MLOps Technical Challenge

1. Objective

The main purpose of this challenge is to assess your skills **(1)** building a scalable data pipeline for feature engineering & training machine learning model using good CI/CD practices **(2)** creating an API to serve those features, and **(3)** creating a prediction service that uses the created features.

2. Data & technology

You will be handed a dataset from Kaggle containing the credit default risk, a jupyter notebook containing code with some feature engineering and the training of a machine learning model.

The columns and their description of the dataset used are described in the references, on the last page of this document.

Both the generation of features and the machine learning model are toy examples, developed solely for the purpose of evaluating your skills in MLOps/DevOps. Skills in data science will not be assessed.

3. Problem context

A data scientist has engineered five new features that he will use for the development of some ML model. The code used to compute these features is contained in the jupyter notebook 1 you were handed. The problem here is that you want to serve the model to production using good CI/CD/CT practices.

Therefore, you in your role as Machine Learning Engineer must help the data scientist to achieve this task.

Also, you should keep in mind that the data generated here can be used by other data scientists to do analysis or even used for training other ML models. For example, the features generated in notebook 1, it is desirable to save them in a table, file, or even a feature store.

Having said this, we will now go to the detailed instructions of the challenge.

4. Instructions

4.1. Understand the code (notebook 1 & 2)

Take a look at the provided jupyter notebooks!

- In the Jupyter Notebook 1 the data scientist provided the code that generates the feature engineering.
- In the Jupyter Notebook 2, the training of the machine learning model is provided using the characteristics generated in notebook 1.

4.2. Build data pipelines to compute these features and train machine learning model.

With CI/CD (or CI/CD/CT) best practices, build a pipeline that automates the process of feature engineering and machine learning model training.

Add a manual approval step of the generated model (It doesn't matter if it's dummy example).

Considerations that may help you:

- Note that jupyter notebooks are .ipynb files, so depending on your solution you may need to convert these notebooks to a .py file.
- In notebook 1 the features for model training are written in a .csv file. Consider if this is the best way to save the file, considering the following points of the challenge.

Explain the reason of your choice. Are you sure (or at least to some point) of the quality of your output?

Note that this task may be related to 4.5

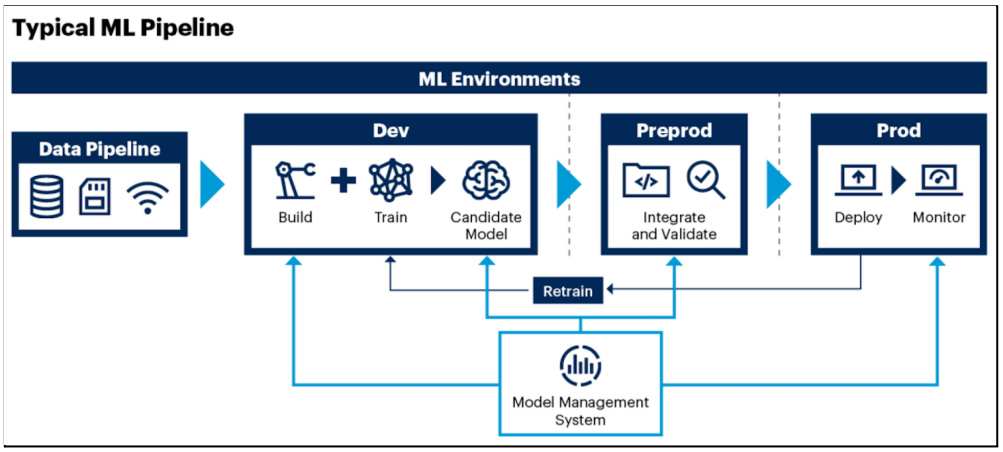

Figure 1. Typical ML Pipeline

4.3. Create an API to serve the features

ML models might use those five features for online evaluation. We would like to have an API serving the features.

Create a local API which:

- receives: the **id** of a user
- returns:
  - **nb_previous_loans**: number of loans granted to a given user, before the current loan.

- **avg_amount_loans_previous**: average amount of loans granted to a user, before the current loan.
- **age**: user age in years.
- **years_on_the_job**: years the user has been in employment.
- **flag_own_car**: flag that indicates if the user has his own car.

Consider that each user will have features at different moments of time, so you must bear in mind that only the most recent (last date per user) are of interest for the prediction.

4.4. Create an API to make predictions

Create an API that, given a user id, makes a prediction of credit risk using the model trained in the notebook 2. Consider using the features already created and available on the endpoint, in point 4.3.
At the end of notebook 2 it is shown how to load a model into memory from the already trained model and make a prediction.

4.5. Create ML Pipeline Service

Create a continuous integration and continuous deployment (CI/CD) pipeline to deploy the previous microservices.
An example of an ideal case can be seen in the figure below.

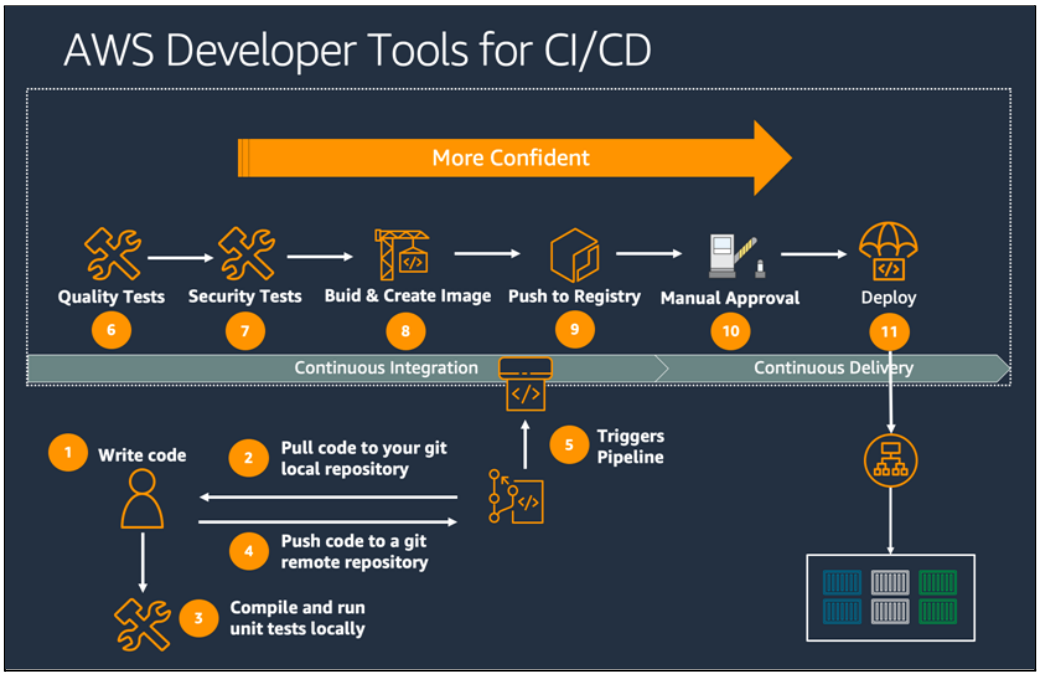

Figure 2. Example of CI/CD workflow.

About the solution to deliver

- You are expected to submit all code for the solution you submit to a code repository.
- You can use open source technologies on premises or cloud services. It is desirable that you use tools like Docker for packaging or similar.
- Hopefully microservices can be called from a CURL, Postman, cloud provider instances, etc.

To consider: *In this exercise there might be some things which are not a very good representation of a real life use case (for example: the structure/format of the source data, the architecture of the API and the data pipeline, the infrastructure, the QA considerations, etc.). Tell us what considerations you would have regarding this API for a real life situation.*

In the event that the exercise is not resolved 100%, we will consider the form adopted for the resolution, proposed design, etc.

About the evaluation: *The topics that are evaluated on the challenge are the following, with priority in the order that they are located in the list. The disapproval of one or more points listed, is not necessarily the disapproval of the challenge :)*

The solutions that are sent and are not within a code repository, are not evaluated.

1. DevOps/MLOps skills
2. Solution Architecture
3. Coding practices
4. Cloud skills
5. Non functional requirements
6. Documentation

Links of interest

MLOps:
https://ml-ops.org/

MLOps by Google:
https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

References:

Columns of the dataset used in notebook 1.

loan_id: loan number
id: client number
code_gender: gender
flag_own_car: if users have a car
flag_own_realty: is there a property
cnt_children: number of children
amt_income_total: annual income
name_income_type: income category
name_education_type: education level
name_family_status: marital status
name_housing_type: way of living
flag_mobil: is there a mobile phone
flag_work_phone: is there a work phone
flag_phone: is there a phone
flag_email: is there an email
occupation_type: occupation
cnt_fam_members: family size
status: 0->paid on time 1->Not paid on time
birthday: birthday
job_start_date: job start date
loan_date: loan date
loan_amount: loan amount