



Machine Learning. Professional

H2O и TPOT - а вы что, за меня и
модели строить будете?



Меня хорошо видно && слышно?



Ставим "+", если все хорошо
"-", если есть проблемы



Тема вебинара

Н2О и ТРОТ - а вы что, за меня и модели строить будете?



Алексей Кисляков

Заведующий научной лабораторией «Кластерный анализ процессов роста и эволюции систем», доктор экон. наук., кандидат техн. наук, доцент.

О себе: преподаватель/ученый-исследователь

- > 10 лет научных исследований в области прикладных методов экономико-математического моделирования с применением инструментария Python, R, MATLAB, Excel
- > 8 лет преподавания в ВУЗах оффлайн и онлайн
- > 30 созданных и реализованных курсов по программам высшего образования и дополнительного профессионального образования

Телефон / эл. почта / соц. сети: +7(904) 261-57-18, ankislyakov@mail.ru

Цели вебинара

познакомиться с библиотеками: H2O и TPOT, Automatic Model Selection and Pipeline Building.

К концу занятия вы сможете:

1. Познакомиться с основными инструментами автоматизации отбора моделей машинного обучения
2. Познакомиться с основными инструментами автоматизации построения пайплайнов
3. Научиться работать с библиотеками для автоматического построения и отбора моделей
4. Научиться работать с библиотеками H2O и TPOT для автоматического построения пайплайнов моделей



Где пригодятся полученные знания?

1. Если вы начинающий специалист DS, AutoML упрощает процесс разработки модели и сократит количество шагов, которые необходимо принять при обучении модели.

2. Если у вас есть опыт работы с машинным обучением, вы можете настраивать гиперпараметры моделей, предоставляемые AutoML, в зависимости от ваших потребностей, используя при этом возможности автоматизации.

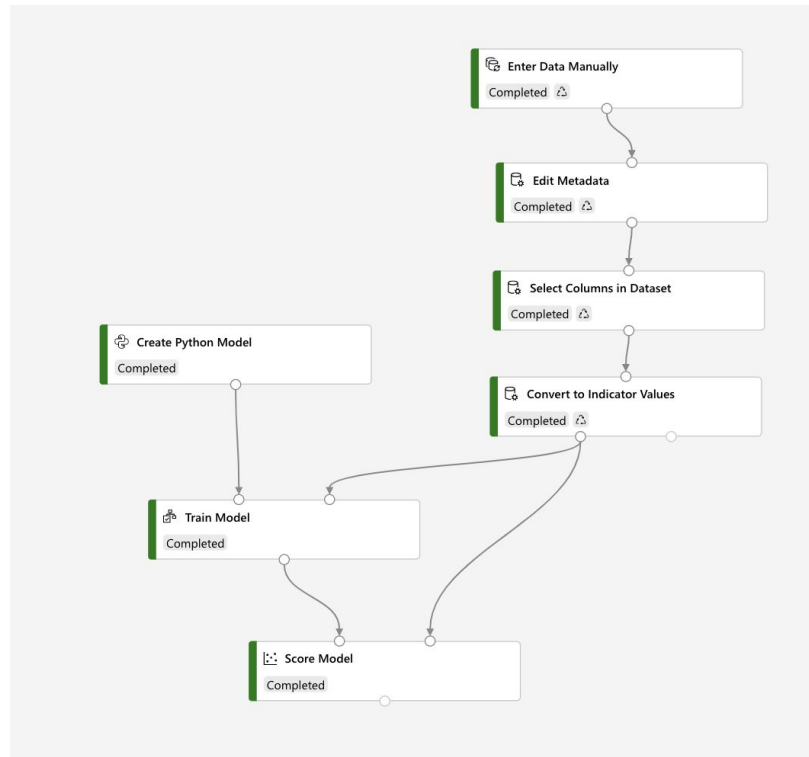
Предпосылки использования автоматизации ML

Основные определения

Пайплайн модели ML – Последовательные стадии работы с данными, включающие извлечение, очистку, разведочный анализ данных (EDA), моделирование, интерпретацию и пересмотр.

Автоматическое машинное обучение (AutoML) представляет собой процесс автоматизации трудоемких и многократно повторяющихся задач разработки моделей машинного обучения с высокой масштабируемостью, эффективностью и производительностью, сохраняя при этом качество модели.

Платформа Auto-ML, - инструмент для создания моделей, который оптимизирует конвейеры машинного обучения.

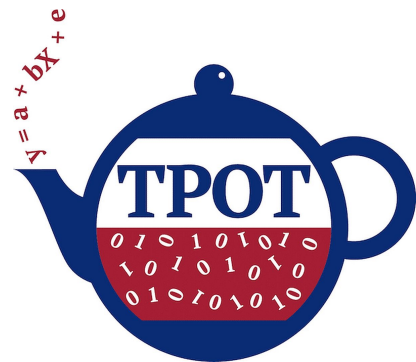
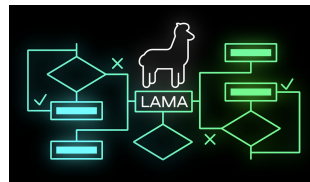


Когда следует использовать AutoML?

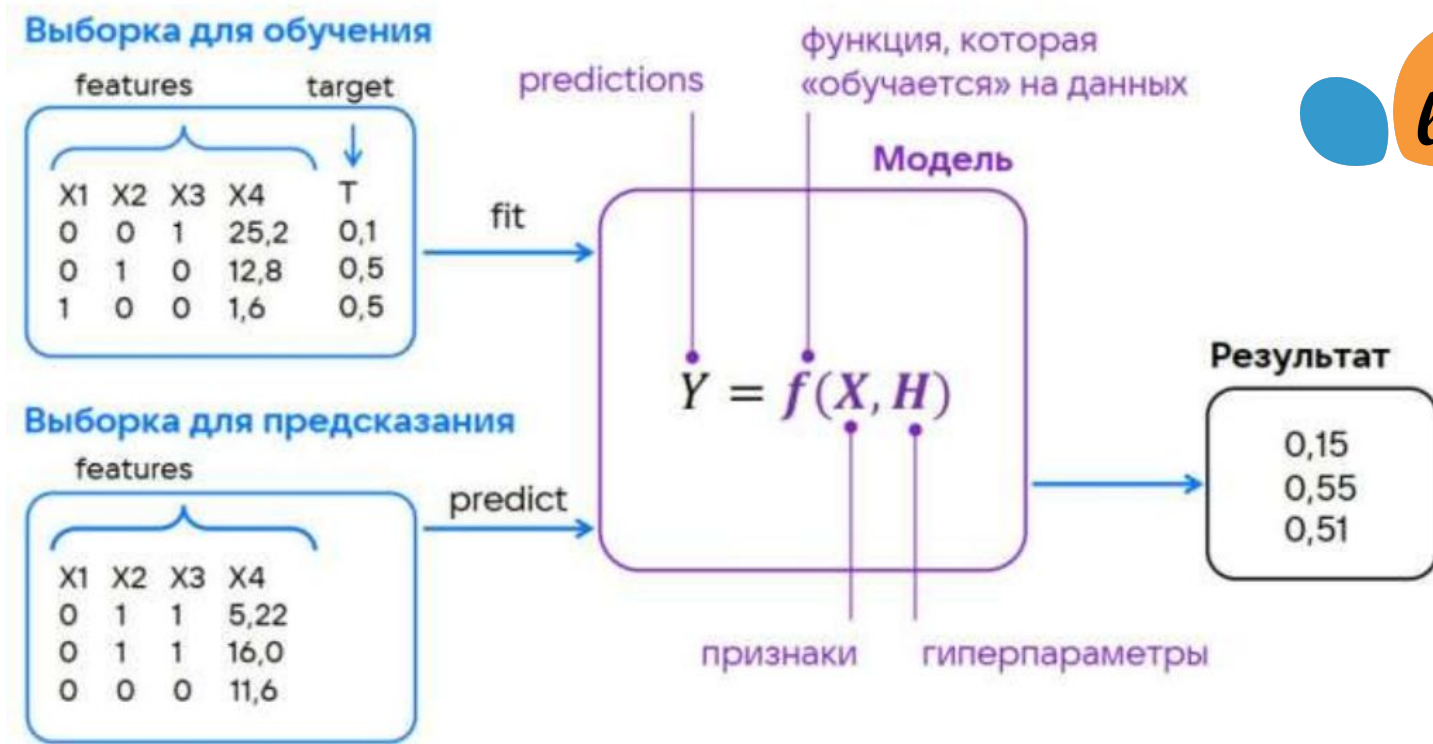
Традиционно, модели ML эффективно и удобно эксплуатировать когда структура данных относительно постоянна. У Вас имеются стабильные каналы получения данных, известен набор признаков. Вы понимаете алгоритм обработки и выстраиваете пайплайн.

Сложности начинаются в момент, когда признаки в данных отличаются от задачи к задаче или же признаков становится мало и качество модели начинает ухудшаться.

Задача AutoML состоит в том, чтобы найти наиболее эффективную комбинацию методов, чтобы вы могли свести к минимуму ошибки в своих прогнозах.

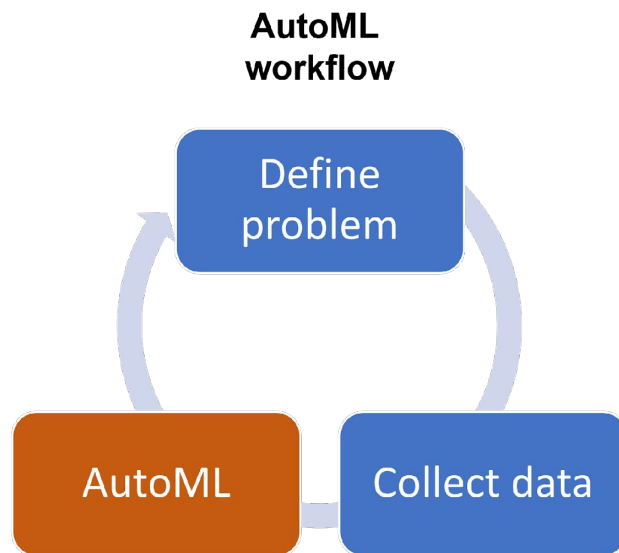
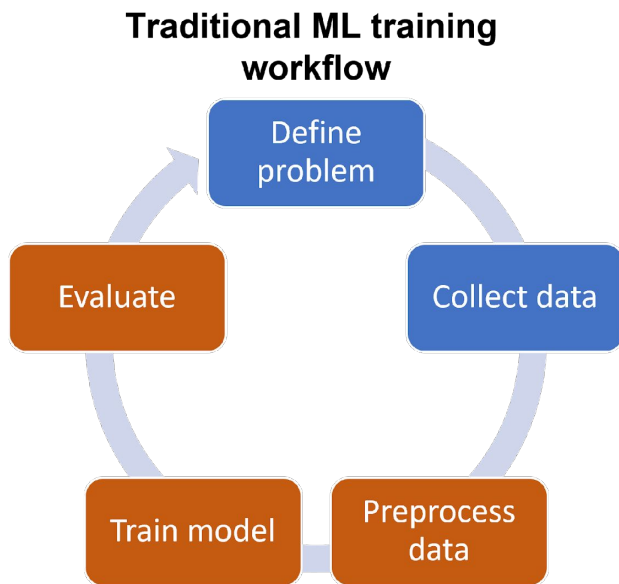


Классический пайплайн обучения модели



Что автоматизирует AutoML?

Инструменты AutoML упрощают процесс обработки данных, подбора моделей и их гиперпараметров, используя имеющуюся информацию, и включают следующие этапы:



Основные этапы AutoML

Загрузка и подготовка данных. Без пайплайна данные должны проходить через отдельные преобразователи: энкодеры (преобразуют категориальные признаки в числовые векторы), импьютеры (заполняют пропуски в данных) и скейлеры (приводят признаки к одному масштабу). Каждый инструмент по отдельности нужно натренировать на обучающей выборке, преобразовать её и отдельно сделать преобразование тестовой выборки. В результате получается много повторяющегося кода.

Пайплайн собирает все инструменты в один конвейер без повторяющегося кода. Достаточно обучить этот конвейер на выборке данных и использовать его для всех нужных преобразований одной командой. Он принимает на вход признаки, преобразует их и выдаёт результат.



Основные этапы AutoML

Генерация моделей. Модели с различными конфигурациями создаются и обучаются с использованием стекинга. При этом большинство систем AutoML не стараются использовать только глубокие нейронные сети, которые могут быть излишними для многих задач, тогда как простая модель, например, логистическая регрессия или деревья решений, может оказаться более подходящей и выигрывает в производительности от оптимизации гиперпараметров.

На этом этапе системы AutoML имеют преимущество, поскольку они способны создавать огромное количество тестовых моделей за очень короткий промежуток времени.



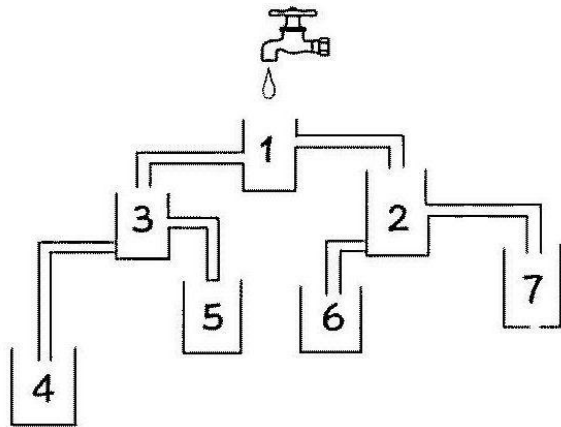
Основные этапы AutoML

Тестирование производительности моделей.

Именно на этом этапе требуется ручная настройка, потому как крайне важно, чтобы пользователь выбрал правильную модель для задачи, оценивая достаточный уровень качества, а также оценил вклад каждого признака в результат.

Например, в библиотеке SHAP для оценки важности фичей рассчитываются значения Шэпли (амер. математик.). Для оценки важности фичи происходит оценка предсказаний модели с и без данной фичи.

КАКАЯ ЧАШКА НАПОЛНИТСЯ
БЫСТРЕЕ ВСЕХ?



Какие инструменты мы рассмотрим?

На вебинаре **разберем два ключевых инструмента**, которые помогут повысить эффективность выбора моделей и построения пайплайнов :

- Автоматизация выбора моделей с помощью платформы **H2O** - это полностью открытая распределенная платформа машинного обучения на базе Java. H2O AutoML можно использовать для автоматизации процесса машинного обучения, который включает автоматическое обучение и настройку многих моделей в течение заданного пользователем времени.
- Автоматизация создания пайплайнов с помощью библиотеки **TPOT** (Tree-based Pipeline Optimization Tool) Auto ML автоматизированный пакет машинного обучения на Python, который использует концепции генетического программирования для оптимизации конвейера машинного обучения.

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

**H2O – Просто добавь
воды?**

Что собой представляет платформа H2O?

H2O JVM предоставляет веб-сервер, так что вся связь происходит через сокет (указанный IP-адресом и портом).

H2O поддерживает наиболее широко используемые статистические алгоритмы и алгоритмы машинного обучения, включая XGBoost, обобщенные линейные модели, глубокое обучение и многое другое.

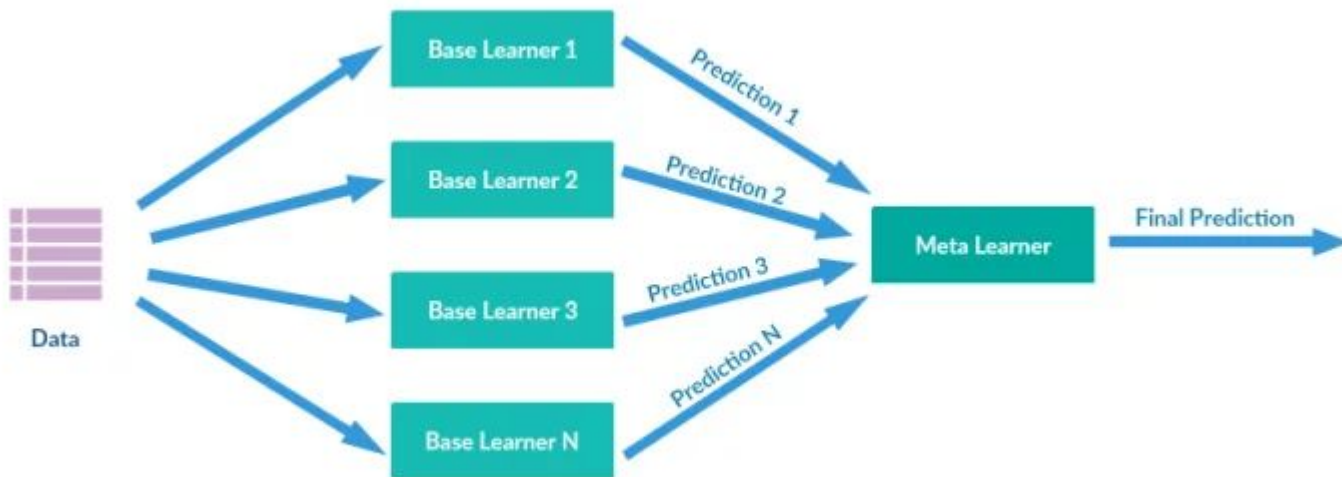
Каждый новый сеанс python начинается с инициализации соединения между клиентом python и вычислительным кластером H2O.

Установка посредством команды*: *pip install h2o*

* Не забудьте чтобы была установлена Java на вашей ОС: <https://www.java.com/ru/download/>

Стекинг в H2O

Стекинг (также называемый метаансамблирование) — это метод ансамблирования моделей, используемый для объединения информации из нескольких прогнозирующих моделей для создания новой модели (модель 2-го уровня)



Как быстро начать работать с H2O?

```
import h2o
from h2o.automl import H2OAutoML
h2o.init()
```

Checking whether there is an H2O instance running at <http://localhost:54321>. connected.
Warning: Your H2O cluster version is (7 months and 30 days) old. There may be a newer version available.
Please download and install the latest version from: https://h2o-release.s3.amazonaws.com/h2o/latest_stable.html

H2O_cluster_uptime:	18 mins 27 secs
H2O_cluster_timezone:	+03:00
H2O_data_parsing_timezone:	UTC
H2O_cluster_version:	3.40.0.2
H2O_cluster_version_age:	7 months and 30 days
H2O_cluster_name:	H2O_from_python_user_jsx1vf
H2O_cluster_total_nodes:	1
H2O_cluster_free_memory:	13.64 Gb
H2O_cluster_total_cores:	12
H2O_cluster_allowed_cores:	12
H2O_cluster_status:	locked, healthy
H2O_connection_url:	http://localhost:54321
H2O_connection_proxy:	{"http": null, "https": null}
H2O_internal_security:	False
Python_version:	3.9.13 final

1. *Импорт библиотеки*
2. *При необходимости импорт отдельных разделов и функций*
3. *инициализация кластера на текущей машине*



Что умеет платформа H2O?

Платформа H2O позволяет работать и через веб интерфейс:

The screenshot displays the H2O FLOW web interface. At the top, the navigation bar includes the H2O FLOW logo and a hamburger menu, followed by tabs for Flow, Cell, Data, Model, Score, Admin, and Help. Below this, the title 'Untitled Flow' is shown. A toolbar with various icons for file operations and execution is visible. The main workspace is divided into three sections: a top bar with 'assist' and a '50ms' timer, a left sidebar titled '? Assistance' containing a table of routines, and a right sidebar titled 'Help'.

Routine	Description
importFiles	Import file(s) into H2O
importSqlTable	Import SQL table into H2O
getFrames	Get a list of frames in H2O
splitFrame	Split a frame into two or more frames
mergeFrames	Merge two frames into one
getModels	Get a list of models in H2O
getGrids	Get a list of grid search results in H2O
getPredictions	Get a list of predictions in H2O
getJobs	Get a list of jobs running in H2O
runAutoML	Automatically train and tune many models
buildModel	Build a model
importModel	Import a saved model
predict	Make a prediction

The right sidebar 'Help' section includes a 'Quickstart Videos' button, a link to 'view example Flows', and a 'STAR H2O ON GITHUB!' button. Below this, a 'GENERAL' section lists several topics: Flow Web UI, Importing Data, Building Models, Making Predictions, Using Flows, and Troubleshooting Flow.

Что умеет платформа H2O?

1. Загружайте данные из различных источников, например, это может быть датафрейм pandas: <https://docs.h2o.ai/h2o/latest-stable/h2o-py/docs/data.html>

```
from sklearn.datasets import load_wine
import pandas as pd
```

```
wine = load_wine()
x = pd.DataFrame(wine.data, columns=wine.feature_names)
y = pd.DataFrame(wine.target, columns=['type'])
wine = pd.concat([x,y], axis=1)
```

```
df = h2o.H2OFrame(wine)
```

Parse progress:  (done) 100%

Что умеет платформа H2O?

2. Выполните препроцессинг и разделите выборку данных на тестовую и обучающую:

```
splits = df.split_frame(ratios=[0.7], seed=1)
train = splits[0]
test = splits[1]
```


3. Укажите фичи и целевую переменную, например, вот таким образом:

```
y = "type"
x = df.columns
x.remove(y)
```

Что умеет платформа H2O?

4. Выполните обучение модели с определенными параметрами (+ внимательно следите за предупреждениями)

```
aml = H2OAutoML(max_runtime_secs=60, seed=1)
aml.train(x=x, y=y, training_frame=train)
```

AutoML progress: 

```
11:14:24.713: AutoML: XGBoost is not available; skipping it.
```

```
11:14:24.756: _min_rows param, The dataset size is too small to split for min_rows=100.0: must have at least 200.0 (weighted) rows, but have only 132.0.
```

```
(done) 100%
```

Что умеет платформа H2O?

5. Изучите отчеты по результатам обучения модели:

Model Summary for Stacked Ensemble:

key	value
Stacking strategy	cross_validation
Number of base models (used / total)	3/5
# GBM base models (used / total)	1/1
# DeepLearning base models (used / total)	1/1
# DRF base models (used / total)	1/2
# GLM base models (used / total)	0/1
Metalearner algorithm	GLM
Metalearner fold assignment scheme	Random
Metalearner nfolds	5
Metalearner fold_column	None
Custom metalearner hyperparameters	None

```
ModelMetricsRegressionGLM: stackedensemble
** Reported on train data. **
```

MSE: 0.07118890966542255

RMSE: 0.26681249908020155

MAE: 0.23100210908955773

RMSLE: 0.18569903566127388

Mean Residual Deviance: 0.07118890966542255

R^2: 0.8790919619835927

Null degrees of freedom: 131

Residual degrees of freedom: 128

Null deviance: 77.7196969696968

Residual deviance: 9.396936075835775

Что умеет платформа H2O?

6. Выполните предсказание по обученной модели:

```
pred = aml.predict(test)
pred.head()
```

```
stackedensemble prediction progress: |████████████████████| (done) 100%
```

predict

0.11639

0.17507

-0.0498891

0.0760086

0.00799315

0.0490118

0.0578239

0.0220743

-0.0863229

0.00572693

```
[10 rows x 1 column]
```

Какую задачу сейчас
решили? напишите в
чат!

Что умеет платформа H2O?

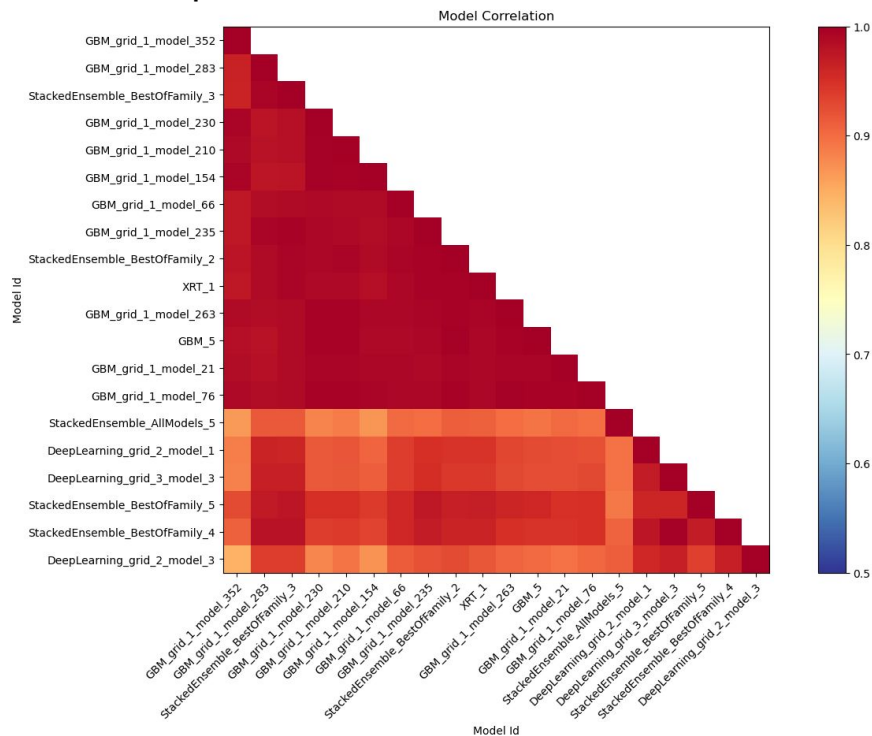
7. Выполните оценку качества предсказания на тестовой выборке:

```
aml.explain(test)
```

	model_id	rmse	mse	mae	rmsle	mean_residual_deviance	training_time_ms	predict_time_
StackedEnsemble_BestOfFamily_3_AutoML_4_20231110_153747		0.126141	0.0159115	0.071541	0.0679613	0.0159115	117	
GBM_grid_1_AutoML_4_20231110_153747_model_8		0.155502	0.0241809	0.10004	0.0883963	0.0241809	73	
DeepLearning_grid_1_AutoML_4_20231110_153747_model_1		0.161465	0.0260708	0.107212	0.074008	0.0260708	2436	
GBM_grid_1_AutoML_4_20231110_153747_model_40		0.165601	0.0274239	0.103135	0.0874217	0.0274239	40	
XRT_1_AutoML_4_20231110_153747		0.172555	0.0297752	0.101581	0.0826361	0.0297752	23	
GBM_grid_1_AutoML_4_20231110_153747_model_240		0.173224	0.0300066	0.120385	0.0987553	0.0300066	57	
GBM_grid_1_AutoML_4_20231110_153747_model_235		0.174431	0.0304263	0.103195	0.0913064	0.0304263	62	
GBM_grid_1_AutoML_4_20231110_153747_model_128		0.174654	0.030504	0.115177	0.100833	0.030504	50	
GBM_grid_1_AutoML_4_20231110_153747_model_123		0.178276	0.0317823	0.11156	0.0996476	0.0317823	43	
GBM_grid_1_AutoML_4_20231110_153747_model_99		0.180787	0.0326839	0.123728	0.103709	0.0326839	29	
GBM_grid_1_AutoML_4_20231110_153747_model_13		0.183056	0.0335095	0.108999	0.0859264	0.0335095	64	

Что умеет платформа H2O?

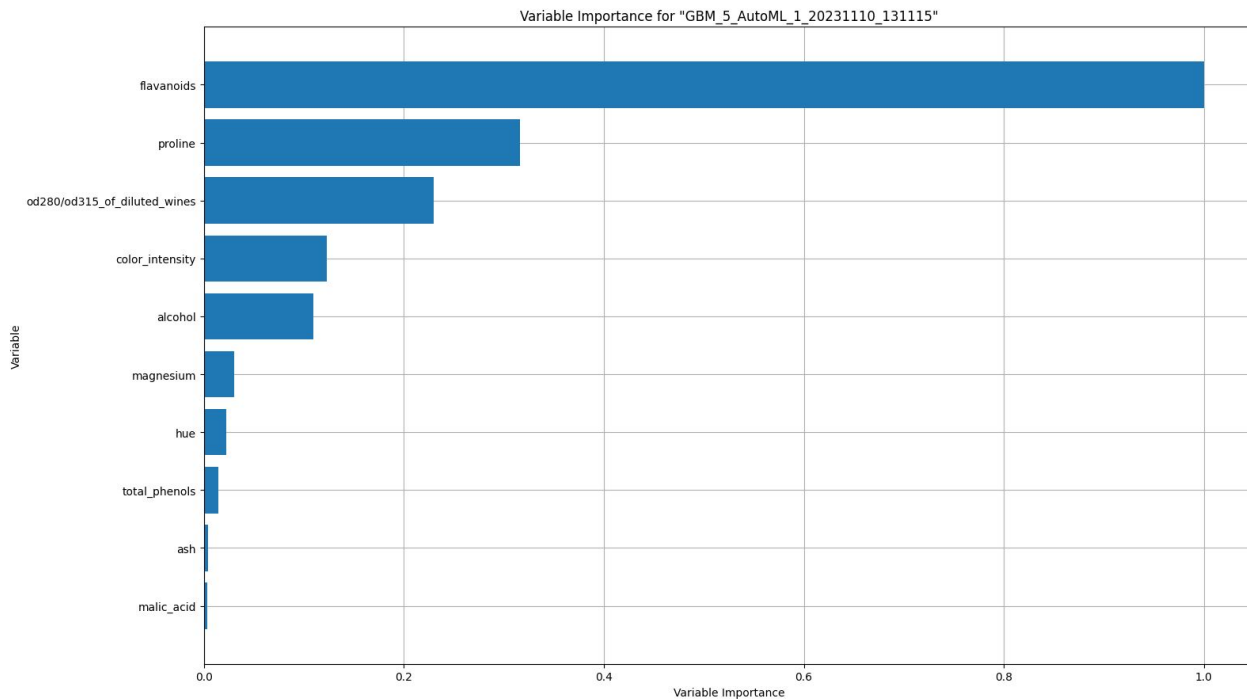
8. Оцените предсказательные способности моделей:



Этот график показывает корреляцию между предсказаниями моделей. Для классификации используется частота одинаковых предсказаний. По умолчанию модели упорядочены по сходству (вычисляемому с помощью иерархической кластеризации).

Что умеет платформа H2O?

9. Выполните оценку важности признаков (переменных): показывает относительную важность переменных по всему стеку моделей.



Что умеет платформа H2O?

10. Постройте график объяснимости переменных:

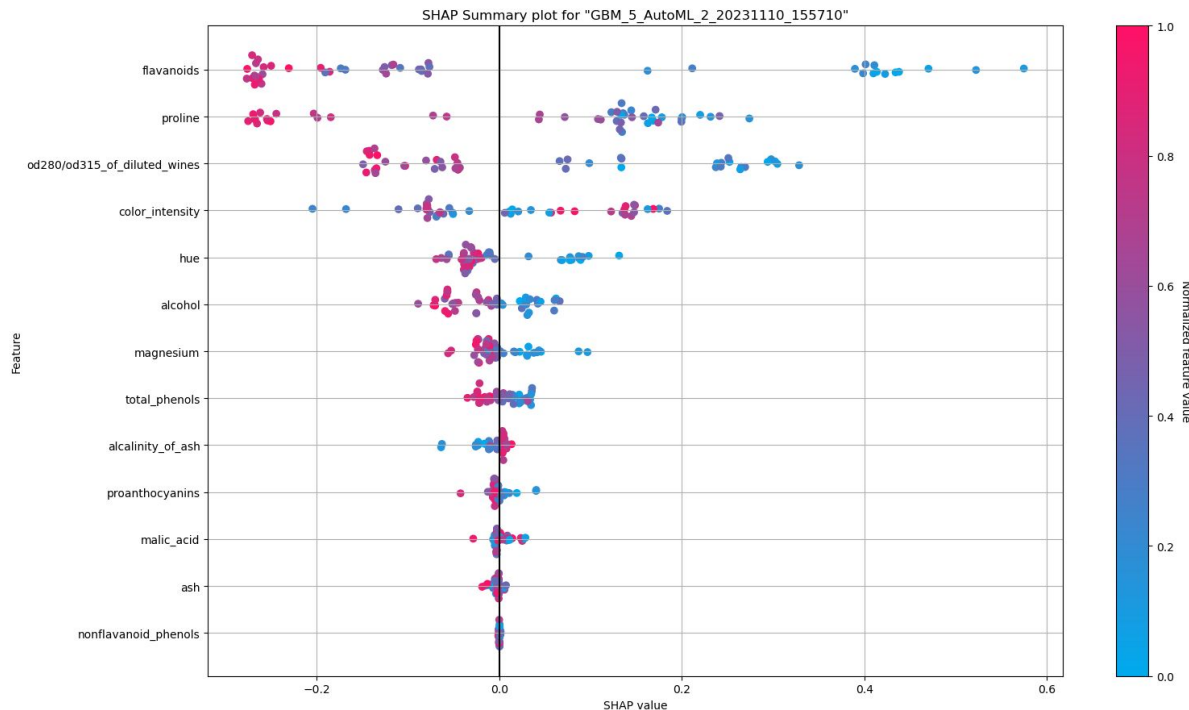
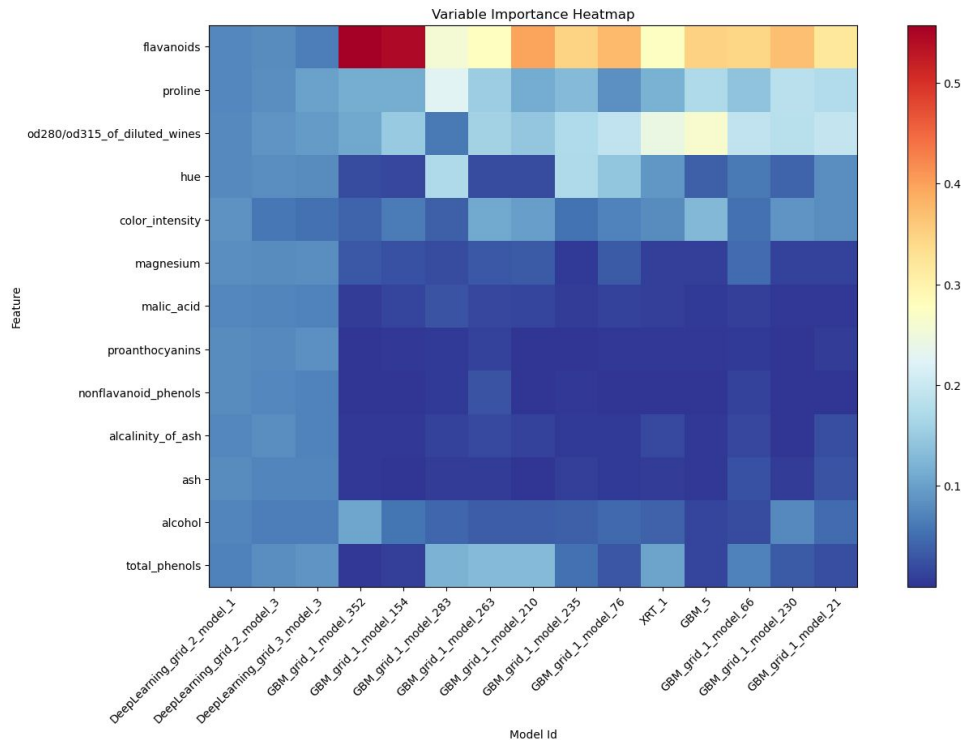


график SHAP показывает вклад признаков для каждого экземпляра (точки данных). Сумма вкладов признаков и смещения (bias) равна прогнозу модели.

Что умеет платформа H2O?

11. Оцените какие переменные оказываются более важными для конкретной модели



Тепловая карта важности переменных показывает важность переменных для нескольких моделей.

Вопросы?



Ставим “+”,
если вопросы есть

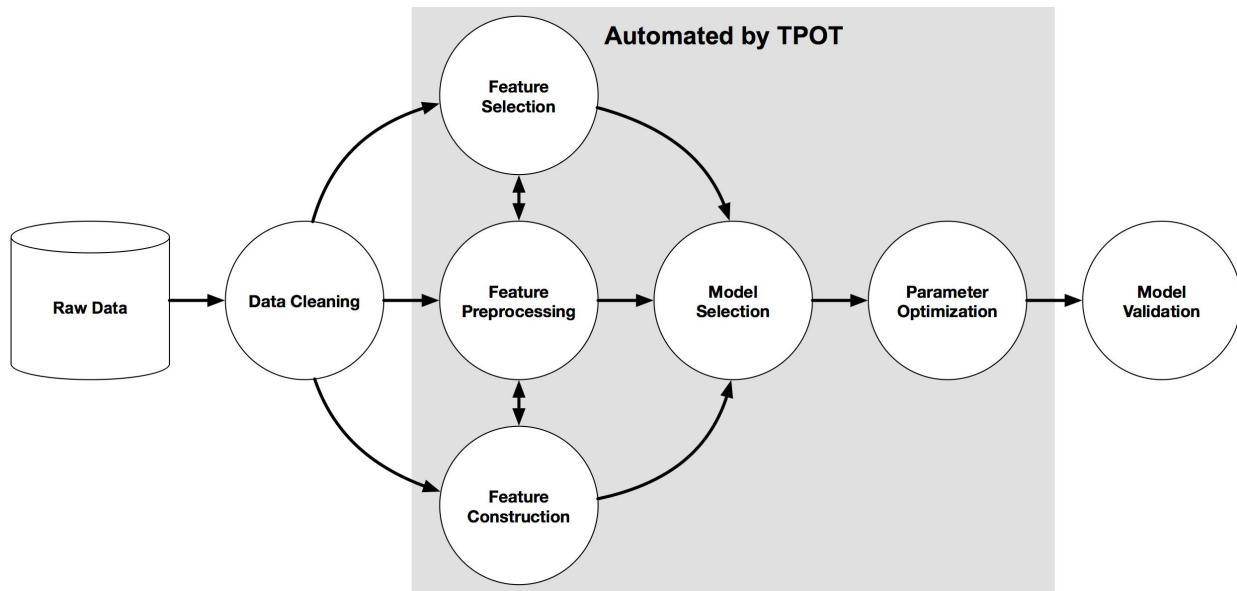


Ставим “-”,
если вопросов нет

**ТР0Т – можно ли
обойтись без
«Five o'clock Tea»?**

Что умеет TPOT?

TPOT позволяет исследовать множество вариантов конвейеров (пайплайнов) и находить лучший конвейер для представленного набора данных, что позволит избежать наиболее утомительной части процесса машинного обучения.



Что «под капотом»?

Библиотека Python, `tpot` созданная поверх `scikit-learn`, использует генетическое программирование для оптимизации вашего конвейера машинного обучения. Например, в машинном обучении после подготовки ваших данных вам нужно знать, какие функции вводить в вашу модель и как вы должны создавать эти функции. Как только у вас появятся эти функции, вы вводите их в свою модель для обучения, а затем настраиваете ее гиперпараметры для получения оптимальных результатов.

Вместо того, чтобы делать все это самостоятельно методом проб и ошибок, `TPOT` автоматизирует эти шаги для вас с помощью генетического программирования и выводит оптимальный код.



Как люди видят мою работу

$$L_p = ||w||^2 - \sum_{i=1}^n a_i y_i (x_i \cdot w + b) + \sum_{i=1}^n c_i$$
$$a_i \geq 0, \forall i$$
$$w = \sum_{i=1}^n a_i y_i x_i, \sum_{i=1}^n a_i y_i = 0$$
$$\nabla g(\theta_t) = \frac{1}{n} \sum_{i=1}^n \nabla \ell(x_i, y_i; \theta_t) + \nabla r(\theta_t)$$
$$\theta_{t+1} = \theta_t - \eta \nabla \ell(x_{(t)}, y_{(t)}; \theta_t) - \eta \cdot \nabla r(\theta_t)$$
$$\mathbb{E}_{(t)}[\ell(x_{(t)}, y_{(t)}; \theta_t)] = \frac{1}{n} \sum_{i=1}^n \ell(x_i, y_i; \theta_t)$$

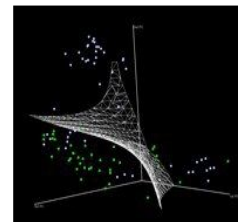
Как программисты видят мою работу



Как мои друзья видят мою работу



Как мои родители видят мою работу



Как я вижу свою работу

```
>>> from sklearn import svm
```

Что я на самом деле делаю

Генетическое программирование

Имея нужные данные, вычислительную мощность и модель машинного обучения, решая прикладные задачи, важно понимать какой алгоритм необходимо использовать. Генетические алгоритмы основаны на дарвиновском процессе естественного отбора и используются для генерации решений в задачах оптимизации.

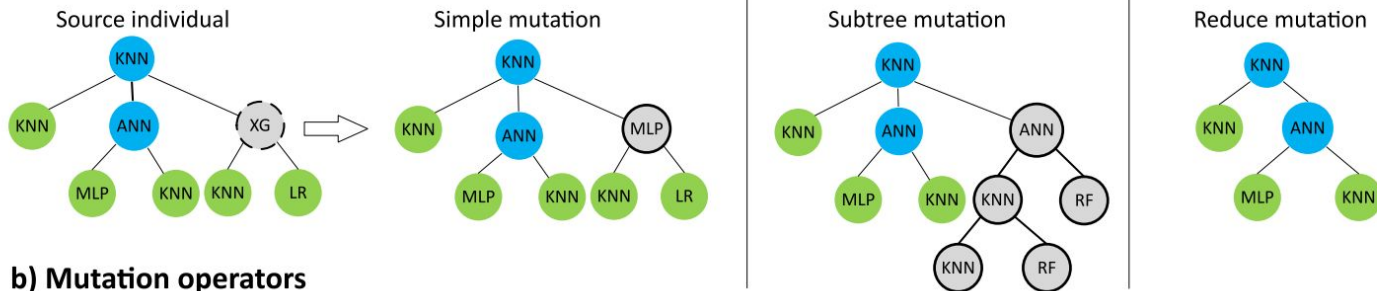
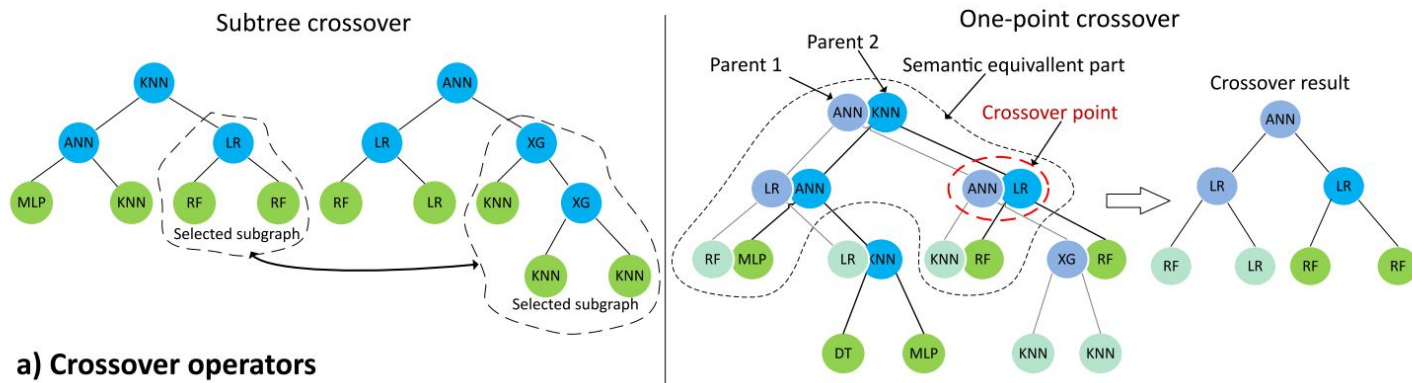
Генетические алгоритмы обладают тремя свойствами:

- 1. Выбор:** у вас есть совокупность возможных решений данной проблемы и функция пригодности. На каждой итерации вы оцениваете, как совместить каждое решение с вашей функцией пригодности.
- 2. Скрещивание:** Затем вы выбираете наиболее подходящие из них и выполняете скрещивание для создания новой совокупности.
- 3. Мутация:** вы берете эти дочерние элементы и изменяете их с помощью некоторой случайной модификации и повторяете процесс до тех пор, пока не получите наиболее подходящее или наилучшее решение.



Генетическое программирование

Автоматизированная подготовка пайплайна — это преимущественно задача комбинаторной оптимизации или поиска наилучшего сочетания возможных факторов — множества вычислительных блоков.



Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

Пример автоматизированного отбора признаков и выбора моделей в H2O

Пример задачи классификации.

Для примера рассмотрим альтернативный вариант запуска H2O с интеграцией в sklearn:

```
# Вариант запуска № 2 с использованием sklearn:
from sklearn.datasets import load_wine
import pandas as pd
from h2o.sklearn import H2OAutoMLClassifier
from sklearn.model_selection import train_test_split
wine = load_wine()
X = wine.data
y=wine.target

aml_c = H2OAutoMLClassifier(max_models=20, seed=1, max_runtime_secs=60)
X_train, X_test, y_train, y_test = train_test_split(X, y)

aml_c.fit(X=X, y=y)
```

```
Parse progress: |██████████████████████████████████████████████████████████████████████████| (done) 100%  
Parse progress: |██████████████████████████████████████████████████████████████████████████| (done) 100%  
AutoML progress: |██████████████████████████████████████████████████████████████████████████|  
14:44:59.583: AutoML: XGBoost is not available; skipping it.
```

```
14:45:00.379: _min_rows param, The dataset size is too small to split for min_rows=100.0: must have at least 200.0 (weighted) r
ows, but have only 178.0.
```

```
H2OAutoMLClassifier(max models=20, max runtime secs=60, seed=1)
```

Пример задачи классификации.

Результаты предсказания:

```
preds = aml.c.predict(X_test)
```

```
Parse progress: |██████████| (done) 100%  
deeplearning prediction progress: |██████████| (done) 100%
```

preds

```
array([1, 2, 2, 1, 0, 1, 0, 1, 0, 2, 0, 0, 1, 1, 1, 1, 2, 0, 0, 1, 2, 1,
       2, 0, 1, 1, 0, 2, 0, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 0, 1, 1, 0,
       2], dtype=int64)
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test, preds))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	12
1	1.00	1.00	1.00	22
2	1.00	1.00	1.00	11
accuracy			1.00	45
macro avg	1.00	1.00	1.00	45
weighted avg	1.00	1.00	1.00	45

Пример задачи классификации.

таблица лидеров:

```
aml_c.estimator.leaderboard
```

	model_id	mean_per_class_error	logloss	rmse	mse
	DeepLearning_grid_2_AutoML_1_20231110_144459_model_2	0.0103446	0.0491584	0.108209	0.0117091
	StackedEnsemble_BestOfFamily_1_AutoML_1_20231110_144459	0.0103446	0.0436445	0.105264	0.0110805
	DeepLearning_grid_3_AutoML_1_20231110_144459_model_2	0.0140845	0.0591254	0.115214	0.0132743
	GBM_5_AutoML_1_20231110_144459	0.0140845	0.0584889	0.130118	0.0169308
	GBM_4_AutoML_1_20231110_144459	0.0140845	0.0506053	0.125983	0.0158717
	GBM_grid_1_AutoML_1_20231110_144459_model_4	0.0140845	0.0513526	0.127037	0.0161385
	GBM_3_AutoML_1_20231110_144459	0.0140845	0.0529837	0.130352	0.0169917
	DRF_1_AutoML_1_20231110_144459	0.0140845	0.123183	0.165229	0.0273006
	GBM_grid_1_AutoML_1_20231110_144459_model_2	0.0140845	0.0584899	0.132432	0.0175381
	GLM_1_AutoML_1_20231110_144459	0.0163341	0.0544163	0.11673	0.0136259

[22 rows x 5 columns]

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

Пример автоматизации построения пайплайнов в TPOТ

Практический пример на Python.

Установка библиотеки TPOT командой `pip install tpot`

Задача: построение пайплайна для классификации объектов.

Шаг 1. Загрузка данных

```
# импорт библиотек
from tpot import TPOTClassifier
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
```

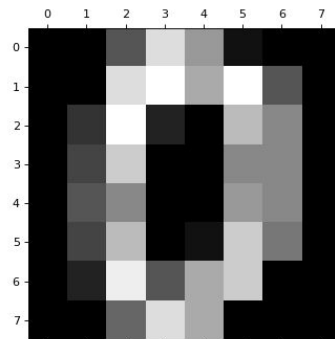
```
digits = load_digits() # загрузка набора изображений рукописных цифр 8x8 пикселей
X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target, train_size=0.75)
```

```
len(digits.data [0])
```

64

```
digits.target
```

```
array([0, 1, 2, ..., 8, 9, 8])
```



Практический пример на Python.

Шаг 2. Запуск алгоритма поиска наилучшего пайплайна:

```
tpot = TPOTClassifier(generations=2, verbosity=2, max_time_mins=0.5, random_state=42)
tpot.fit(X_train, y_train)
```

Optimization Progress: 16%



16/100 [00:20<05:12, 3.72s/pipeline]

0.76 minutes have elapsed. TPOT will close down.

TPOT closed during evaluation in one generation.

WARNING: TPOT may not provide a good pipeline if TPOT is stopped/interrupted in a early generation.

TPOT closed prematurely. Will use the current best pipeline.

Best pipeline: LogisticRegression(input_matrix, C=0.5, dual=False, penalty=l2)

```
TPOTClassifier
TPOTClassifier(generations=2, max_time_mins=0.5, verbosity=2)
```

Практический пример на Python.

Типовые параметры:

```
class tpot.TPOTClassifier(generations=100, population_size=100,  
                          offspring_size=None, mutation_rate=0.9,  
                          crossover_rate=0.1,  
                          scoring='accuracy', cv=5,  
                          subsample=1.0, n_jobs=1,  
                          max_time_mins=None, max_eval_time_mins=5,  
                          random_state=None, config_dict=None,  
                          template=None,  
                          warm_start=False,  
                          memory=None,  
                          use_dask=False,  
                          periodic_checkpoint_folder=None,  
                          early_stop=None,  
                          verbosity=0,  
                          disable_update_check=False,  
                          log_file=None  
                          )
```

Количество итераций для запуска процесса оптимизации конвейера.

Количество особей, которые необходимо сохранять в популяции генетического программирования каждое поколение

Количество потомков, которое необходимо произвести в каждом поколении генетического программирования.

Частота мутаций для алгоритма генетического программирования

и др.

Практический пример на Python.

Шаг 3. Тестирование модели с заданной метрикой и сохранение лучшего варианта пайплайна:

```
print( tpot.score(X_test, y_test) )
```

0.9688888888888889

```
tpot.export('tpot_images.py')
```

tpot_images.py X

```
1 import numpy as np
2 import pandas as pd
3 from sklearn.linear_model import LogisticRegression
4 from sklearn.model_selection import train_test_split
5
6 # NOTE: Make sure that the outcome column is labeled 'target' in the data file
7 tpot_data = pd.read_csv('PATH/TO/DATA/FILE', sep='COLUMN_SEPARATOR', dtype=np.float64)
8 features = tpot_data.drop('target', axis=1)
9 training_features, testing_features, training_target, testing_target = \
10 | | | | train_test_split(features, tpot_data['target'], random_state=None)
11
12 # Average CV score on the training set was: 0.9651053283767037
13 exported_pipeline = LogisticRegression(C=0.5, dual=False, penalty="l2")
14
15 exported_pipeline.fit(training_features, training_target)
16 results = exported_pipeline.predict(testing_features)
```



Выводы

1. Платформа H2O позволяет реализовать отбор признаков и конструирование моделей машинного обучения, однако эти разработки сосредоточены главным образом на повышении скорости генерации готовых моделей, а не на изучении того, как можно улучшить технологию для решения более сложных проблем.
2. Выбор правильной модели машинного обучения и наилучших гиперпараметров для этой модели по своей сути является задачей оптимизации, для решения которой можно использовать генетическое программирование. После использования TPOT весь код для предобработки, обучения и предсказания занимает буквально полстраницы.
3. Важным ограничением использования Auto ML является требовательность к вычислительным ресурсам, относительно длительное время работы на объемных датасетах, а также отсутствие связи между предметной областью и отбором признаков в AutoML общего назначения

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

Ключевые тезисы занятия

Подведем итоги

1. Рассмотрели какие существуют инструменты автоматизации выбора моделей и построения пайплайнов
2. Сформировали представление о том какие основные принципы лежат в основе реализации процесса AutoML
3. Выяснили, что AutoML позволяет повысить эффективность решения прикладных задач и облегчить процессы построения моделей
4. Научились развертывать вычислительный кластер H2O и обучать с его помощью модели
5. Научились формированию пайплайнов в TPOT.

Дополнительные материалы

Список материалов для изучения

1. Документация по H2O <https://docs.h2o.ai/h2o/latest-stable/h2o-py/docs/intro.html#>
2. Документация по TPOT <https://epistasislab.github.io/>
3. Полезный тьюториал по H2O на Kaggle:
<https://www.kaggle.com/code/paradiselost/tutorial-automl-capabilities-of-h2o-library>
4. Тьюториал по интеграции H2O и Sklearn :
https://github.com/h2oai/h2o-tutorials/blob/master/tutorials/sklearn-integration/H2OAutoML_as_sklearn_estimator.ipynb
5. Пример кода из вебинара:
 - а) Задача регрессии в H2O :
https://drive.google.com/file/d/1U0GnvO_iBt4Lmu5-xNJMWegpmVJpdVow/view?usp=sharing
 - б) Задача классификации H2O+sklearn:
https://drive.google.com/file/d/1bbDnUljU_CBG2yDtBCxO25y55S0W-2Jk/view?usp=sharing
 - в) Работа с TPOT:
<https://colab.research.google.com/drive/177l7sbcEVI8xd6Y3nVK58dTt1pJJnj76?usp=sharing>
 - г) бонус: Пример по прогнозированию временного ряда в H2O
<https://colab.research.google.com/drive/1l0aDcGlqUwl7xNqvn3QjNUuHMihKXMGj?usp=sharing>

**Заполните, пожалуйста,
опрос о занятии
по ссылке в чате**

Спасибо за внимание!

Приходите на следующие вебинары



Алексей Кисляков

Заведующий научной лабораторией «Кластерный анализ процессов роста и эволюции систем», доктор экон. наук., кандидат техн. наук, доцент.

О себе: ученый-исследователь

- > 10 лет научных исследований в области прикладных методов экономико-математического моделирования с применением инструментария Python, R, MATLAB, Excel
- > 8 лет преподавания в ВУЗах оффлайн и онлайн
- > 30 созданных и реализованных курсов по программам высшего образования и дополнительного профессионального образования

Телефон / эл. почта / соц. сети: +7(904) 261-57-18, ankislyakov@mail.ru