3	0.782582	0.560218	0.800749	0.34334	0.435944	0.190047	2695	0.002269
GBM_5_AutoML_20191213_174603	0.78219	0.558353	0.800234	0.319658	0.435512	0.18967	768	0.004318

AutoML Alex

Features:

- Automated Data Clean (Auto Clean)
- Automated Feature Engineering (Auto FE)
- Smart Hyperparameter Optimization (HPO)
- Feature Generation
- Feature Selection
- Models Selection
- Cross Validation
- Timelimit and EarlyStopping





LIVE



Основные мысли

- 1 Сделать фичи
- 2 Запустить automl
- 3 ...
- 4 ...
- 5 ...
- 6 PROFIT




Рефлексия



С какими основными мыслями и инсайтами уходите с вебинара?



Достигли ли вы цели вебинара?

The background of the image is an aerial photograph of a city with many skyscrapers, overlaid with a semi-transparent blue layer. A network of white lines connects various points across the blue area, creating a digital or technological aesthetic. The text is centered within this blue band.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате

Спасибо за внимание!
Приходите на следующие вебинары



Пантелеев Максим Александрович

RnD TeamLead

maxim.al.panteleev@gmail.com