TPOT (Tree-based Pipeline Optimization Tool) – инструмент для автоматического машинного обучения с использованием генетических алгоритмов. TPOT автоматически строит и оптимизирует конвейеры обработки данных. Он позволяет автоматически настраивать как отдельные модели, так и последовательности преобразований данных. Разработан на Python и базируется на библиотеке машинного обучения Scikit-learn. Исходный код: <https://github.com/EpistasisLab/tpot>

**Общий алгоритм работы**.

Создание первого поколения

В начале работы TPOT создает задаваемое пользователем (population\_size) количество особей, представляющих собой случайным образом инициализированные конвейеры машинного обучения (первое поколение). Обязательная часть конвейера - конечный регрессор или классификатор в зависимости от того, решается ли задача регрессии (TPOTRegressor) или классификации (TPOTClassifier). В конвейер случайным образом могут включаться операции предварительной обработки данных (масштабирование, отбор признаков и т.п.) с различными параметрами.

Обучение и сохранение лучшего результата

После создания поколения конвейеры запускаются и по результатам обучения оценивается их точность (метрики точности могут быть заданы пользователем).

Самый лучший (точный) конвейер в поколении сравнивается с самым точным из предыдущих поколений и, если он лучше, то сохраняется в специальной переменной для последующего сравнения с результатами будущих поколений;

Создание нового поколения. Мутации и скрещивания.

Часть особей нового поколения (по умолчанию mutation\_rate = 90%) появляются в результате мутации: случайным образом в родительской особи меняются как сами операторы конвейеров, так и их параметры. Это означает, что регрессор в конвейере может быть заменен другим регрессором, или параметры регрессора могут быть изменены. Также могут быть добавлены или удалены дополнительные операторы предварительной обработки.

Часть особей нового поколения (по умолчанию crossover\_rate = 10%) появляются в результате скрещивания: два случайно выбранных родительских конвейера комбинируются для создания потомства. В новом конвейере часть операторов одного родителя могут быть заменены операторами из другого, например, операторы предварительной обработки, выбора признаков.

Если пользователь задал mutation\_rate + crossover\_rate < 100%, то оставшаяся до 100% часть нового поколения появляется в результате репродукции – часть родительских особей переносится в новую популяцию.

Окончание процесса оптимизации

Процесс создания новых поколений и оценки их качества продолжается до тех пор, пока не будет достигнуто заданное пользователем число поколений (generations, по умолчанию=100) или другие условия остановки (например, временной лимит (**max\_time\_mins**: по умолчанию=None)) или отсутствие улучшения в течение определенного количества поколений (early\_stop: по умолчанию=None))). После завершения оптимизации лучший найденный конвейер можно экспортировать в формате Python-кода для дальнейшего использования.

**Пример использования**

from tpot import TPOTRegressor

from sklearn.datasets import load\_boston

from sklearn.model\_selection import train\_test\_split

digits = load\_boston()

X\_train, X\_test, y\_train, y\_test = train\_test\_split(digits.data, digits.target, train\_size=0.75, test\_size=0.25)

tpot = TPOTRegressor (generations=5, population\_size=50)

tpot.fit(X\_train, y\_train)

print(tpot.score(X\_test, y\_test))

tpot.export('tpot\_boston\_pipeline.py')

**Используемые инструменты**

Поскольку TPOT основан на scikit-learn, он использует инструменты этой библиотеки.

Регрессоры, используемые по умолчанию:

ElasticNetCV

ExtraTreesRegressor

GradientBoostingRegressor

AdaBoostRegressor

DecisionTreeRegressor

KNeighborsRegressor

LassoLarsCV

LinearSVR

RandomForestRegressor

RidgeCV

SGDRegressor

XGBRegressor (если установлен XGBoost)

Перечень используемых по умолчанию инструментов (регрессоры, преобразователи и скейлеры) для задач регрессии с параметрами можно посмотреть здесь: <https://github.com/EpistasisLab/tpot/blob/master/tpot/config/regressor.py>

Стоит отметить наличие в перечне xgboost.XGBRegressor, который не входит в Scikit-learn, но его поддержка включена в TPOT.

**Пользовательские настройки**

TPOT поддерживает пользовательскую настройку путем создания конфигурационного словаря, в котором определяется перечень операций предварительной обработки данных, моделей и их параметров, которые должны использоваться при оптимизации конвейера. При задании такого словаря TPOT будет использовать только указанные в нем инструменты.

Пример словаря:

tpot\_config = {

# Модели

'sklearn.ensemble.RandomForestClassifier': {

'n\_estimators': [10, 50, 100, 200],},

'sklearn.neighbors.KNeighborsClassifier': {},

# Преобразования

'sklearn.preprocessing.StandardScaler': {},

}

tpot = TPOTRegressor(config\_dict=tpot\_config, ...)

В конфигурационный словарь можно включить любой регрессор, если он имеет API, совместимый со Scikit-learn (т.е. методы fit(), predict() и т. д.).

**Метрики точности**

При инициализации экземпляра TPOT (TPOTClassifier или TPOTRegressor), можно задать параметр scoring, чтобы указать, какую метрику следует использовать для оценки конвейеров. Если этот параметр не задан, TPOT использует метрику по умолчанию (точность для классификации и MSE для регрессии). В TPOT доступны все метрики точности, которые используются в Scikit-learn.

tpot = TPOTRegressor(scoring=' neg\_median\_absolute\_error', ...)

TPOT ищет конвейер, который максимизирует выбранную метрику. Часть метрик в Scikit-learn, такие как MSE, MAE, Log Loss, MSLE, определяют "ошибку", и их необходимо минимизировать, а не максимизировать. Чтобы привести задачу к форме максимизации (таким образом, чтобы TPOT мог работать с этой метрикой как с остальными), используют "отрицательную" версию этих метрик (например, "neg\_mean\_squared\_error"). Таким образом, минимизация отрицательной метрики будет эквивалентна максимизации исходной метрики.