Wybrane zagadnienia uczenia maszynowego

MLOps

Wstęp

TODO 1

Przypominj sobie dotychczas realizowane zadania na laboratoriach i spróbuj odpowiedzieć na następujące pytania:

- w jaki sposób poszukiwałaś/poszukiwałeś najlepszego modelu realizującego określone zadania? Jak porównywałaś/porównywałeś wyniki działania?
- jak długo realizowane były operacje uczenia? Czy było to kilka-kilkanaście sekund, kilka minut, a może dłuższy czas?
- czy zdarzyła się sytuacja, że potrzebowałaś/potrzebowałeś wcześniej wytrenowanego modelu i jego wyników?
- czy zapisywałaś/zapisywałeś wyniki i/lub model? Czy wiesz jak to zrobić? Czy program, w którym zapisywałaś/zapisywałeś model nadpisywał go przy każdym uruchomieniu czy zapisywał jako osobny?

Odpowiedzi na powyższe pytania wskazują występującą trudność - jak zarządzać całym procesem uczenia maszynowego. Od etapu przygotowania danych (jakie dane zostały użyte? jak zostały podzielone? czy to powtarzalne? jak zostały przetworzone? jakie parametry miały wykorzystywane funkcje?) po proces treningu i eksperymentów (jakie parametry modeli? na jakich danych? czy da się je powtórzyć?) po etap ewaluacji (na jakim zbiorze testowym? jakie parametry) i zapisywania wyników. A to dopiero część całego procesu, gdyż do tej pory nasze opracowywane modele nie były wykorzystywane w finalnym produkcie (tzw. wdrażanie na produkcję). A wiąże się z tym kilka dodatkowych trudności - wersjonowanie, douczanie modeli, uruchomienie w środowisku docelowym, zarządzanie zależnościami, itp.

Więcej informacji można znaleźć w artykule na temat długu technologicznego w systemach uczenia maszynowego od Google (jeden z pierwszych na ten temat):

Hidden Technical Debt in Machine Learning Systems

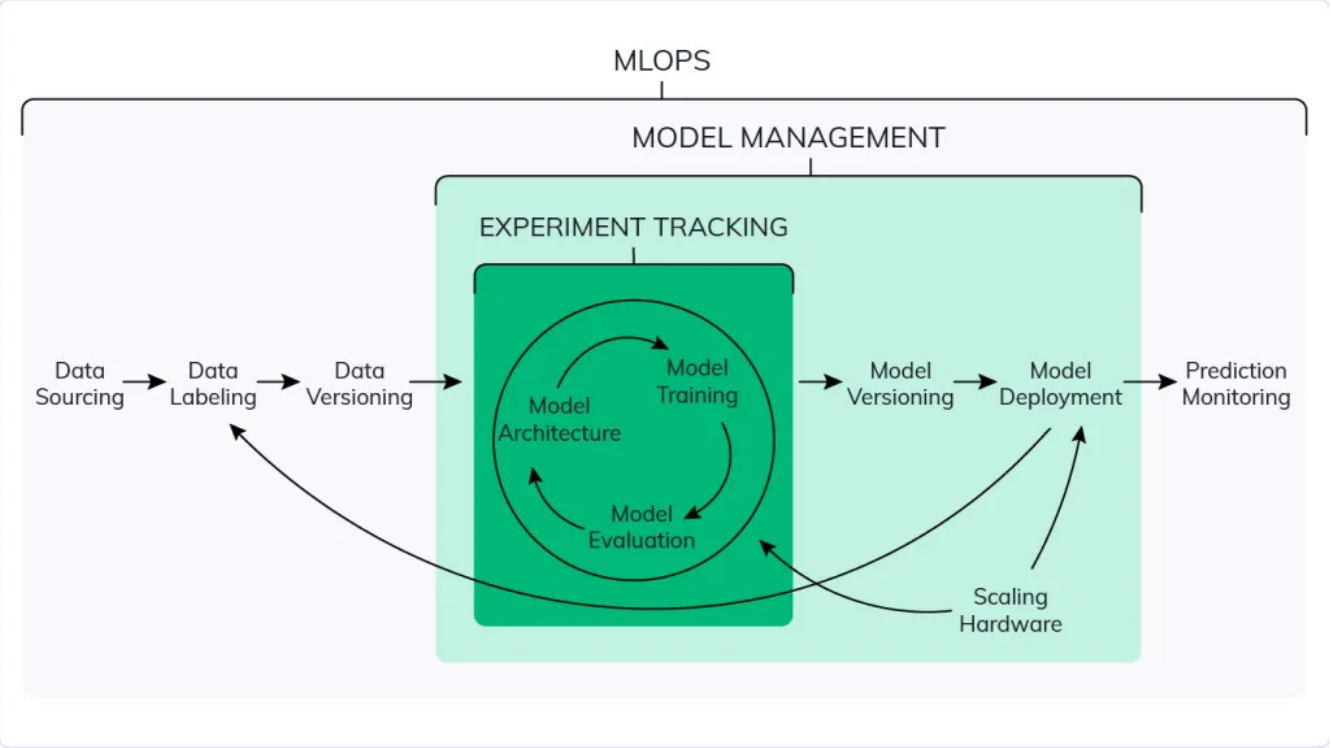
Dla procesu wytawarzanie standardowego oprogramowania (frontend/backend) przyjętym standardem zostało wykorzystanie narzędzi do wersjonowania kodu (git) oraz zestawu praktyk zwanych DevOps (software **Dev**elopment IT **Op**eration**s**).

Dla uczenia maszynowego zaproponowano analogiczną nazwę: **M**achine **L**earning **Op**erations - zbiór praktyk mających na celu wdrożenie i utrzymanie modeli uczenia maszynowego w środowisku produkcyjnym w sposób niezawodny i efektywny. [https://en.wikipedia.org/wiki/MLOps]

Uczenie maszynowe zyskało na popularności w ostatnich latach i tematyka związana z MLOps jest wciąż nowa, a standardy się zmieniają. Podobnie jak z systemami kontroli wersji i narzędziami do ciągłej integracji z czasem zapewne staną się narzędziem wykorzystywanym w każdym projekcie.

Przykładowy przebieg procesu tworzenia i wdrażania do produkcji modelu uczenia maszynowego przedstawiono na obrazku poniżej.

Źródło: https://neptune.ai/blog/mlops



MLflow

Popularną biblioteką umożliwiającą realizację dużej części z przedstawionych problemów jest MLflow (A Machine Learning Lifecycle Platform - https://github.com/mlflow/mlflow).

Składa się z kilku modułów, tj. [źródło]:

- MLflow Tracking: An API to log parameters, code, and results in machine learning experiments and compare them using an interactive UI.
- MLflow Projects: A code packaging format for reproducible runs using Conda and Docker, so you can share your ML code with others.

- MLflow Models: A model packaging format and tools that let you easily deploy the same model (from any ML library) to batch and real-time scoring on platforms such as Docker, Apache Spark, Azure ML and AWS SageMaker.
- MLflow Model Registry: A centralized model store, set of APIs, and UI, to collaboratively manage the full lifecycle of MLflow Models.

W ramach zajęć skupimy się na module Tracking do rejestrowania przebiegu i parametrów eksperymentów.

Uwaga! Ze względu na swoją specyfikę (domyślnie zapis wszystkich informacji jest do lokalnego systemu plików, a UI oferowane jest poprzez lokalny serwer www, który interpretuje te zapisane wyniki) uruchomienie powyższych przykładów w google colab nie będzie możliwe. W związku z tym proszę zadanie zrealizować lokalnie na komputerach. Google colab może logować z użyciem mlflow, ale skonfigurowany wówczas powinien być do współpracy z zewnętrznym serwerem, na który działa mlflow.

Uwaga! Niektóre kombinacje wersji `mlflow` i systemu operacyjnego wymagają dostępności polecenie `git` w konsoli. Dla pewności proszę zainstalować oprogramowanie git:

https://git-scm.com/downloads

i zapewnić aby polecenie było możliwe do wywołania w konsoli (dodanie git do path).

TODO 2

- Przeczytaj o założeniach *mlflow* - https://mlflow.org/docs/latest/concepts.html
- Zainstaluj bibliotekę *mlflow* i wykonaj pierwsze kroki: https://mlflow.org/docs/latest/quickstart.html (do Running MLflow Projects).
- Zapoznaj się z kodem przykładu i obejrzyj wyniki korzystając z *mlflow ui*.

TODO 3

- Wykorzystaj kod z jednych z poprzednich zajęć z trenowaniem i dodaj do niego autologowanie dla wykorzystywanej biblioteki (https://www.mlflow.org/docs/latest/tracking.html#automatic-logging).
- Sprawdź co dokładnie zostało zapisane w wyniku przeprowadzonego eksperymentu.
- Ustaw własną nazwę eksperymentu.

TODO 4

- Dodaj zapis własnego parametru (np. ile czasu zostało do deadline, poziom stresu, czy typ pogody).
- Wyznacz dodatkową metrykę dla wytrenowanego modelu i również ją zapisz

- Wyznacz dodatkową metrykę dla wytrenowanego modelu i również ją zapisz.
- Zapisz wytrenowany model jako artefakt.
- Sprawdź w UI czy dodane elementy zostały zapisane i są dostępne.

TODO 5

- Wykorzystaj UI do przeglądania wykonanych eksperymentów. Znajdź te, które zakończyły się sukcesem, a wartość wyniku jest wyższa niż dobrany próg (https://www.mlflow.org/docs/latest/search-syntax.html#syntax).
- Wykonaj powyższe zadanie realizując zapytanie za pośrednictwem kodu - https://www.mlflow.org/docs/latest/search-syntax.html#python

Za pomocą API możliwe jest przeszukiwanie całej bazy eksperymentów i np. automatyzacja wyboru modelu.

TODO 6

Poprzez UI znajdź eksperyment na dysku i otwórz tę lokalizację w przeglądarce plików. Następnie usuń go z listy w UI. Czy został usunięty z dysku?

Wykorzystaj: https://www.mlflow.org/docs/latest/cli.html#mlflow-gc

aby usunąć go permanentnie.

Jak przechowywane są dane?

TODO 7

- Przeczytaj https://www.mlflow.org/docs/latest/tracking.html#how-runs-and-artifacts-are-recorded (pierwsze dwa scenario).

Jak powiązać z kodem?

TODO 8

Stwórz repozytorium na wybranej paltformie (np. github) i umieść tam kod z trenowaniem modelu z mlflow. Uruchom go ponownie, zaraportowany powinien być hash commita (co to jest?).

Jak działa autolog?

Monkey patching!

https://bulldogjob.pl/news/1196-czym-jest-monkey-patching-w-pythonie

Inne systemy do mlops:

- https://www.datarobot.com/platform/mlops/
- https://neptune.ai/
- https://wandb.ai/site

Ciekawe linki, dodatkowe materiały:

- https://blogs.nvidia.com/blog/2020/09/03/what-is-mlops/
- https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning
- https://azure.microsoft.com/mediahandler/files/resourcefiles/gigaom-Delivering-on-the-Vision-of-MLOps/Delivering%20on%20the%20Vision%20of%20MLOps.pdf
- https://ml-ops.org/