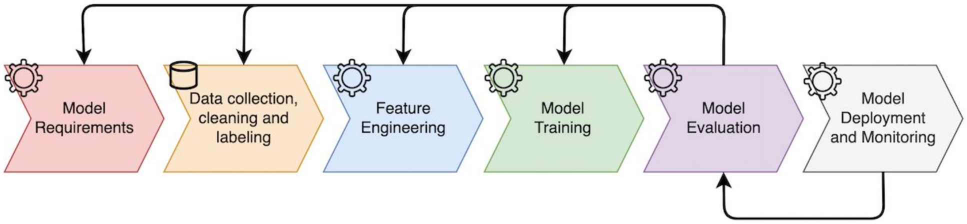
Interview protocol

➢ Give an overview of the Machine Learning development stages presented in the workflow:


Model Requirements
Data collection, cleaning and labeling
Feature Engineering
Model Training
Model Evaluation
Model Deployment and Monitoring

Machine Learning Workflow [1].

Questions:

- Usually, who is involved in executing each stage?
- Which stage(s) do you often work on?
- Which stage do you have the most experience with?
  To measure the experience you must consider the number of projects or the time you worked on each step.

➢ Initially, **we will ask developers the general and specific questions only about the stage where he/she said to have more experience** with.

➢ Finally, **we will ask developers the general questions from the stages where he/she said to work** with.

➢ **General and specific questions of all stages:**

- **Model Requirements**

  General questions:

  - How do you specify the model requirements?
  - How do you verify whether the products of this stage (the model requirements) satisfy the conditions imposed by the client?

    Verification: are we building the product right?
    Verification: the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610]
  - What are the biggest challenges in executing the *Model Requirement* stage?

Specific questions:

1. How the customers' needs are collected and specified for machine learning models?

2. What are the most common functional and nonfunctional requirements for a machine learning model?
   Examples: maintainability, architecture, scalability, security, reliability, interpretability, fairness.

3. How do you verify requirements for completeness, correctness and Testability?

4. How to mitigate incomplete or incorrect requirements?

5. How do you verify requirements violation or compliance and its potential causes?

- ## Data collection, cleaning, and labeling

  General questions:

  - How do you collect, clean and label data?
  - How do you verify whether the results of this stage (dataset) satisfy the conditions imposed at the start of this same stage?
    Verification: are we building the product right?
    Verification: the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610]
  - What are the biggest challenges in executing *Data collection, Cleaning and Labelling*?

  Specific questions:

  Data Collection

  1. How do you choose the data sources to build a machine learning model?
     - Do you use multi source data?
     - Do you use synthetic data generation?

  2. What mechanisms are used to collect data?
     - How to ensure the reliability of these mechanisms? In other words, how to ensure that the employed mechanisms are properly collecting the data as expected?

  3. Are the data collected usually validated? How is the data validated?
     Examples: Detecting unreliable data entries, instances violations, schema violations, structural conflicts, data drifts, wrong computations and others.

Data Cleaning

1. How do you clean data?
   - Which tools are used?

2. How are outliers in attributes identified and handled?
   - What are the criteria to identify outliers?

3. How do you establish the data type of each attribute?
   - What are the criteria to define the features' data type?

4. How do you establish the scale of each attribute?
   - What are the criteria to define the features’ scales?

5. How do you handle invalid values of attributes?
   Examples: delete instances with missing values or use default values (enforce schema constraints).

6. What do you identify as **dirty data**?

Data Labeling (only for supervised learning)

1. How do you label instances?
   - Do instances come from the data collection already labeled?
   - Do you automatically or manually label the data?
     - Automatically: How automatically do you perform the data labeling?
       - System execution? Simulation? Static analysis? Synthetic data?
     - Manually: Who performs data labeling and how?
       - Do people with little understanding of the application domain perform data labeling?

2. Is there a process to assess the quality of the labeled data?

Code

1. How do you assess the quality of the code for data processing?

2. What are the main problems that can be triggered by an error in the data processing code?

Feature Engineering

General questions:

- How do you perform feature engineering?
- How do you verify whether the products of this stage (features for training) satisfy the conditions imposed at the start of this same stage?

  Verification: are we building the product right?

Verification: the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610]

- What are the biggest challenges in executing *Feature Engineering*?

Specific questions:

1. How do you identify irrelevant features?
   Irrelevant features *are those that minimally impact the model results, either positively or negatively. Leaving irrelevant features in the model implies more complexity and opportunities for failures.*

2. How do you detect and remove dependencies between features?
   For some algorithms, the features must be as independent as possible. Dependent features could negatively influence the model training and results.

3. How do you identify opportunities for feature transformations?
   Transformations refer to operations to create new features using existing features to improve model accuracy.

4. How important is the presence of an expert in the application domain, for feature engineering?
   - Who is the expert?
   - What problems could the expert help to identify?

5. What is the relevance of data enrichment on the model quality?
   - Please, could you give an example?

6. How do you assess that the adopted features are the best for training your model?

Code

7. How do you assess the quality of the code for feature selection?

Model Training

General questions:

- How do you perform the model training?
- How do you verify whether the products of this stage (trained model) satisfy the conditions imposed at the start of this same stage?

Verification: are we building the product right?
Verification: the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610]

- What are the biggest challenges in executing *Model Training*?

Specific questions:

1. How do you choose training, testing and validation basis?

- How much train, test and validation data do you use?
- Do you use any automated strategy to perform this task?
  Example: hold-Out, Cross-Validation, Bootstrapping or Dynamic Sector Validation.

2. How do you ensure that training data does not differ from real data?

3. How do you select **algorithms** and **hyperparameters**?

- ○ Do you compare algorithms with baseline implementations?
- ○ Do you use optimizers to set hyperparameters?
  - ■ What optimizer(s) do you use?
- ○ Do you manually set hyperparameters?
  - ■ Is it based on expert knowledge?

4. How to deal with data drift? Is the model regularly retrained on recent data to detect decaying model performance and adjust the hyperparameters?
   Examples: monitoring thresholds.

5. How do you assess that the algorithms and parameters chosen for the model training were the best?

Code

6. How do you implement the machine learning algorithm used to train the model? Do you implement it from scratch? Or do you use third-party libraries?

- ○ Why?
- ○ What third-party libraries do you use?

7. How do you ensure that libraries for training a model are **correctly implemented**?

- ○ How to test the code?
- ○ How to debug the code?

8. What mechanisms can be used to assess the **correctness of the code** for training a model?

● Model Evaluation

General questions:

- ■ How do you perform the model evaluation?
- ■ How do you verify whether the products of this stage (model evaluation) satisfy the conditions imposed at the start of this same stage?
  Verification: are we building the product right?
  Verification: the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610]
- ■ What are the biggest challenges in executing the *Model Evaluation?*

Specific questions:

1. How do you evaluate the model using the validation set?

2. How to define the most appropriate metrics for evaluation of the model?
   - ○ Is it defined by an expert in the application domain? Or does it exclusively depend on the adopted machine learning algorithm? Do you compare different metrics?
   - ○ How do you define evaluation metrics target values?
   - ○ Do you consider performance metrics, energy consumption or machine resources to avoid problems in the production environment?

3. Do you usually collect feedback from users to improve the model quality?

4. How is **overfitting** identified and adjusted in a model?
   Overfitting *occurs when a model is too complex for the data, even the noise of training data is fitted by the model [2].*

5. How is **underfitting** identified and adjusted in a model?
   Underfitting *occurs when a learner has low training accuracy regardless of the presence of noise [2].*

6. How to ensure that a machine learning model is **robust**?
   Robustness *is a non-functional characteristic of a machine learning system. A natural way to measure robustness is to check the correctness of the system with the existence of noise [2].*
   - ○ How to analyze and mitigate wrong results?

7. How to ensure that a machine learning model is **interpretable**?
   There are two types of interpretability: global interpretability means understanding the entirety of a trained model; local interpretability means understanding the results of a trained model on a specific input and the corresponding output [2].
   - ○ How to measure and guarantee *interpretability*?

8. Is the assessment of the aforementioned properties (metrics, overfitting, underfitting, robustness and interpretability) enough to ensure that a model is ready for deployment?

Code

9. How do you implement the metrics used to evaluate the model? Do you use third-party libraries? What third-party libraries do you use?

- ## Model Deployment and Monitoring

  General questions:

  - ■ How do you perform the model deployment and monitoring?

- How do you verify whether the products of this stage (a monitored model in production) satisfy the conditions imposed at the start of this same stage?

  Verification: are we building the product right?

  Verification: the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610]
- What are the biggest challenges in executing *Model Deployment and Monitoring*?

Specific questions:

1. After the deployment, which **characteristics of the model** should be monitored?
   - Why should a model be monitored?

2. What problems usually occur after the deployment of the model which **requires maintenance**? Why should a model be monitored?
   - How to detect poor data quality, poor model quality, and data drift?
   - Is the model regularly retrained on recent data to detect decaying model performance and adjust the hyperparameters?

3. What mechanisms can be used to **monitor a deployed model**?

Participants' background

1. What is your role in the organization?
2. How many years of experience do you have with Machine Learning?
3. What kind of learning do you currently work with?
   Possible examples: unsupervised learning, supervised learning and reinforcement learning. Subtasks: classification, regression, object detection, image classification.
4. What machine learning technologies have you used recently?

References

[1] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software engineering for machine learning: a case study. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice* (*ICSE-SEIP '19*). IEEE Press, 291–300. DOI:https://doi.org/10.1109/ICSE-SEIP.2019.00042

[2] Jie M. Zhang (1), Mark Harman (1 and 2), Lei Ma (3), Yang Liu (4). Machine Learning Testing: Survey, Landscapes and Horizons. https://arxiv.org/abs/1906.10742