

Инструкция.

Список файлов.

- **anomaly_mapping.csv**, один из служебных csv-файлов, входящих в состав модели
- **api.py**, файл, содержащий код основной функции **separate_the_flees**
- **app.py**, файл, содержащий код эндпоинта **predict**
- **Dockerfile.txt**, докерфайл
- **eda_analysis.ipynb**, ноутбук, в котором строится модель и поясняется её выбор
- **hdbscan_model.zip**, сама модель, которую надо вручную достать из архива
- **model.py**, один из служебных файлов
- **part_10.csv**, исходные данные
- **popularity_stats_CLIENT_USERAGENT**, один из служебных csv-файлов, входящих в состав модели
- **popularity_stats_EVENT_ID**, один из служебных csv-файлов, входящих в состав модели
- **popularity_stats_MATCHED_VARIABLE_NAME**, один из служебных csv-файлов, входящих в состав модели
- **popularity_stats_MATCHED_VARIABLE_SRC**, один из служебных csv-файлов, входящих в состав модели
- **popularity_stats_MATCHED_VARIABLE_VALUE**, один из служебных csv-файлов, входящих в состав модели
- **processing.py**, один из служебных файлов
- **processing_for_rest.py**, один из служебных файлов
- **processing_in_prod.py**, один из служебных файлов
- **requirements.txt**, зависимости
- **test_api.py**, тест на api, оттуда можно взять пример вызова по REST
- **test_apply_the_model.py**, два теста на модель, один из них на всём датасете
- **test_parsing_http_requests.py**, один из тестов
- **test_prepare_the_data.py**, один из тестов
- **test_prepare_the_results.py**, один из тестов

- **анализ.txt**, файл который для каждого аномального кластера содержит статистику этого кластера и один из примеров (нужен для интерпретируемости результатов)

Тестирование.

Тесты запускаются через **pytest**:

```
Windows PowerShell
>> [internal] load build definition from Dockerfile.txt
>> transferring dockerfile: 299B
>> [internal] load metadata for docker.io/library/python:3.9
>> [internal] load build context
>> transferring context: 7.27kB
>> [1/6] FROM docker.io/library/python:3.9@sha256:6ea9d9fc96d7914c5c1d199f1f0195c4e05cf017b10666ca84cb7ce8e2699d51
>> CACHED [2/6] RUN mkdir /code
>> CACHED [3/6] WORKDIR /code
>> CACHED [4/6] COPY requirements.txt /code/requirements.txt
>> CACHED [5/6] RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt
>> [6/6] COPY . /code/.
>> exporting to image
>> exporting layers
>> writing image sha256:a1681ff8767605497331e602762bfac875b6e814cde67e268b834a3e2b0469d9
>> naming to docker.io/library/myimage
PS C:\Users\Семья\Documents\positive_technologies> docker run -d --name mycontainer -p 80:80 myimage
099f95049f71f8fa0535514e8fa0bca3bbfac71d03f2a6bdac758e5fb5eb385d
PS C:\Users\Семья\Documents\positive_technologies> pytest
===== test session starts =====
platform win32 -- Python 3.9.12, pytest-7.1.1, pluggy-1.0.0
rootdir: C:\Users\Семья\Documents\positive_technologies
plugins: anyio-3.5.0, dash-2.7.1, typeguard-2.13.3
collected 6 items

test_api.py .
test_apply_the_model.py .
test_parsing_http_requests.py .
test_prepare_the_data.py .
test_prepare_the_results.py .

===== warnings summary =====
test_apply_the_model.py::test_apply_the_model
test_apply_the_model.py::test_apply_the_model_to_all
C:\Users\Семья\Documents\positive_technologies\processing_in_prod.py:82: FutureWarning: Passing 'suffixes' which cause duplicate columns {'index_x', 'result' is deprecated and will raise a MergeError in a future version.
    tab = pd.merge(

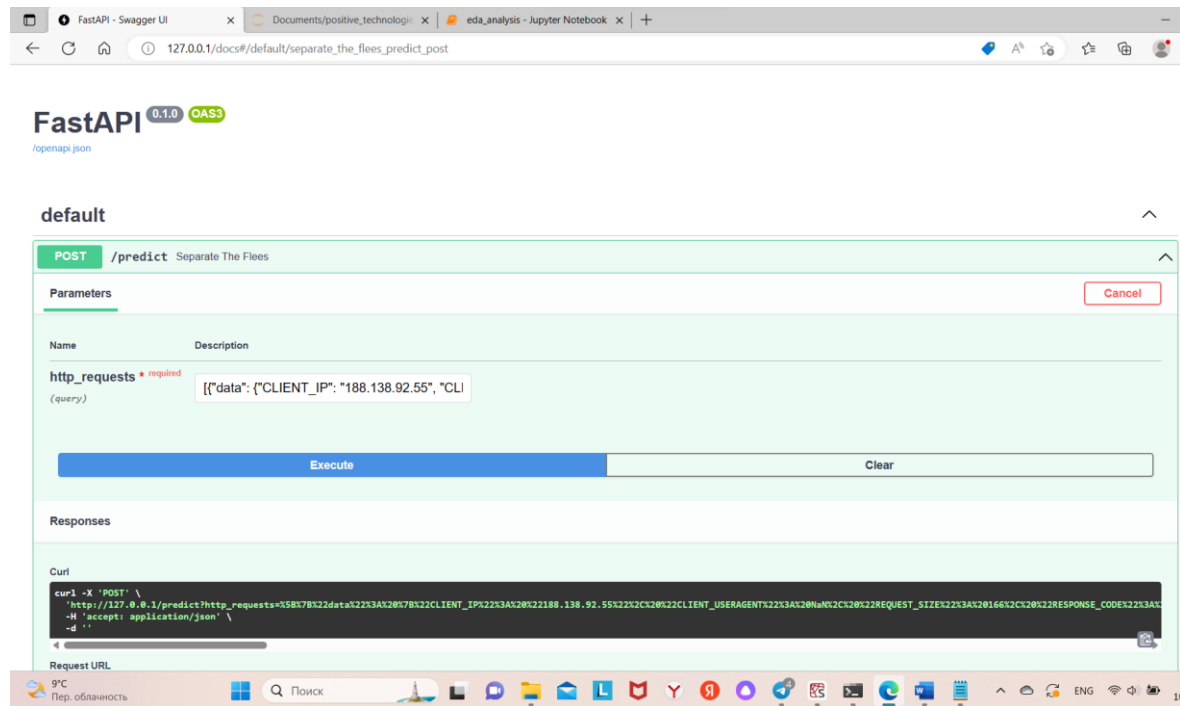
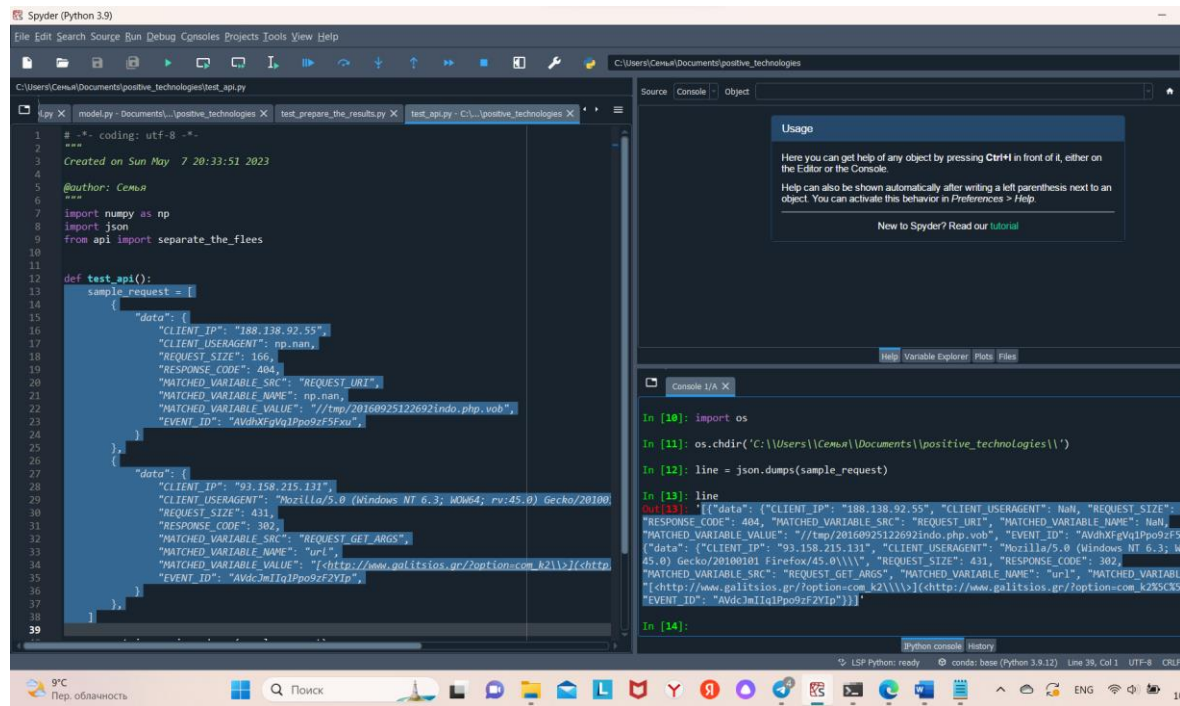
-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 6 passed, 2 warnings in 433.15s (0:07:13) =====
PS C:\Users\Семья\Documents\positive_technologies>
```

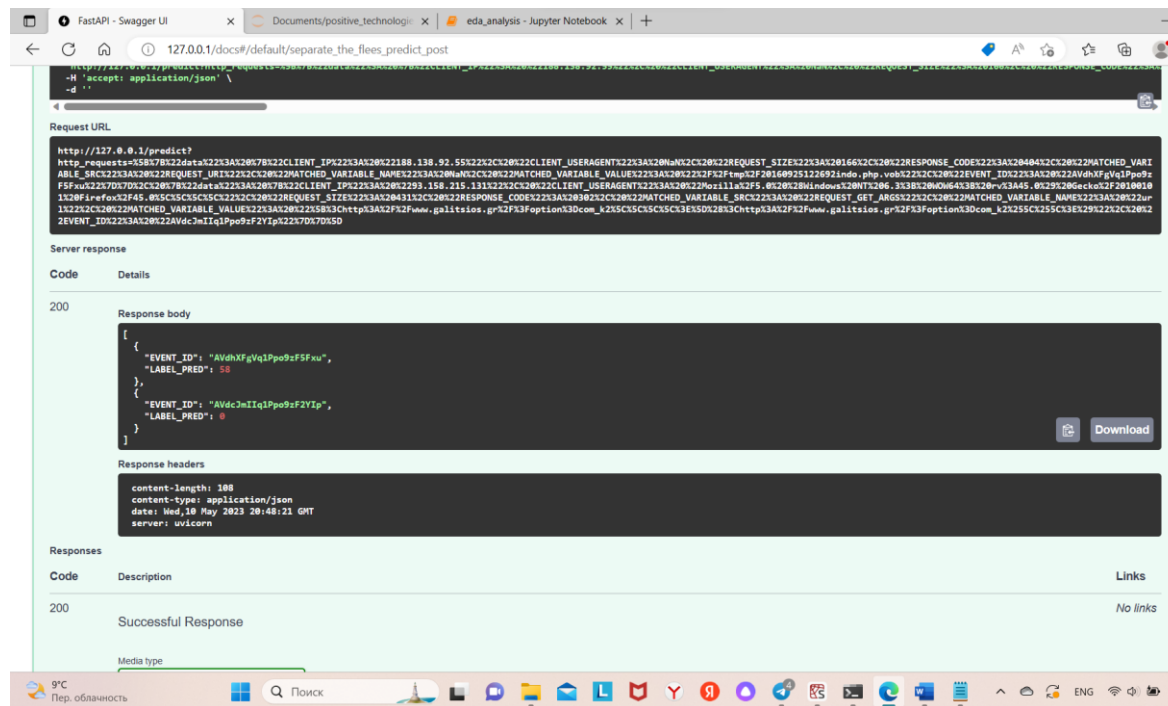
Запуск через докер.

```
Windows PowerShell
test_apply_the_model.py::test_apply_the_model
test_apply_the_model.py::test_apply_the_model_to_all
C:\Users\Семья\Documents\positive_technologies\processing_in_prod.py:82: FutureWarning: Passing 'suffixes' which cause duplicate columns {'index_x', 'result' is deprecated and will raise a MergeError in a future version.
    tab = pd.merge(

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 6 passed, 2 warnings in 433.15s (0:07:13) =====
PS C:\Users\Семья\Documents\positive_technologies> docker stop mycontainer
mycontainer
PS C:\Users\Семья\Documents\positive_technologies> docker rm mycontainer
mycontainer
PS C:\Users\Семья\Documents\positive_technologies> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
myimage latest a1681ff87676 22 minutes ago 1.33GB
PS C:\Users\Семья\Documents\positive_technologies> docker rmi a1681ff87676
Untagged: myimage:latest
Deleted: sha256:a1681ff8767605497331e602762bfac875b6e814cde67e268b834a3e2b0469d9
PS C:\Users\Семья\Documents\positive_technologies> docker build -t myimage . --file=Dockerfile.txt
[+] Building 4.0s (11/11) FINISHED
>> [internal] load .dockerignore
>> transferring context: 2B
>> [internal] load build definition from Dockerfile.txt
>> transferring dockerfile: 299B
>> [internal] load metadata for docker.io/library/python:3.9
>> [internal] load build context
>> transferring context: 50.21MB
>> [1/6] FROM docker.io/library/python:3.9@sha256:6ea9d9fc96d7914c5c1d199f1f0195c4e05cf017b10666ca84cb7ce8e2699d51
>> CACHED [2/6] RUN mkdir /code
>> CACHED [3/6] WORKDIR /code
>> CACHED [4/6] COPY requirements.txt /code/requirements.txt
>> CACHED [5/6] RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt
>> [6/6] COPY . /code/.
>> exporting to image
>> exporting layers
>> writing image sha256:14e7bdb5e05ffe5ab64a44dd3b0d1c6e58c967102acaf8c42a0f754d0b8e9e4e
>> naming to docker.io/library/myimage
PS C:\Users\Семья\Documents\positive_technologies> docker run -d --name mycontainer -p 80:80 myimage
1786ae7b70277ae6260c98a94ed25e53803ec0d6785fc4426af3c80466263068
PS C:\Users\Семья\Documents\positive_technologies>
```

Идём в файл **test_api.py**, берём пример **sample_request**.
Сериализуем его с помощью **json.dumps**.





Интерпретация выходных параметров.

Если в **LABEL_PRED** стоит пропущенное значение, то это не аномалия.

Если в **LABEL_PRED** стоит число от 0 до 53, то это один из аномальных кластеров, полученных HDBSCAN.

Если в **LABEL_PRED** стоит число от 54 до 58, то это один из аномальных кластеров, соответствующих пропущенным значениям.

Пример как автоматически присылать ответы.

Файл **test_apply_the_model.py**, функция

test_apply_the_model_to_all показывает как запустить основную логику автоматически на csv-файле.