Инструкция пользователя

FEDOT — open-source фреймворк автоматического машинного обучения (AutoML), который позволяет автоматически создавать и оптимизировать пайплайны - цепочки задач машинного обучения или отдельные их элементы (рисунок 1).

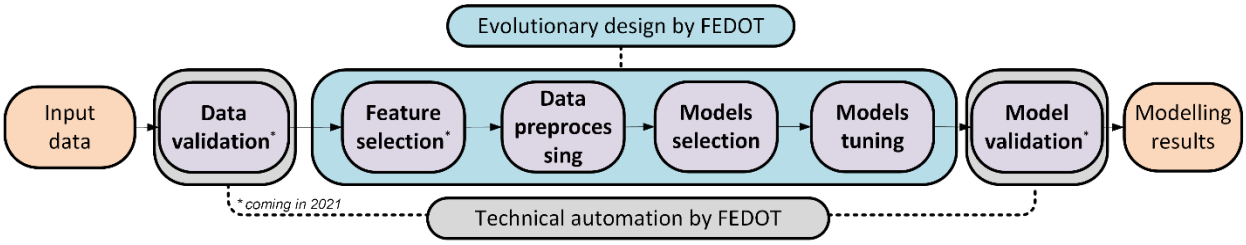

Рисунок 1 - Структура пайплайна машинного обучения и роль фреймворка FEDOT в его автоматизации и оптимизации

Основной акцент в работе фреймворка сделан на автоматизацию относительно сложного процесса управления взаимодействиями между различными вычислительными блоками пайплайнов. Эту особенность легко отметить на этапе непосредственного создания модели машинного обучения, так как FEDOT может не просто подобрать лучший тип модели из уже имеющихся, а позволяет создать сложную (композитную) структуру модели. Композитное моделирование позволяет совместно использовать несколько моделей различной сложности в виде единой неделимой структуры. Такой подход позволяет добиться лучшего качества моделирования, чем при использовании любого блока сложной модели по-отдельности. В рамках фреймворка композитные модели описываются в виде направленного графа, определяющего связи между блоками структуры – операциями предобработки данных и моделями.

На каких принципах основан фреймворк?

Применение фреймворка не ограничивается отдельными AutoML-задачами, такими как предобработка исходных данных, подбор признаков или оптимизация гиперпараметров моделей. Его использование позволяет решать более общую задачу структурного обучения. Для заданного набора данных строится решение в виде графа (DAG), узлы которого могут содержать модели МО, процедуры предобработки и трансформации данных. Структура графа, а также параметры каждого из узлов настраиваются методами оптимизации с точки зрения эффективности применения на предоставленных данных. Результирующий граф называется композитной моделью, как правило в его структуре содержатся сразу несколько моделей МО и блоков предобработки данных (пример на рисунке 2).

Структура такой модели может включать в себя большое количество элементов, которые последовательно соединены друг с другом и формируют иерархическую структуру. Примеры моделей, которые могут быть получены с помощью FEDOT, показаны ниже на рисунке 2: от простейшего варианта A) до сложного мультимодального варианта C).

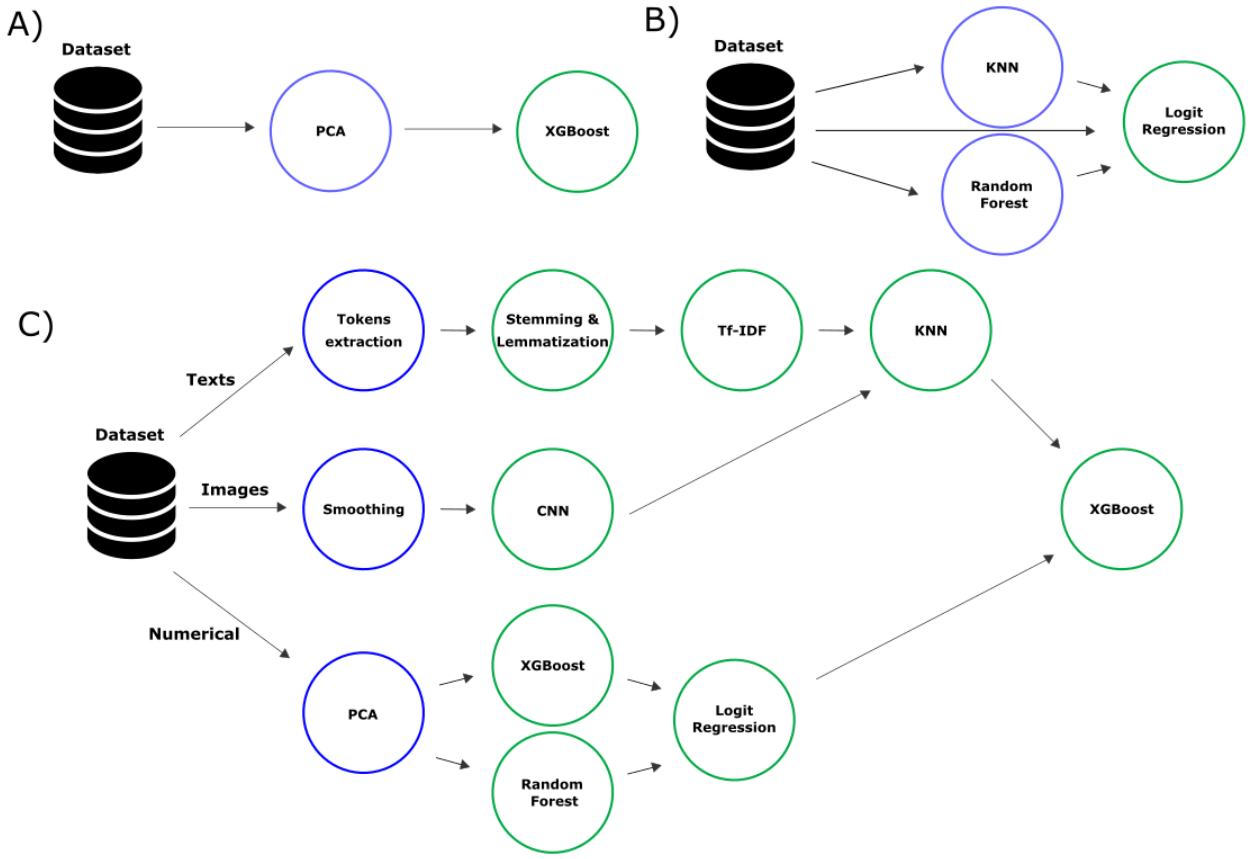

Рисунок 2 - Различные варианты пайплайнов, которые могут быть автоматически получены с помощью фреймворка FEDOT

Подбор наиболее подходящей структуры модели для конкретной задачи производится автоматически. Для этого используется эволюционный алгоритм оптимизации GPComp. Генерируется популяция из множества ML-пайплайнов и последовательно происходит поиск лучшего решения. При этом применяются методы генетического программирования - мутации и кроссовера. Также происходит избегание нежелательного переусложнения структуры модели за счет применения процедур регуляризации и многокритериальных подходов.

Архитектура фреймворка позволяет гибко расширять его возможности - добавлять новые модели, методы предобработки, алгоритмы настройки гиперпараметров, другие типы данных. Взаимодействие фреймворка с различными этапами пайплайна показано на рисунке 3.

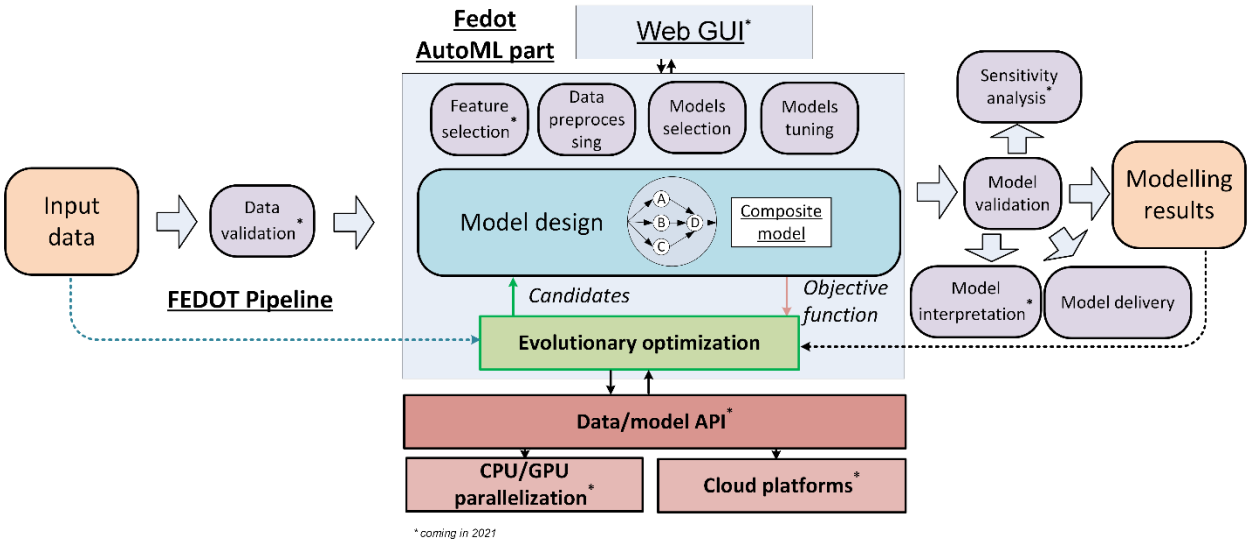

Рисунок 3 - Автоматизация различных этапов пайплайна машинного обучения с помощью фреймворка FEDOT

Пример работы с фреймворком

- **Шаг 1**. Определяем задачу и выбираем набор данных (датасет)

```
import pandas as pd

model = Fedot(problem='classification')

dataset_to_train = pd.read_csv(train_file_path)
dataset_to_validate = pd.read_csv(train_file_path)
```

- **Шаг 2**. Создаем экземпляр *Pipeline (последовательность, цепочку задач)*, создаем ноды (узлы) с необходимыми моделями

```
node_first = PrimaryNode('logit')
node_second= PrimaryNode('xgboost')
node_final = SecondaryNode('knn', nodes_from = [node_first, node_second])
pipeline = Pipeline(node_final)
```

- **Шаг 3**. Осуществляем обучение пайплайна (цепочки задач) с использованием метода *fit*.

```
model.fit(features=dataset_to_train, target='target', predefined_model=pipeline)
```

- **Шаг 4**. Делаем прогноз с использованием метода *predict*.

```
prediction = model.predict(features=dataset_to_validate)
```

Пример автоматизированного поиска модели и осуществления ее обучения и предсказания.

```
auto_model = Fedot(problem='classification')
pipeline = auto_model.fit(features=dataset_to_train, target='target')
prediction = auto_model.predict(features=dataset_to_validate)
```

```
auto_metrics = auto_model.get_metrics()
```

Импорт композитной модели

Для импортирования композитной модели вам необходимо создать пустой экземпляр пайплайна или использовать существующий, но тогда все данные будут перезаписаны в ходе импорта. Метод *'load_pipeline'* принимает на вход путь к файлу с расширением JSON

Пример:

```
from sklearn.metrics import mean_squared_error

test_data = InputData.from_csv(test_file_path)

pipeline = Pipeline()
pipeline.load_pipeline("data/Month:Day:Year, Time Period my_pipeline/my_pipeline.json")
predicted_values = pipeline.predict(test_data).predict
actual_values = test_data.target

mean_squared_error(predicted_values, actual_values)
```

Примечание: Обязательными полями для загрузки модели являются: **'model_id'**, **'model_type'**, **'preprocessor'**, **'params'**, **'nodes_from'**.

Дополнительная информация также доступна по адресу:

https://www.youtube.com/channel/UC4K9QWaEUpT_p3R4FeDp5jA