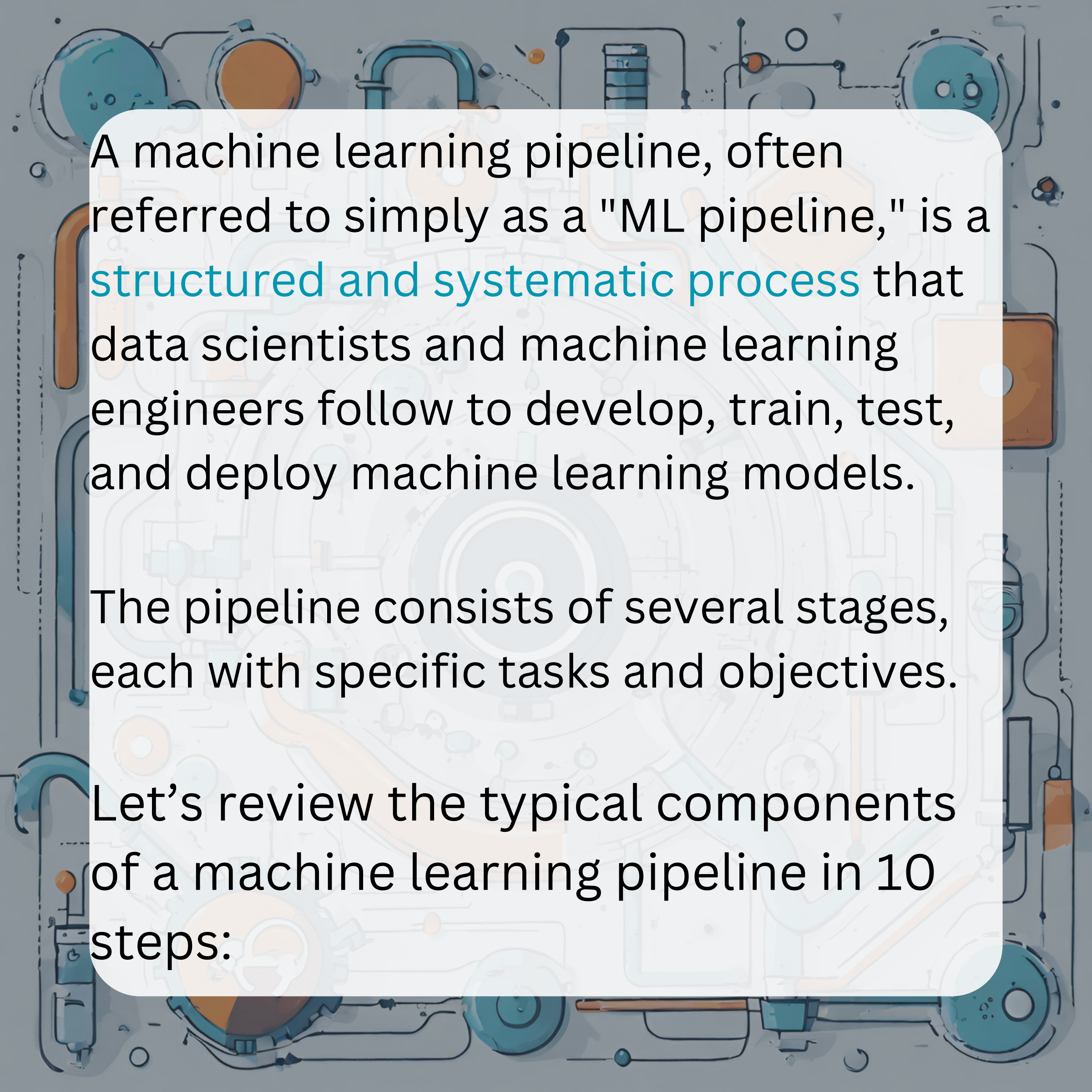


Machine Learning Pipelines

Mohsen Kazemian



A machine learning pipeline, often referred to simply as a "ML pipeline," is a **structured and systematic process** that data scientists and machine learning engineers follow to develop, train, test, and deploy machine learning models.

The pipeline consists of several stages, each with specific tasks and objectives.

Let's review the typical components of a machine learning pipeline in 10 steps:

1.Data Collection and Gathering

Acquire and collect relevant data from various sources, which may include databases, APIs, CSV files, or external datasets.

2.Data Preprocessing

- Clean the data to ensure it is accurate, complete, and consistent.
- Handle missing values through techniques like imputation.
- Encode categorical features into numerical representations.
- Scale or normalize numerical features.
- Perform data exploration and visualization to understand the data.



3.Feature Engineering

- Create new features from existing ones to capture relevant information and improve model performance.
- Select relevant features and eliminate irrelevant ones through feature selection.

4.Data Splitting

- Divide the dataset into three subsets: training data, validation data, and test data.
- The training data is used to train the model.
- The validation data helps in hyperparameter tuning and model selection.
- The test data is used to evaluate the model's performance on unseen data.

5. Model Selection and Training

- Choose an appropriate machine learning algorithm or model architecture based on the problem type (e.g., regression, classification, clustering).
- Train the selected model using the training data.
- Tune hyperparameters to optimize model performance, often using techniques like cross-validation.

6. Model Evaluation

- Assess the model's performance on the validation dataset using relevant evaluation metrics (e.g., accuracy, precision, recall, F1-score, mean squared error).
- Analyze model bias and variance to prevent overfitting or underfitting.

7. Model Testing

- Use the test dataset to evaluate the final model's performance on completely unseen data.
- Ensure that the model generalizes well and performs as expected.

8. Model Deployment

- If the model meets the desired performance criteria, deploy it to a production environment where it can make predictions on new, real-world data.
- Integration with existing systems and infrastructure may be necessary for deployment.

9. Monitoring and Maintenance

- Continuously monitor the deployed model's performance in production.
- Retrain the model periodically to keep it up-to-date with changing data distributions or requirements.
- Address any issues or drift in model performance as they arise.

10.Documentation and Reporting

- Maintain comprehensive documentation throughout the pipeline, including data preprocessing steps, model architecture, hyperparameters, and evaluation results.
- Communicate findings and results through reports or presentations.