Идентификация ботов среди пользовательских сессий

Вы работаете в продуктовой компании в сфере розничной торговли, по данным исследований аналитиков ваши объемы продаж снизились из-за конкурентов, которые сбивают цены предлагая более низкие чем на вашей платформе. У аналитиков данных есть подтвержденная гипотеза о странном росте количества выбросов в поведении пользователей, которые скорей всего являются краулерами или ботами и возможно как раз именно они и собирают информацию о ценах на товары размещенные у вас конкурентам.

Вам поставлена задача реализовать классификатор таких сессии для того, чтобы отсеивать их показывая им капчу, но для этого необходимо понимать идентификатор подозрительной сессии (session_id).

Для начала изучите структуру предоставленных данных от инженеров которые были собранные за достаточно узкий промежуток времени и описывают поведение пользователей вашего сайта.

P.S. Для разработки лучше использовать PyCharm.

Структура исходных данных

Поле	Описание
session_id	уникальный идентификатор сессии пользователя
user_type	тип пользователя [authorized/guest]
duration	время длительности сессии (в секундах)
platform	платформа пользователя [web/ios/android]
item_info_events	число событий просмотра информации о товаре за сессию
select_item_events	число событий выбора товара за сессию
make_order_events	число событий оформления заказа на товар за сессию
events_per_min	число событий в минуту в среднем за сессию
is_bot	признак бота [0 - пользователь, 1 - бот]

Задача

Реализуйте две задачи для обучения лучшей модели данных и для ее применения.
Используйте за основу следующие шаблоны:

1) PySparkMLFit.py - шаблон для процесса обучения вашей модели, задача должна создавать план обучения модели с учетом оптимизации гиперпараметров, определять лучшую модель и сохранять ее для дальнейшего использования.

Параметры запуска задачи:
data_path - путь к файлу с данными
model_path - путь куда будет сохранена модель
Строка запуска:

python PySparkMLFit.py --data_path=session-stat.parquet --model_path=spark_ml_model # ИЛИ spark-submit PySparkMLFit.py --data_path=session-stat.parquet --model_path=spark_ml_model

2) PySparkMLPredict.py - шаблон для процесса применения вашей модели, задача должна загружать вашу модель, применять ее к указанному датасету и сохранять результат предсказаний в parquet формат содержащий всего две колонки - [session_id, prediction].

Параметры запуска задачи:
data_path - путь к файлу с данными (тестовый датасет)
model_path - путь откуда будет загружена модель
result_path - путь куда будет сохранен результат предсказаний
Строка запуска:

python PySparkMLPredict.py --data_path=test.parquet --model_path=spark_ml_model --result_path=result # ИЛИ spark-submit PySparkMLPredict.py --data_path=test.parquet --model_path=spark_ml_model --result_path=result

Предоставленные датасеты:

1) session-stat.parquet - файл с данным со статистикой по сессиям (как указано в таблице)

2) test.parquet - файл с данными со статистикой по сессиям, для оценки вашей модели (как указано в таблице, не имеет признака is_bot)