Домашнее задание №5mla. Тренировка модели с MLflow. Deadline 22 апреля в 17-59.

В этом задании вы обучаете модель sklearn и сохраняете ее и делаете предсказания с помощью MLFlow. Обратите внимание, в задании используется MLflow версии 2.10.2. Все выполняется только на логин-ноде. Кластер не используется.

Задание

Возьмите вашу модель и train.py из задания №1 и добавьте туда код для логирования параметров, метрики (log_loss) и модели используя библиотеку mlflow. Сделайте хотя бы один параметр модели, который должен передаваться в train.py (см. ниже)

Оформление

В вашем приватном репо ai-masters-bigdata создайте подпапку projects/5mla. В ней должен быть проект MLflow, то есть следующие файлы:

- MLProject - в нем команда для точки входа main должна иметь как минимум два параметра: train_path - путь к тренировочному датасету и model_param1 - какой-либо параметр вашей модели. Последний должен быть определен со значением по умолчанию.
- conda.yaml - файл для создания виртуальной среды питона с требуемыми вашей моделью пакетами.
- train.py - принимает как минимум два аргумента: train_path и model_param1

Вы можете оставить определение модели в model.py как в Задании 1, или поместить определение модели в тот же файл train.py

Проверка

Приготовление

Закомитьте ваш код в репозиторий и синхронизируйте с Github. Внимание, добавьте в .gitignore следующие строчки: mlruns, *.sqlite и убедитесь, что данные mlflow не попадают в ваш репо.

Если вы разрабатывали у себя на лаптопе или десктопе, зайдите на наш сервер, склонируйте ваш репо.

С помощью команды conda activate dsenv активируйте среду dsenv, в которой также установлен пакет mlflow.

В одном окне, в домашней директории, запустите tracking server с базой sqlite в качестве backend store, и директорией ./mlruns в качестве artifact store. Порт для tracking server должен быть 5000 + id вашего пользователя на сервере, который выдается командой id -u. Внимание! Если разорвалось соединение, то tracking server остается запущенным. Если надо его перезапустить, то ищите ваши процессы gunicorn и убивайте их.

В другом окне опять активируйте dsenv установите (export) переменную среды MLFLOW_TRACKING_URI со значением http://localhost:port - где port как описано выше. Перейдите в папку projects/5mla.

Далее, запустите тренировку модели командой:

mlflow run . -P train_path=/home/users/datasets/criteo/train1000.txt --env-manager=local

Последняя опция нужна для того, чтобы не создавать новую среду и исполнять в нашей активированной dsenv.

Убедитесь (например, с помощью mlflow ui или mlflow runs), что в бэкэнд записана информация об успешно прошедшем запуске: run id, параметры модели, среди который model_param1, метрика log_loss, и сама модель в artifacts.

Запустите сервис инференса обученной модели (mlflow models serve) на порту 6000 + id вашего пользователя на сервере. Обратите внимание на опцию --env-manager=local, которая позволяет обойтись без создания новой среды conda, обеспечивая полную совместимость. Если у вас на данном этапе странная ошибка, скорее всего создается новая среда с несовместимой версией mlflow.

Запустите чекер: checker.sh 5mla

Этапы проверки

На первом этапе чекер обратится к вашему трекеру для получения информации о последнем успешном запуске, его параметрах, модели. Затем сделает следующий запрос

с одной тестовой записью к вашему сервису предсказаний (используйте его для самопроверки):

curl http://127.0.0.1:port/invocations -H 'Content-Type: application/json' -d '{"dataframe_split": {"columns": ["if1", "if2", "if3", "if4", "if5", "if6", "if7", "if8", "if9", "if10", "if11", "if12", "if13", "cf1", "cf2", "cf3", "cf4", "cf5", "cf6", "cf7", "cf8", "cf9", "cf10", "cf11", "cf12", "cf13", "cf14", "cf15", "cf16", "cf17", "cf18", "cf19", "cf20", "cf21", "cf22", "cf23", "cf24", "cf25", "cf26"], "data": [[12,85,4,59,6,0,0,7,1,0,2,5878,4,"ad98e872","3dbb483e","9a898c89","67ecc871","1315f676","6fcd6dcb","0e005bd7","54cd7262","2e4e821f","62da11e3","14874876","11a67268","a77a4a56","be4ee537","a63cedcf","4cdc3efa","d20856aa","b8170bba","cc7a7d35","156cbe87","96fbe197","15562d5d","d3df7183","893704a5","ff654802","e1be5ef2"]]}}'

На втором этапе чекер склонирует ваш репо и запустит run самостоятельно и проверит, что этот запуск успешно логирован в ваш tracking server. В этом запуске будет использоваться –env-manager=local , то есть ваш conda.yaml будет исползоваться для создания новой среды.

Далее чекер запустит предсказание (mlflow models predict) на обученной модели и сравнит с ранее полученным результатом.

Работа чекера

Вы должны добиться PASSED 1, иначе смотрите сообщения чекера.

Перед запуском чекера в вашем tracking server должен быть успешный run. Это значит, что не получится просто запускать чекер снова после неудачной попытки. Нужно будет снова вручную запустить тренировку, в результате которой в трекинг сервере появится успешный run, после чего можно запускать чекер.