Тестовое задание. ML в PT

Привет! Рады будем найти с тобой точки соприкосновения для работы.

Для этого нам было бы важно узнать твои сильные стороны, чтобы понять как их можно применить у нас. Для этого мы подготовили тестовое задание.

Задание рассчитано примерно на 2 вечера. Но в зависимости от опыта и перфекционизма, всё может затянуться. Поэтому по дефолту ожидается, что задание решается за неделю.

Мы не против уточнений, и если что-то непонятно, всегда можно задать вопрос Ире @IraZavyalova в ТГ.

Наташа передаст вопрос нам и поделится. А может быть у нее сразу уже будет ответ.

Теперь к самому заданию:

Задание состоит из 2х частей:

1. Решение ML задачи
2. Сервис для inference

🔥🔥🔥

Данные тут

🔥🔥🔥

1 ML задача

Тебе даны данные HTTP запросов.

Глобальная задача — найти или разделить вредоносные от хороших. Как мы знаем, вредоносных классов может быть несколько.

Важнее отделить «мух от котлет».

Попробуй сделать EDA, понять, а точно ли данные не избыточны и всё, что ты вообще можешь сказать?!

В ходе решения этой части ожидаем, что будут предоставлены артефакты (например jupyter notebook) с экспериментами, которые помогут понять, почему принято решение использовать такой подход к задаче.

2 Сервис для inference

Наши задачи заканчивается тогда, когда решение внедрено в прод.

Поэтому вторая часть посвящена inference части.

Представь, что те запросы — это поток данных, которые приходят, или ты можешь их забирать из какого-то внешнего сервиса.

Твоя задача — реализовать интерфейс сервис — принимать каким-либо образом запросы и отдавать ответы — номер класса. Ты можешь самостоятельно определить, что за что будет отвечать.

Можешь в тесты? - отлично, покажи что можешь. Можешь в Docker? - именно этого нам и надо! умеешь настраивать несложные CI — нам уже очень нравится!

У нас есть тестовые примеры, и мы бы хотели запустить их на твоем коде.

▼ Нам для проверки решения удобно, когда формат общений с сервисом унифицирован. Можно предложить свой вариант.

Но предлагаем посмотреть на openapi.json и реализовать метод `predict` например так:

Запрос

```
curl -X 'POST' \
'http://127.0.0.1:80/predict' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '[{"data": "{\"CLIENT_IP\": \"188.138.92.55\", \"CLIENT_USERAGENT\": NaN,
\"REQUEST_SIZE\": 166, \"RESPONSE_CODE\": 404, \"MATCHED_VARIABLE_SRC\":
\"REQUEST_URI\", \"MATCHED_VARIABLE_NAME\": NaN, \"MATCHED_VARIABLE_VALUE\":
\"//tmp/20160925122692indo.php.vob\", \"EVENT_ID\":
\"AVdhXFgVq1Ppo9zF5Fxu\"}"}, {"data": "{\"CLIENT_IP\": \"93.158.215.131\",
\"CLIENT_USERAGENT\": \"Mozilla/5.0 (Windows NT 6.3; WOW64; rv:45.0)
Gecko/20100101 Firefox/45.0\", \"REQUEST_SIZE\": 431, \"RESPONSE_CODE\":
302, \"MATCHED_VARIABLE_SRC\": \"REQUEST_GET_ARGS\",
\"MATCHED_VARIABLE_NAME\": \"url\", \"MATCHED_VARIABLE_VALUE\":
\"http://www.galitsios.gr/?option=com_k2\", \"EVENT_ID\":
\"AVdcJmIIq1Ppo9zF2YIp\"}"}]'
```

Ответ от твоего сервиса:

```
[
{
"EVENT_ID": "AVdhXFgVq1Ppo9zF5Fxu",
"LABEL_PRED": 42
},
{
"EVENT_ID": "AVdcJmIIq1Ppo9zF2YIp",
"LABEL_PRED": 3
}
]
```

Если приведешь пример, как можно автоматически присылать такие ответы, взяв за образец выданный csv, будет здорово.

> Решение мы ожидаем получить в виде доступа к приватному репозиторию. Для GitHub добавь пожалуйста: amurzina и nlyf. И сообщи пожалуйста Наташе @IraZavyalova когда готово, чтобы точно не потеряли.

▼ Подсказка 1

Классов может быть до 50, но не обязательно 50

▼ Подсказка 2

Кластеризация очень помогает

▼ Подсказка 3

Да, типично что у нас нет числа классов. Но что делать, это жизнь? Мы для базового решения использовали DBSCAN, может быть и ты можешь начать с него?

▼ Подсказка 4

посмотри на поля, точно ли все они нужны?