> Задание 1

Вы работаете над автоматизацией процессов страховой компании. Есть задача классификации водителей, по предсказанию вероятности аварии водителя в будущем году, на который водитель планирует приобрести страховку. Для этого вы используете собранную статистику по клиентам.

Baшего ML Engineer'a похитили пришельцы и пока рекрутеры ищут нового, вам нужно доделать задачу которую он оставил. Добавьте разметку эксперимента для трекинга процесса обучения модели в MLFlow.

MLFlow Host: https://mlflow.lab.karpov.courses

Приложения:

- 1) PySparkFit.py (процесс обучения модели)
- 2) **TRAIN.PARQUET** (данные для обучения модели)
- 3) **TEST. PARQUET** (данные для оценки модели)

Структура датасетов:

Поле	Описание
driver_id	уникальный идентификатор водителя
age	возраст водителя на момент анализа
sex	пол водителя
car_class	класс машины водителя
driving_experience	опыт вождения
speeding_penalties	количество штрафов за превышение скорости в течении года
parking_penalties	количество штрафов за неправильную парковку в течении года
total_car_accident	число аварий за весь опыт вождения
has_car_accident	идентификатор аварии в текущем году (целевой признак [0/1])

Команды:

python PySparkFit.py --train=train.parquet --test=test.parquet # ИЛИ spark-submit PySparkFit.py --train=train.parquet --test=test.parquet

Конфигурация ТОЛЬКО для ЛОКАЛЬНОЙ отладки:

#Вставьте в PySparkFit.py os.environ['MLFLOW_S3_ENDPOINT_URL'] = 'HTTPS://STORAGE.YANDEXCLOUD.NET' os.environ['AWS_ACCESS_KEY_ID'] = '<Ваш ключ из файла credentials>' os.environ['AWS_SECRET_ACCESS_KEY'] = '<Ваш ключ из файла credentials>'

Требования к логированию эксперимента

Ваш эксперимент должен называться в виде вашего логина!

Метрики: f1, weightedPrecision, weightedRecall, accuracy

Параметры:

input columns - список колонок исходного датасета (Пример: ['col1, 'col2'])

maxDepth - параметр максимальной глубины обученной модели

maxIter - параметр максимального числа итераций обучения модели

maxBins - параметр максимального числа ветвлений

target - целевая переменная для предсказаний

features - список колонок использованных для векторизации (Пример: ['col1, 'col2'])

stage_0 - Тип трансформера первого стейджа пайплайна (obj.class.name)

stage_1 - Тип трансформера второго стейджа пайплайна (obj.class.name)

stage 2 - Тип трансформера третьего стейджа пайплайна (obj.class.name)

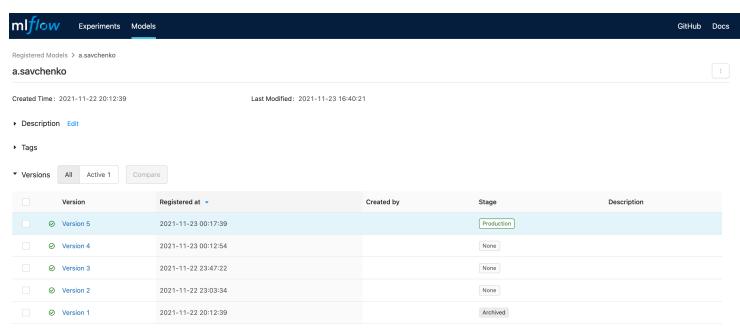
stage_3 - Тип трансформера четвертого стейджа пайплайна (obj.class.name)

Модель:

Baшa модель (registered_model_name) и ее размещение (artifact_path) так же должны быть в виде логина.

> Задание 2

В списке ваших моделей (зарегистрированные вашим экспериментом) раздела Models MLFLow назначьте работающую модель на ваш выбор как **Production** модель. Запустите проверку решения.



> Задание 3

Peaлизуйте задачу PySparkPredict.py которая будет загружать ВАШУ модель из MLFlow и применят к данным. Полученный датасет с предсказаниями сохраните в дерикторию результатов.

Убедитесь, что используете рабочую обученную модель.

Документация: MLFLOW.SPARK.LOAD_MODEL

Приложения:

- 1) PySparkPredict.py (шаблон)
- 2) <u>DATA.PARQUET</u> (данные для модели)

Команды:

python PySparkPredict.py --data=data.parquet --result=result #ИЛИ spark-submit PySparkPredict.py --data=data.parquet --result=result

Конфигурация ТОЛЬКО для ЛОКАЛЬНОЙ отладки:

#Вставьте в PySparkPredict.py os.environ['MLFLOW_S3_ENDPOINT_URL'] = 'HTTPS://STORAGE.YANDEXCLOUD.NET' os.environ['AWS_ACCESS_KEY_ID'] = '<Ваш ключ из файла credentials>' os.environ['AWS_SECRET_ACCESS_KEY'] = '<Ваш ключ из файла credentials>'

Отправленные файлы:

PySparkPredict.py