CRISP-ML(Q)

Lifecycle process

Marcel van Velzen

Junior Marte Garcia

CRISP-ML(Q)

The Cross-Industry Standard Process for the development of Machine Learning applications with Quality assurance methodology (**CRISP-ML(Q)**) guides ML practitioners through the development life cycle. It helps to maintain quality, sustainability, robustness, and cost management throughout the ML life cycle.

CRISP-ML(Q) is an upgraded version of Cross Industry Standard Process for Data Mining (CRISP-DM) to ensure quality products.

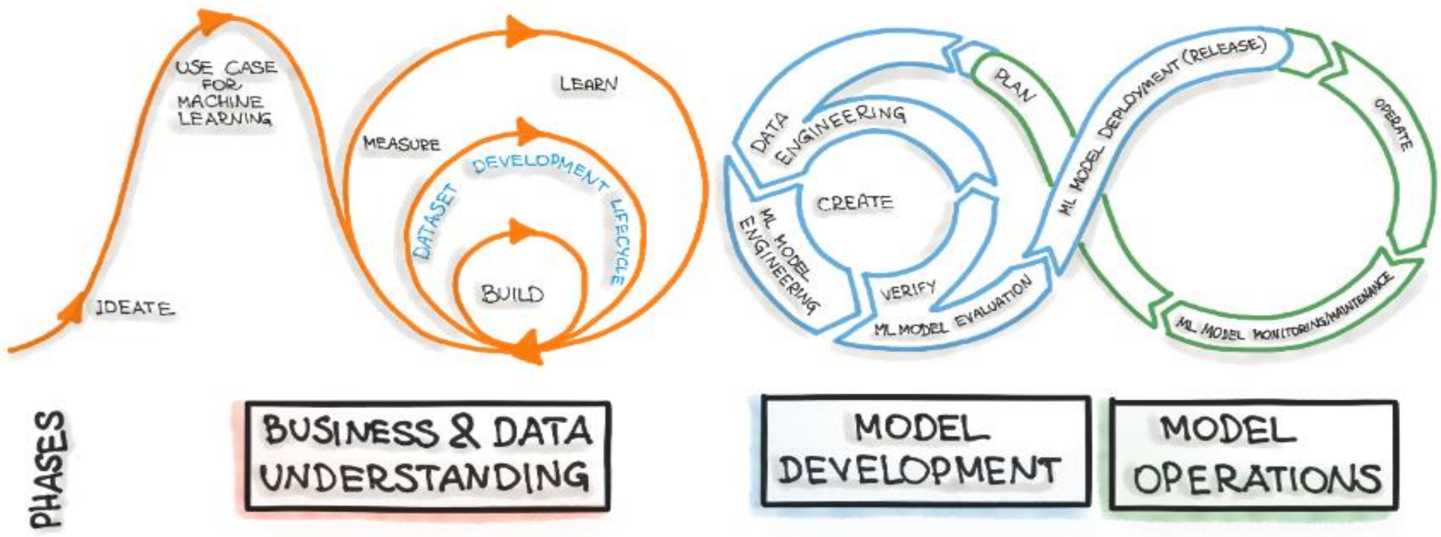
CRISP-ML(Q) is designed for the development of machine applications i.e. application scenarios where a ML model is deployed and maintained as part of a product or service

- **Quality assurance methodology is introduced in each phase and task**
  The quality methodology serves to mitigate risks that affect the success and efficiency of the machine learning application.

- **Covers a monitoring and maintenance phase**
  To address risks of model degradation in a changing environment and this extends the scope of the process model as compared to CRISP-DM.

- **Business and Data Understanding are merged into a single phase**
  Industry practice has taught that these two activities, which are separate in CRISP-DM, are strongly intertwined.

ML Development Life Cycle Process



It starts with an idea and defining the use case for machine learning, followed by understanding the business problem and the data. The next part is modeling and deployment of the models, including monitoring the installed models. Operation includes data visualization and the development of intelligent applications. The arrows illustrate the cyclically of the development.

Many phases in ML development are iterative. Sometimes, we might need to review the business goals, KPIs, and available data from the previous steps to adjust the outcomes of the ML model results.

Phases

CRISP-ML(Q) has six individual phases:

1. **Business and Data Understanding**
   Developing machine learning applications starts with identifying the scope of the ML application, the success criteria, and a data quality verification. The goal of this first phase is to ensure the feasibility of the project.

2. **Data Engineering (Data Preparation)**
   Prepare data for the following modeling phase.

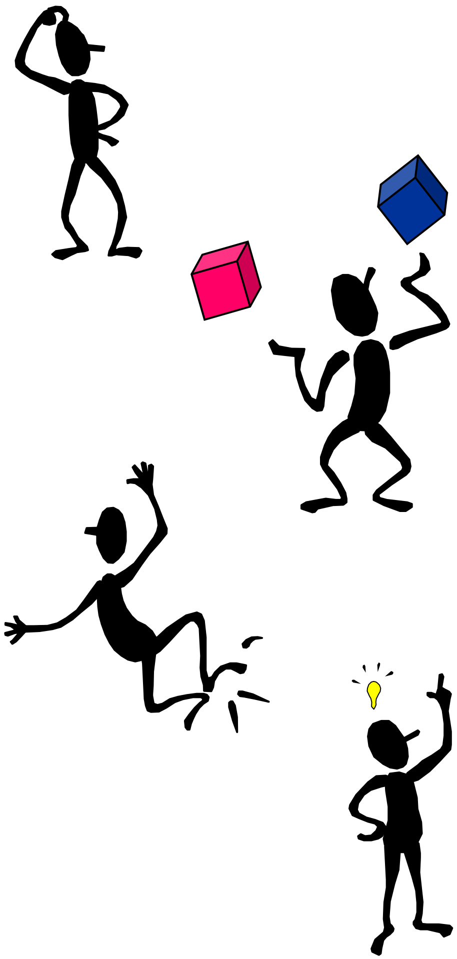
3. **Model Engineering**
   Specify one or several machine learning models to be deployed in the production.

4. **Model Evaluation**
   The performance of the trained model needs to be validated. Additionally, the model robustness should be assessed. Furthermore, it is best practice to develop an explainable ML model.

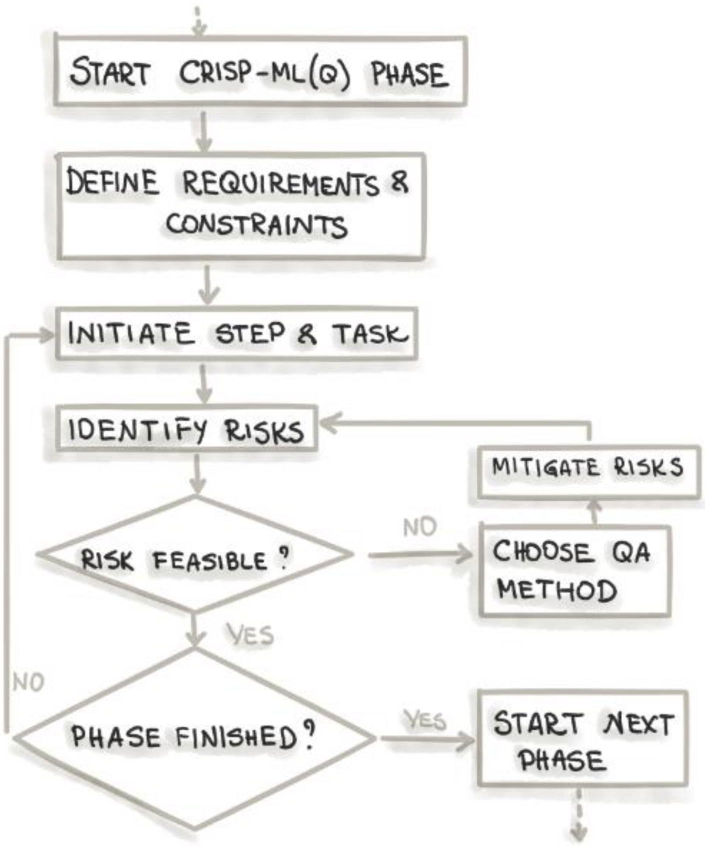
5. **Model Deployment**
   The ML model is graduated to be deployed in the (pre-) production environment.

6. **Monitoring and Maintenance**
   It is essential to monitor the performance of the ML model and to maintain it.

These phases require constant iteration and exploration for building better solutions. Even though there is an order in a framework, the output of the later phase can determine whether we have to re-examine the previous phase or not.



Quality Assurance



In the **CRISP-ML(Q)** methodology, each phase of the process model incorporates a quality assurance approach.

This approach involves several steps.

First, it requires defining the requirements and constraints for the project, such as performance expectations, data quality requirements, and model robustness.

Next, specific tasks are instantiated, which may include ML algorithm selection and model training, among others.

Additionally, potential risks that could hinder the efficiency and success of the ML application are identified, such as bias, overfitting, and lack of reproducibility.

To address these risks, appropriate quality assurance methods are employed.

These methods aim to mitigate the identified risks and may include techniques like cross-validation and documenting the process and results.

Business and Data Understanding

The Business and Data Understanding phase has the following tasks:

- Define business objectives
- Translate business objectives into ML objectives
- Collect and verify data
- Assess the project feasibility
- Create POC

Some potential risks in this phase:

1. Undefined or ambiguous business objectives: Lack of clarity in defining the business objectives can lead to misalignment between the ML objectives and the overall project goals.
2. Inadequate data collection or verification: Insufficient or unreliable data can hinder the development of accurate and effective machine learning models.
3. Infeasible project scope: Unrealistic expectations or constraints that make the project unachievable within given resources or timeframes can jeopardize its success.

Some potential quality assurance methods in this phase:

1. Stakeholder collaboration: Engage with stakeholders to clearly define and validate the business objectives, ensuring a shared understanding and alignment between the ML objectives and overall project goals.
2. Data quality assessment: Perform data profiling, conduct data quality checks, and verify the availability and suitability of data sources to ensure the reliability and relevance of the data used in the project.
3. Feasibility assessment: Evaluate the project's feasibility by considering the available resources, timelines, and technical requirements, conducting a proof of concept (POC) to validate the approach and demonstrate its potential value.

It's important to note that the specific risks and quality assurance methods may vary depending on the project context and requirements.

Data Engineering

The Data Engineering phase has the following tasks:

- Feature selection
- Data selection
- Class balancing
- Cleaning data (noise reduction, data imputation)
- Feature engineering (data construction)
- Data augmentation
- Data standardization

Some potential risks in this phase:

1. Incomplete or inconsistent data: Data may have missing values, outliers, or inconsistencies that can impact the quality and reliability of the subsequent modeling phase.
2. Biased or unrepresentative data: The data used for modeling may not adequately represent the real-world scenarios, leading to biased or less accurate models.
3. Incorrect feature selection or engineering: Inappropriate selection or engineering of features can result in models that are sensitive to irrelevant or noisy information, affecting their performance and interpretability.

Some potential quality assurance methods in this phase:

1. Data profiling and cleaning: Conduct thorough data analysis to identify missing values, outliers, inconsistencies, and apply appropriate techniques like imputation, noise reduction, or data cleaning to ensure data quality.
2. Data sampling and balancing: Address class imbalance issues by employing techniques like oversampling, undersampling, or generating synthetic samples to create a more balanced representation of the data.
3. Feature engineering validation: Assess the impact of feature selection and engineering techniques on the model's performance, interpretability, and generalizability through proper validation and evaluation methods.

It's important to note that the specific risks and quality assurance methods may vary depending on the project context and requirements.

Model Engineering

The Model Engineering phase has the following tasks:

- Define quality measure of the model
- ML algorithm selection (baseline selection)
- Adding domain knowledge to specialize the model
- Model training optional using pre-trained models
- Model compression
- Ensemble learning
- Documenting the ML model and experiments

Some potential risks in this phase:

1. Model underperformance: The selected machine learning model may fail to meet the defined quality measures or performance expectations, resulting in suboptimal results.
2. Lack of generalizability: The model may not generalize well to unseen data or different contexts, leading to poor performance in real-world scenarios.
3. Model complexity: Overly complex models can be challenging to interpret, maintain, and deploy efficiently, impacting the overall efficiency of the application.

Some potential quality assurance methods in this phase:

1. Quality measure definition: Clearly define the quality measures and evaluation metrics that the model needs to achieve, ensuring they align with the project objectives and desired performance criteria.
2. Model validation and comparison: Evaluate and compare different ML algorithms or model variants to select the most appropriate one based on performance, interpretability, and feasibility for deployment.
3. Documentation and experiment tracking: Maintain thorough documentation of the model development process, including the steps taken, parameters used, and experimental results. This enables reproducibility, traceability, and transparency of the ML model and experiments.

It's important to note that the specific risks and quality assurance methods may vary depending on the project context and requirements.

Model Evaluation

The Model Evaluation phase has the following tasks:

- Validate model's performance
- Determine robustess
- Increase model's explainability
- Make a decision whether to deploy the model
- Document the evaluation phase

Some potential risks in this phase:

1. Poor model performance: The trained model may not achieve the desired level of performance, such as accuracy, precision, recall, or other evaluation metrics, leading to suboptimal results.
2. Lack of model robustness: The model may perform well on the training data but fail to generalize to unseen or real-world data, indicating a lack of robustness.
3. Lack of explainability: If the model's decision-making process is not transparent or interpretable, it may be challenging to understand and justify its predictions or classifications.

Some potential quality assurance methods in this phase:

1. Performance validation: Conduct thorough performance evaluation of the trained model using appropriate evaluation metrics and validation techniques, such as cross-validation or hold-out validation, to assess its accuracy, robustness, and generalization capabilities.
2. Robustness assessment: Test the model's performance on various datasets or evaluate its sensitivity to perturbations and variations to determine its robustness under different conditions.
3. Explainability enhancement: Utilize techniques such as feature importance analysis, model interpretability methods or using inherently explainable models (e.g., decision trees) to increase the transparency and interpretability of the model's predictions.

It's important to note that the specific risks and quality assurance methods may vary depending on the project context and requirements.

Model Deployment

The Model Deployment phase has the following tasks:

- Evaluate model under production condition
- Assure user acceptance and usability
- Model governance
- Deploy according to the selected strategy (A/B testing, multi-armed bandits)

Some potential risks in this phase:

1. Poor performance in the production environment: The deployed model may not perform as expected when applied to real-world data, leading to suboptimal results and decreased efficiency.
2. User acceptance and usability issues: Users may have difficulties understanding or using the deployed model, which can hinder its adoption and effectiveness.
3. Model governance challenges: Ensuring proper governance and monitoring of the deployed model, including version control, tracking changes, and compliance with regulations, may pose challenges.

Some potential quality assurance methods in this phase:

1. Production environment evaluation: Assess the performance and behavior of the deployed model under realistic production conditions to verify that it meets the desired performance standards and operates effectively.
2. User acceptance testing: Conduct usability testing and gather feedback from users to ensure that the deployed model is user-friendly, understandable, and meets their needs and expectations.
3. Model governance and monitoring: Implement processes and mechanisms to govern the deployed model, including version control, documentation, and ongoing monitoring to ensure its reliability, accuracy, and compliance with relevant regulations or standards.
4. Deployment strategy testing: Employ strategies such as A/B testing or multi-armed bandits to validate the deployment strategy and compare the performance of different versions or configurations of the model in a controlled manner.

It's important to note that the specific risks and quality assurance methods may vary depending on the project context and requirements.

Monitoring and Maintenance

The Monitoring and Maintenance phase has the following tasks:

- Monitor the efficiency and efficacy of the model prediction serving
- Compare to the previously specified success criteria (thresholds)
- Retrain model if required
- Collect new data
- Perform labelling of the new data points
- Repeat tasks from the Model Engineering and Model Evaluation phases
- Continuous, integration, training, and deployment of the model

Some potential risks in this phase:

1. Degradation of model performance: Over time, the performance of the ML model may deteriorate due to changes in data patterns, shifting user behavior, or other external factors, leading to reduced accuracy or efficacy.
2. Data quality issues: New data collected for model updates may contain errors, missing values, or inconsistencies, impacting the model's performance and reliability.
3. Drift and concept shift: The underlying data distribution or relationship between features may change over time, causing the model to become outdated and less effective in making accurate predictions.

Some potential quality assurance methods in this phase:

1. Performance monitoring: Continuously monitor the model's performance to detect degradation and compare it against predefined success criteria or thresholds.
2. Retraining and updating the model: If the model's performance deteriorates or if significant changes in data patterns are observed, retrain the model using the new data and perform necessary feature engineering or algorithm adjustments to maintain or improve performance.
3. Data quality control: Implement robust data quality assurance processes to detect and address issues in the new data collected for model updates. This can include data cleaning, validation, and verification techniques.

It's important to note that the specific risks and quality assurance methods may vary depending on the project context and requirements.

Strengths and weaknesses

Strengths of CRISP-ML(Q):

- ❑ Structured approach guides the entire machine learning project
- ❑ Feasibility assessment ensures project viability
- ❑ Quality assurance approach throughout each phase, ensuring the reliability, robustness, and overall quality of the machine learning application by addressing potential risks
- ❑ Dedicated phase for model evaluation ensures performance, robustness, and explainability
- ❑ Emphasis on monitoring and maintenance for long-term effectiveness

Weaknesses of CRISP-ML(Q):

- ❑ Data engineering phase may lack sufficient focus
- ❑ Limited coverage of ethical considerations

References

Awan, A. A. (n.d.). The Machine Learning Lifecycle - KDNuggets. KDnuggets. https://www.kdnuggets.com/2022/06/making-sense-crispmlq-machine-learning-lifecycle-process.html

Hanzelik, P. P., Kummer, A., & Abonyi, J. (2022). Edge-Computing and Machine-Learning-Based framework for software sensor development. Sensors, 22(11), 4268. https://doi.org/10.3390/s22114268

Ml-ops.org. (2023, February 22). https://ml-ops.org/content/crisp-ml

Studer, S., Binh Bui, T., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., & Müller, K.-R. (2021). Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. arXiv. https://arxiv.org/pdf/2003.05155.pdf

Sudheer, P. (n.d.). Short note on CRISP-ML(Q) and Quality Involved. www.linkedin.com. https://www.linkedin.com/pulse/short-note-crisp-mlq-quality-involved-pavan-sudheer/