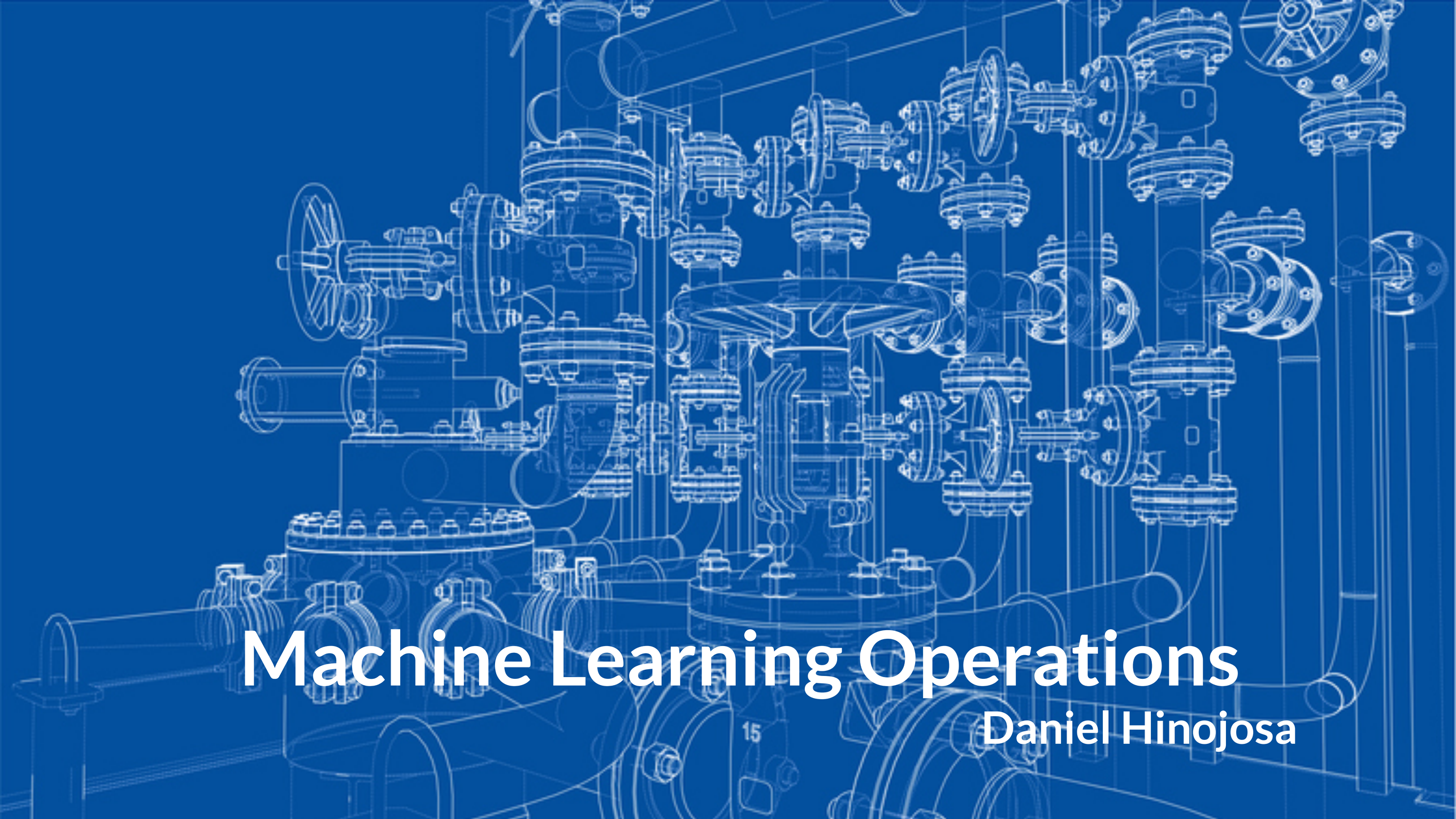


Machine Learning Operations

Daniel Hinojosa


NFJS

Applications

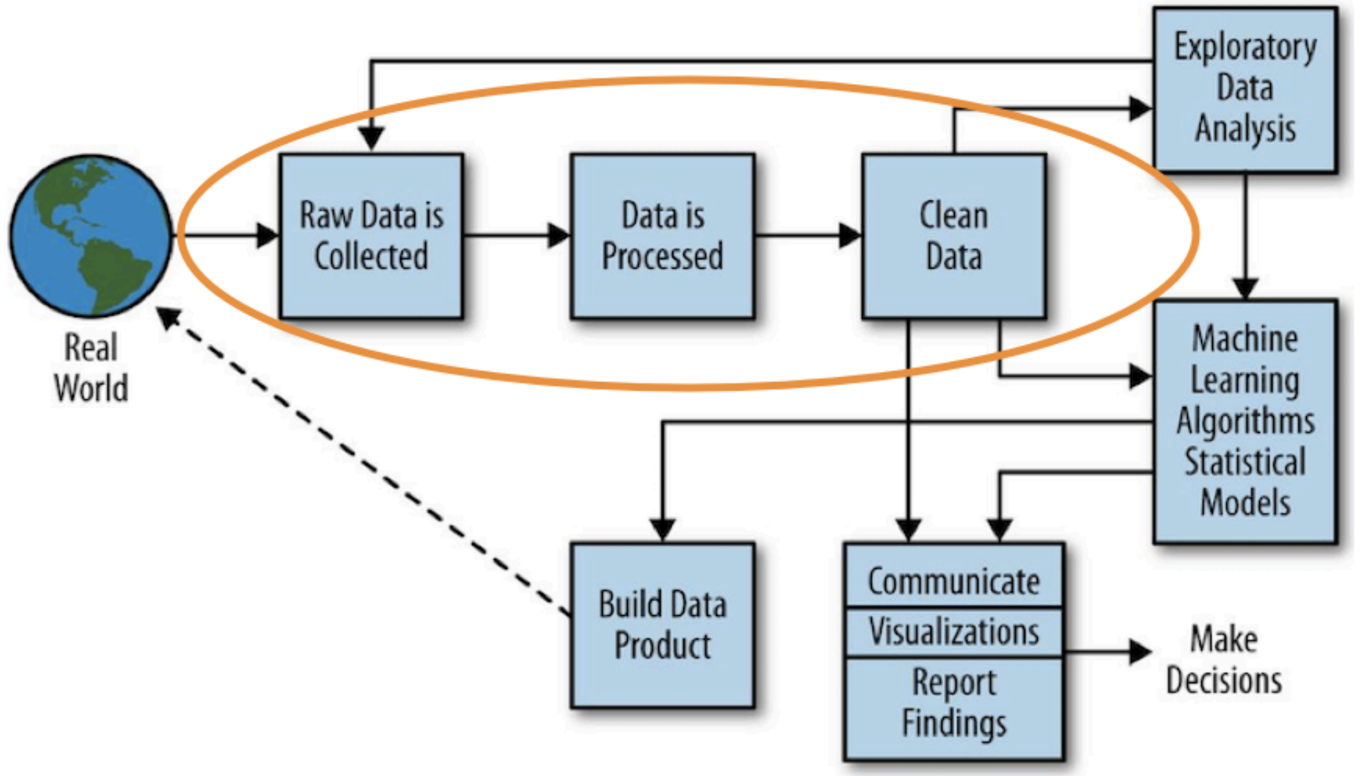
Fraud Detection

Financial Approvals

Picture Recognition

Your Idea Here

Process of Machine Learning


Exploratory
Data
Analysis
Raw Data is
Collected
Data is
Processed
Clean
Data
Real
World
Machine
Learning
Algorithms
Statistical
Models
Build Data
Product
Communicate
Visualizations
Report
Findings
Make
Decisions

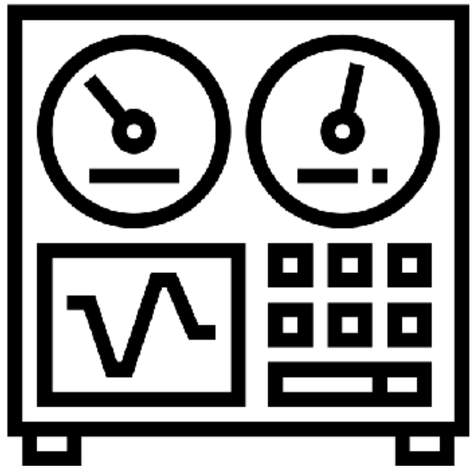
training

0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1

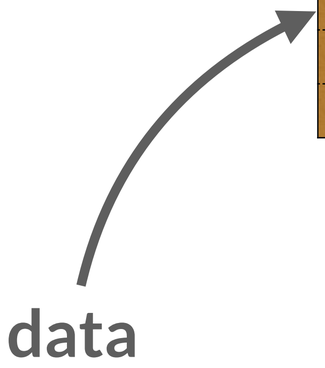
testing

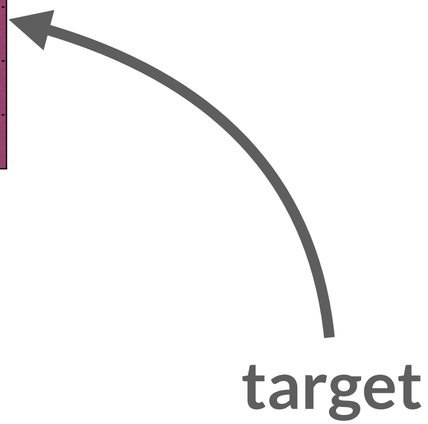
10	54	1	239	1
11	48	0	275	0
12	49	1	266	1
13	64	1	211	0
14	58	0	283	1

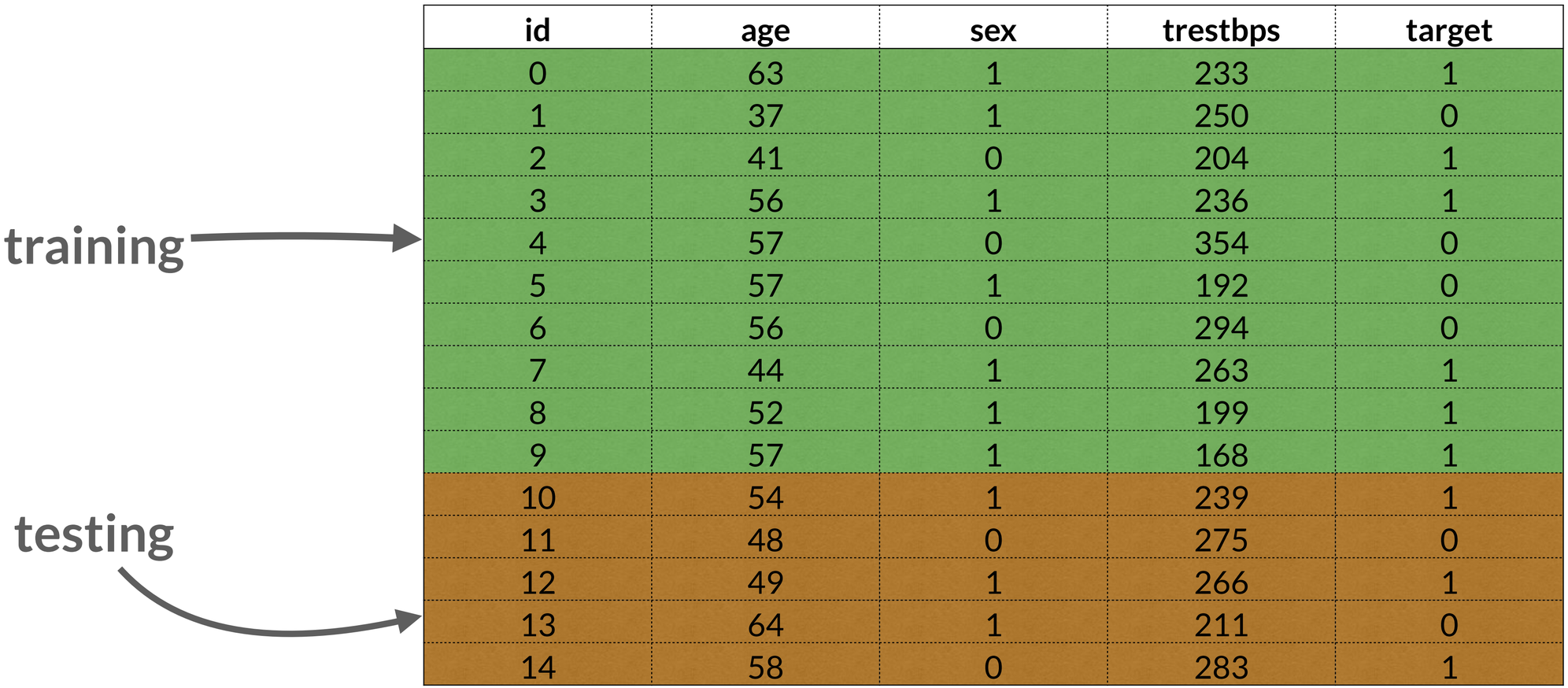
training phase



id	age	sex	trestbps	risk
0	20	1	145	1
1	45	1	130	1
2	43	0	130	0
3	33	1	150	0







id	age	sex	trestbps	target
0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1
10	54	1	239	1
11	48	0	275	0
12	49	1	266	1
13	64	1	211	0
14	58	0	283	1

id	age	sex	trestbps	target
0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1
10	54	1	239	1
11	48	0	275	0
12	49	1	266	1
13	64	1	211	0
14	58	0	283	1

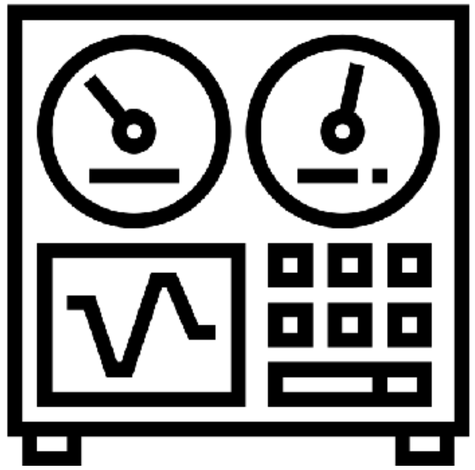
random distribution between training and testing

training

0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1

testing

10	54	1	239	1
11	48	0	275	0
12	49	1	266	1
13	64	1	211	0
14	58	0	283	1



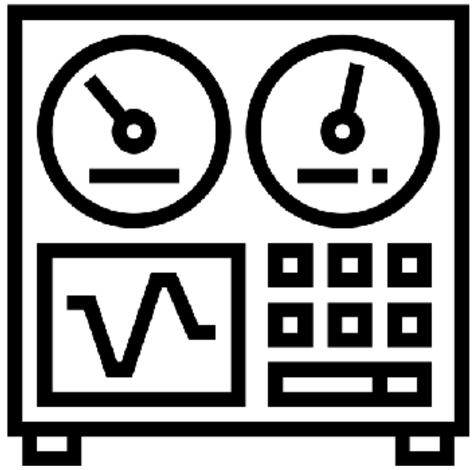
training

0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1

testing

10	54	1	239	1
11	48	0	275	0
12	49	1	266	1
13	64	1	211	0
14	58	0	283	1

model



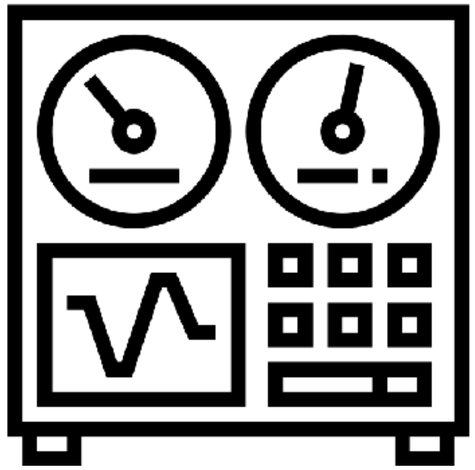
training

0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1

testing

10	54	1	239	1
11	48	0	275	0
12	49	1	266	1
13	64	1	211	0
14	58	0	283	1

training phase

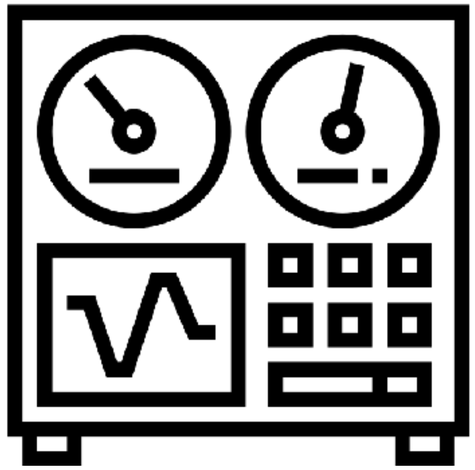


training

0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1

testing

10	54	1	239	1
11	48	0	275	0
12	49	1	266	1
13	64	1	211	0
14	58	0	283	1



training

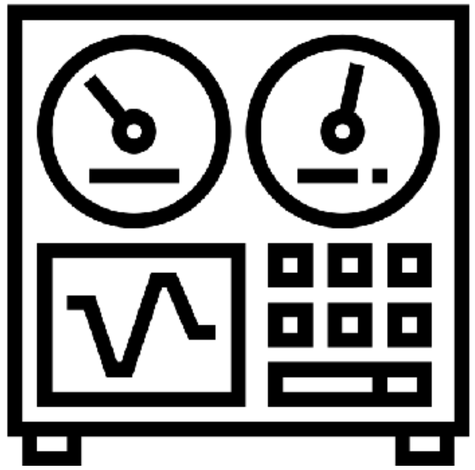
0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1

testing

10	54	1	239
11	48	0	275
12	49	1	266
13	64	1	211
14	58	0	283

actual

1
0
1
0
1



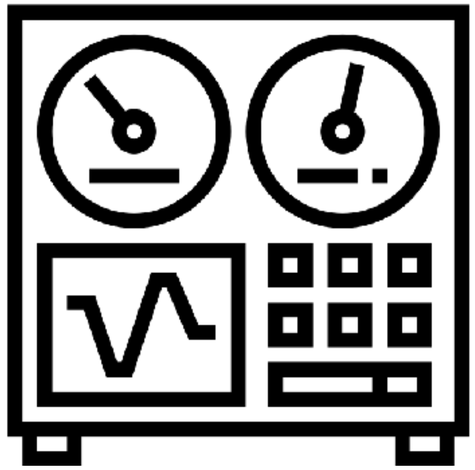
reserve the actual later for verification

training

0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1

testing

10	54	1	239
11	48	0	275
12	49	1	266
13	64	1	211
14	58	0	283



testing phase

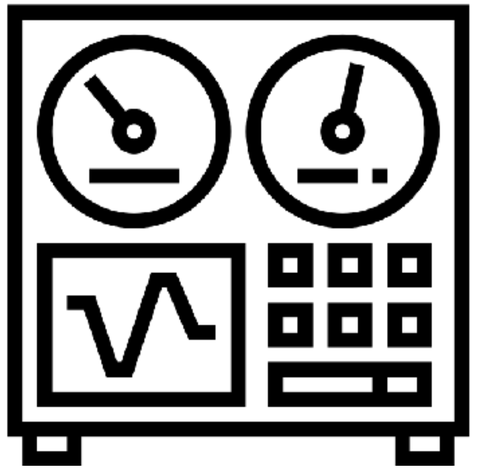
training

0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1

testing

10	54	1	239
11	48	0	275
12	49	1	266
13	64	1	211
14	58	0	283

!!!!!!!



generated result

1
0
1
1
1

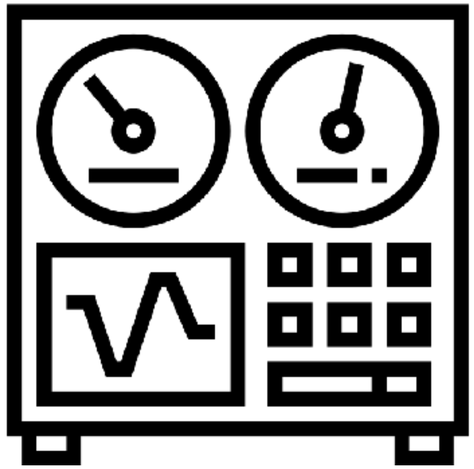
training

0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1

testing

10	54	1	239
11	48	0	275
12	49	1	266
13	64	1	211
14	58	0	283

!!!!!!!



generated result

1
0
1
1
1

actual

1
0
1
0
1

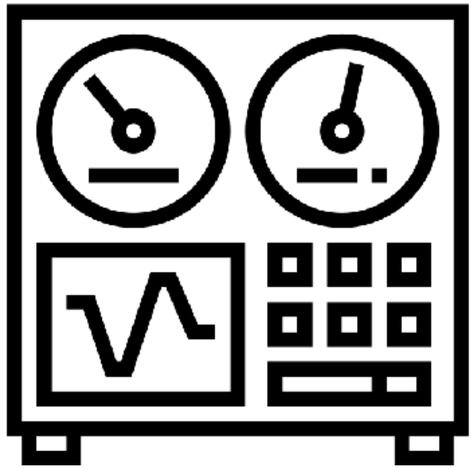
training

0	63	1	233	1
1	37	1	250	0
2	41	0	204	1
3	56	1	236	1
4	57	0	354	0
5	57	1	192	0
6	56	0	294	0
7	44	1	263	1
8	52	1	199	1
9	57	1	168	1

testing

10	54	1	239
11	48	0	275
12	49	1	266
13	64	1	211
14	58	0	283

!!!!!!!



generated result

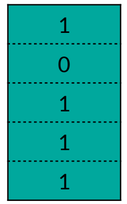
1
0
1
1
1

actual

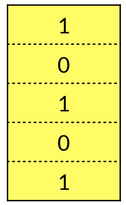
1
0
1
0
1

How did we do?

generated result



actual



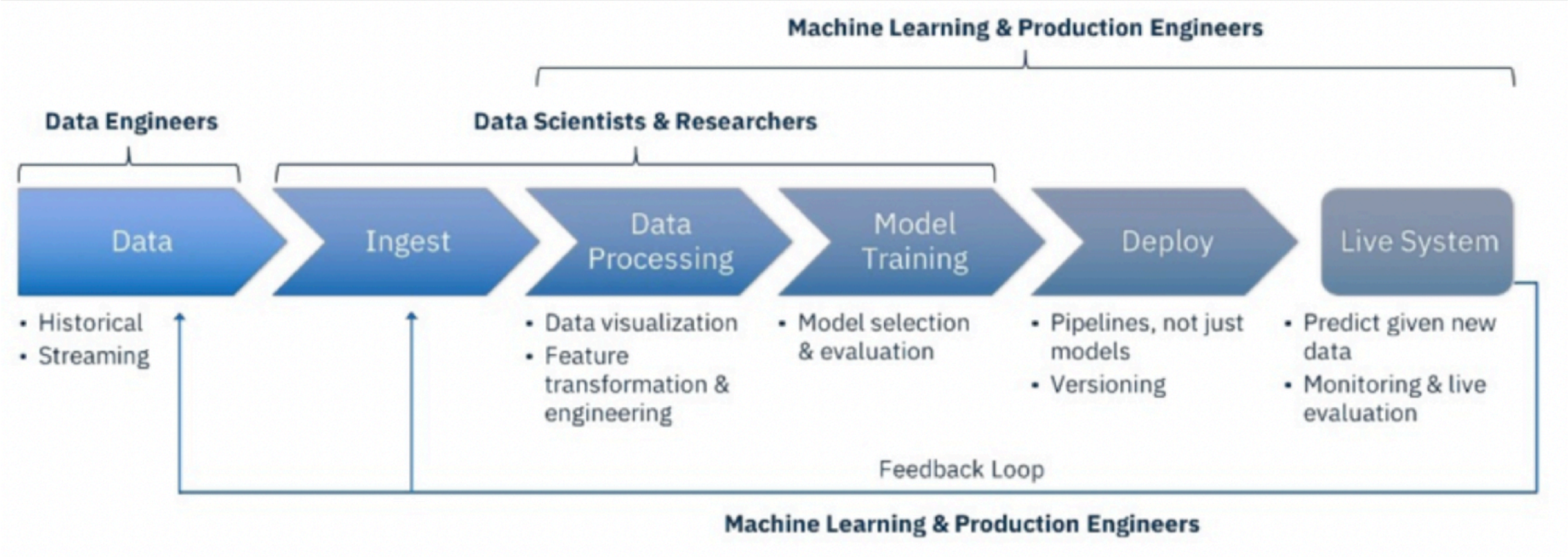
How did we do?

Machine Learning Models

Machine Learning	Deep Learning
KNN	Convolutional Neural Networks
Naive Bayes	Recurrent Neural Networks
Linear Regression	Autoencoders

Roles and Responsibilities

Roles and Responsibilities



What is MLOps?

MLOps (a compound of “machine learning” and “operations”) is a practice for collaboration and communication between data scientists and operations professionals to help manage production ML lifecycles.

Why do we need MLOps?


Java™

Data Scientists are not Software Engineers

- They are focused on and specialized in model building and assessment
- They can become specialized on operationalization and deployment
- Data Scientists can possibly be stretched too thin,

Many Dependencies

- Business Needs Change
- Results need to be relayed back to the business
- We have to ensure the reality of the model

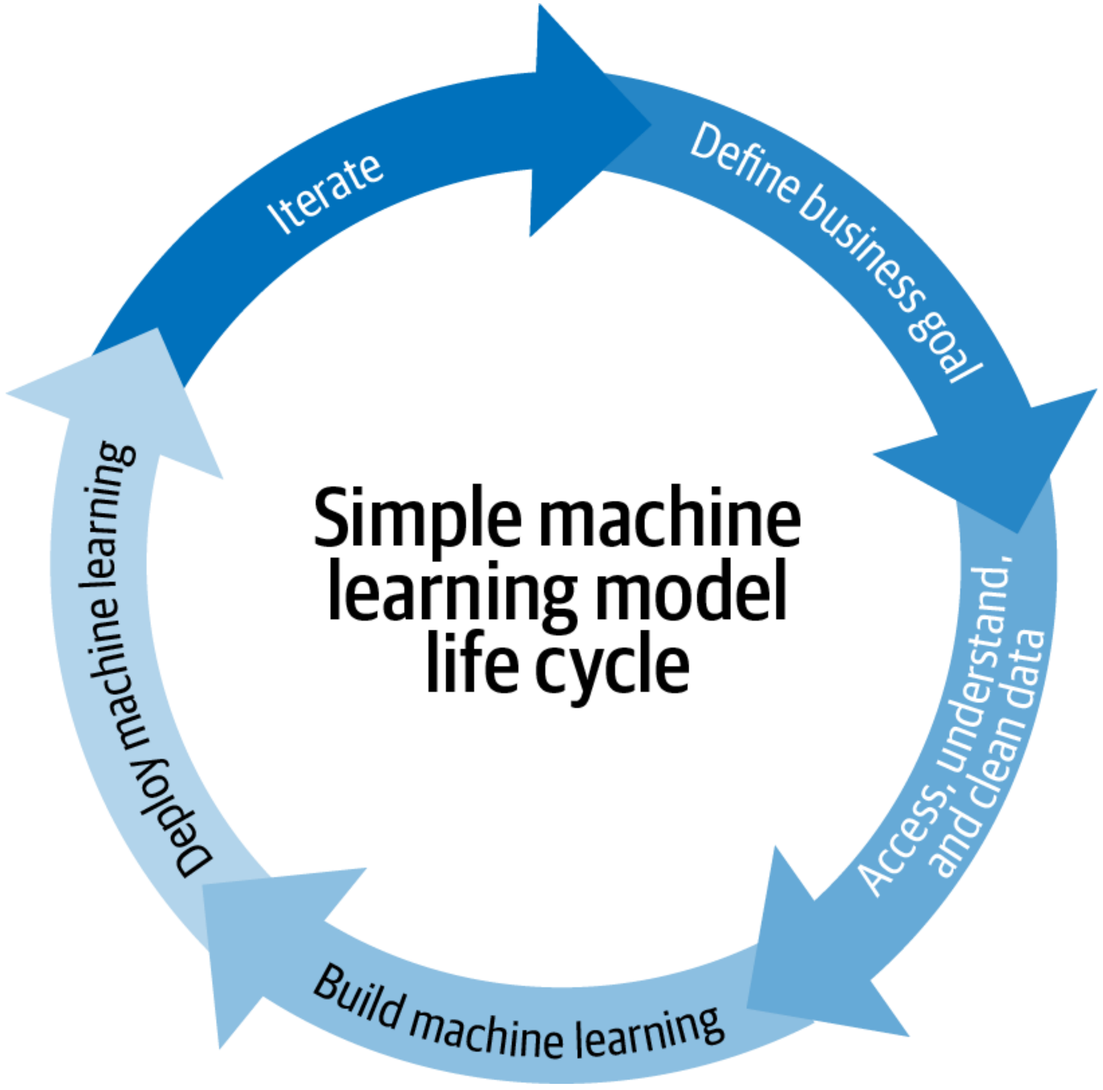
What is involved with MLOps ?

MLOps is similar to DevOps

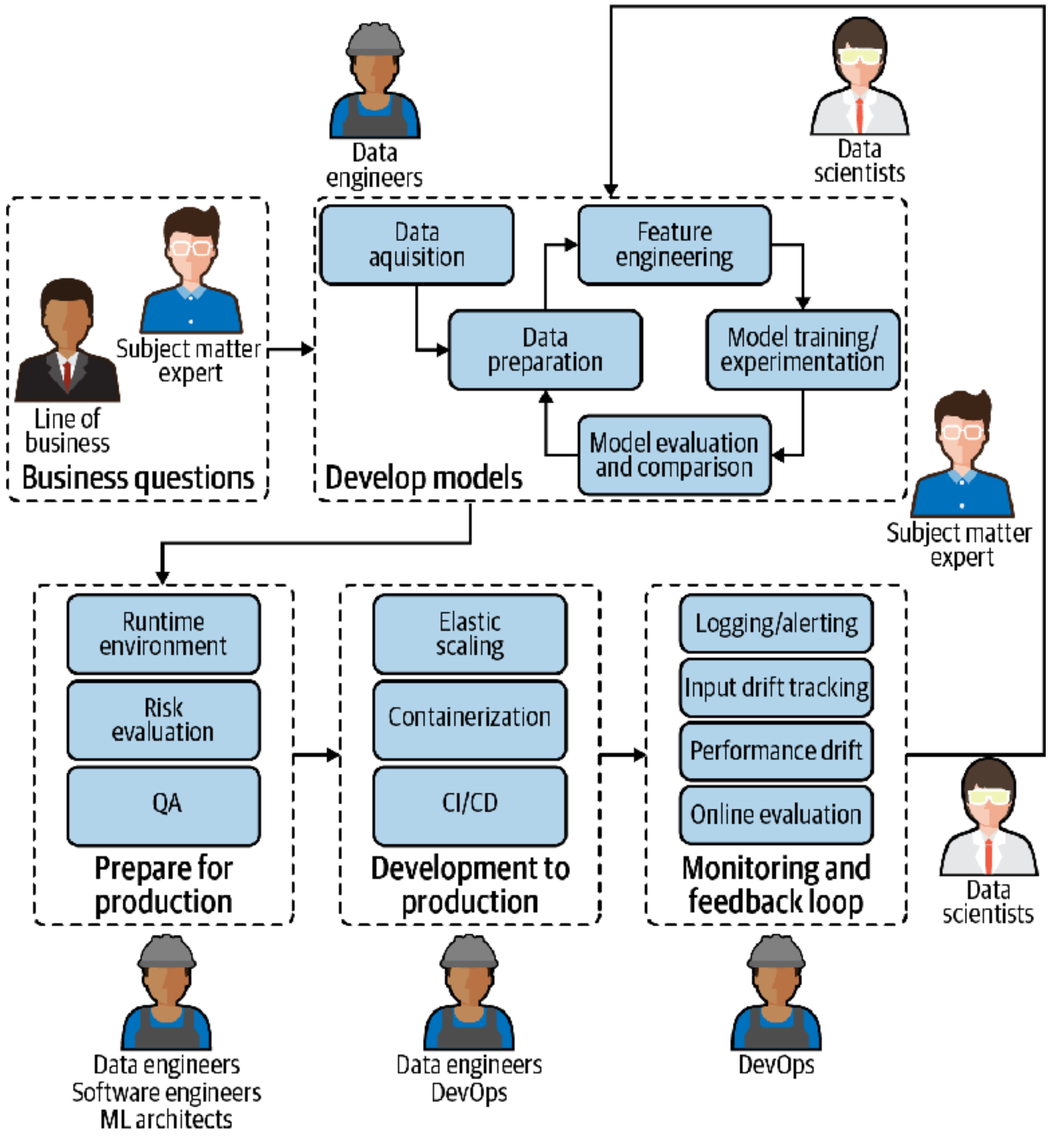
- Robust automation and trust between teams
- The idea of collaboration and increased communication between teams
- The end-to-end service life cycle (build, test, release)
- Prioritizing continuous delivery and high quality

MLOps is different to DevOps

- Deploying software code into production is fundamentally different than deploying machine learning models into production
- Data is always changing, which means machine learning models are constantly learning and adapting
- Machine learning models are made up of both code and data


Simple machine learning model life cycle
Define business goal
Access, understand, and clean data
Build machine learning
Deploy machine learning
Iterate

MLOps Procedures Checklist

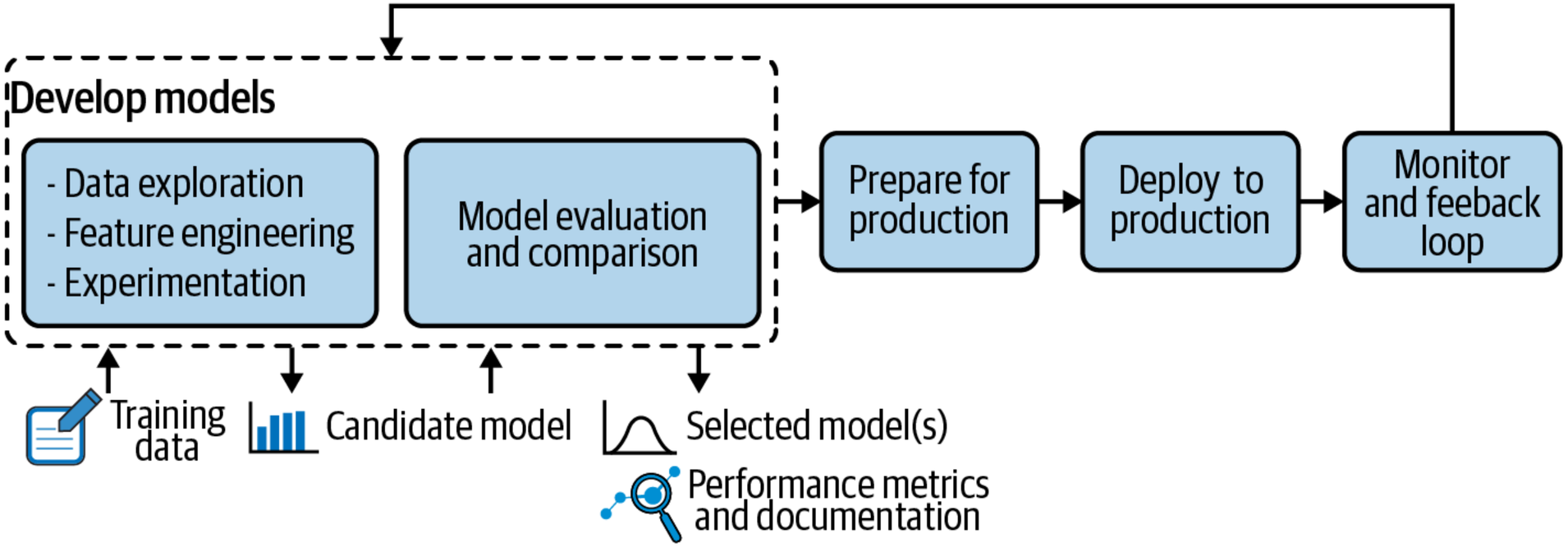


Source: Introducing MLOps

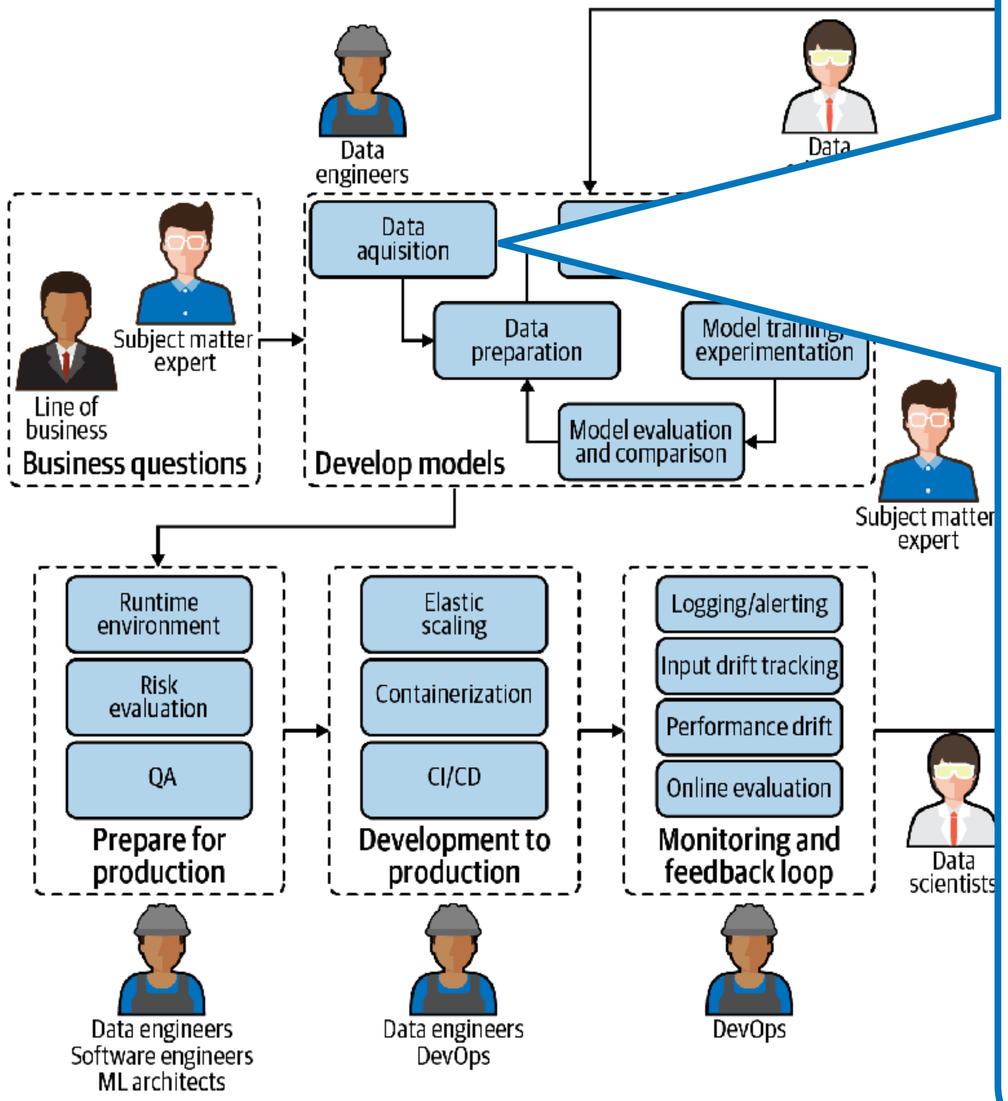
Mark Treveil, Nicolas Omont, Clément Stenac, Kenji Lefevre, Du Phan, Joachim Zentici, Adrien Lavoillotte, Makoto Miyazaki, Lynn Heidmann

Published by O'Reilly Media, Inc.

Develop Models

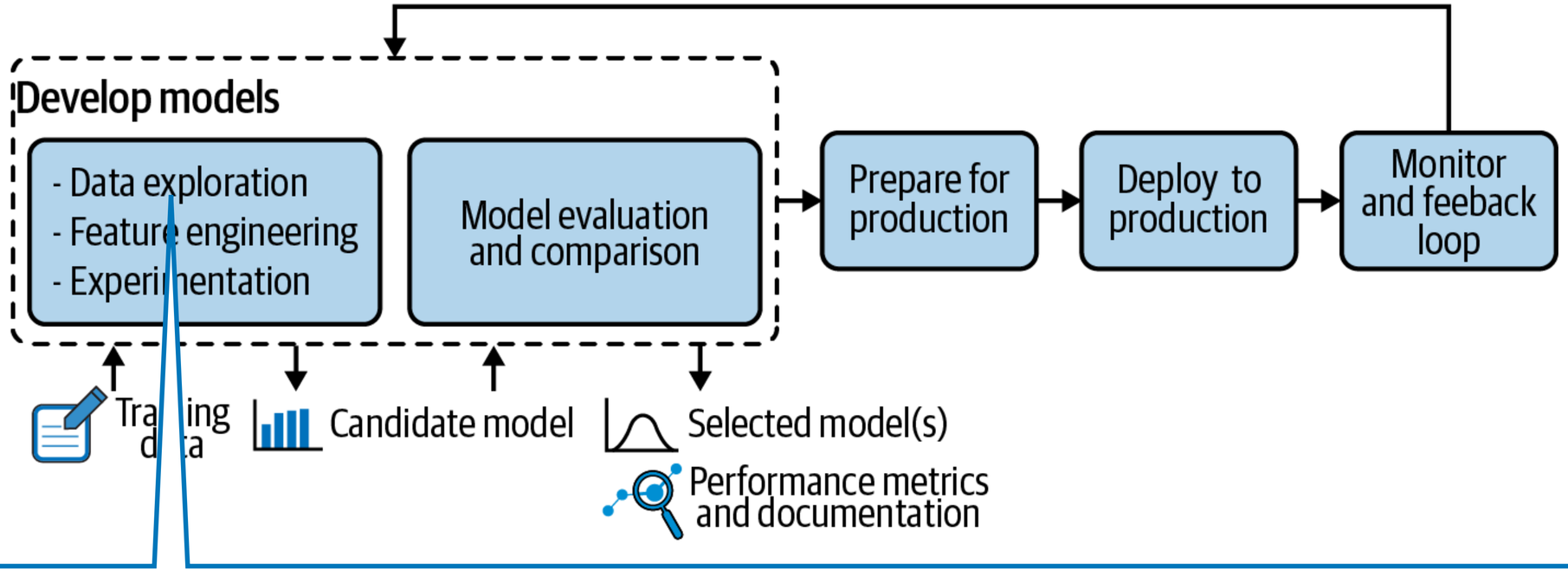


Source: Introducing MLOps
Mark Treveil, Nicolas Omont, Clément Stenac, Kenji Lefevre, Du Phan, Joachim Zentici, Adrien Lavoillotte, Makoto Miyazaki, Lynn Heidmann
Published by O'Reilly Media, Inc.



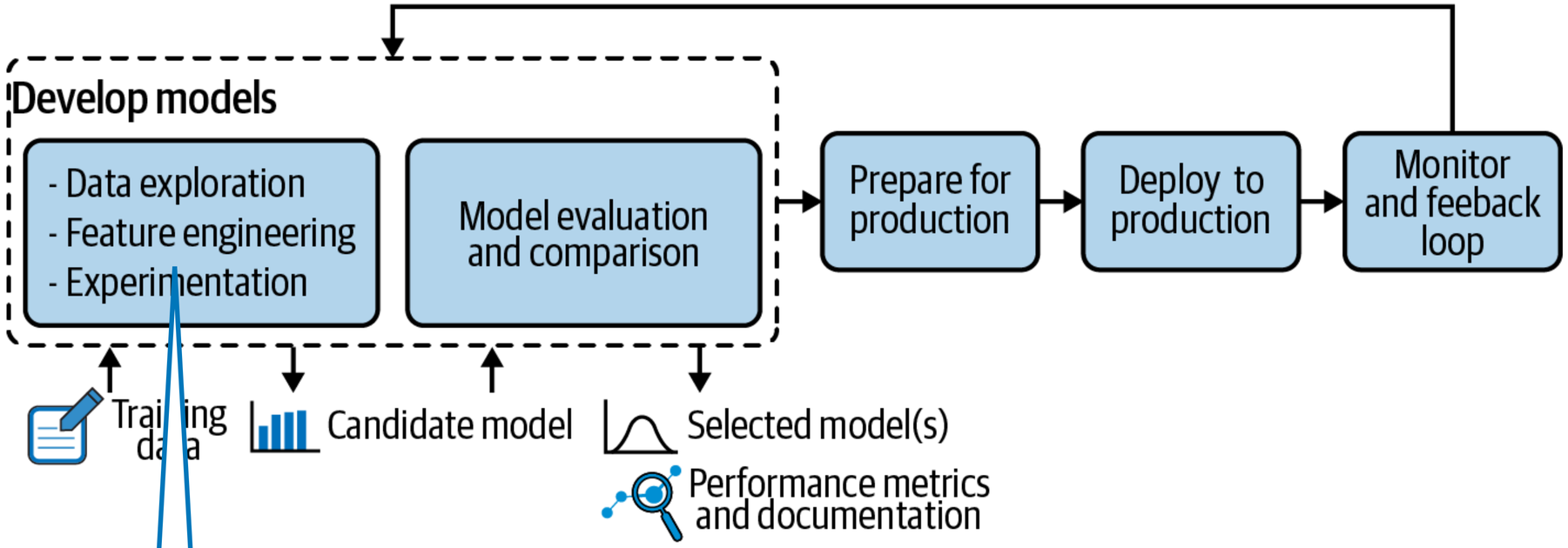
Data Acquisition

- **Data Collection in MLOps is the process of gathering and preparing raw data from various sources to be used in machine learning workflows**
  - **Data Sources:** Data can come from a variety of sources, such as databases, APIs, sensors, user interactions, or third-party providers.
  - **Automation:** MLOps pipelines often automate the collection process, ensuring timely and consistent retrieval of data for model training and retraining.
  - **Data Quality:** Ensures that the collected data is clean, complete, and accurate, with checks for missing or incorrect values before it is used in model development.
  - **Storage and Management:** Collected data is stored in data lakes, warehouses, or databases and is organized for easy access and versioning.
  - **Compliance:** Data collection follows legal and ethical guidelines, ensuring adherence to privacy regulations like GDPR and CCPA.



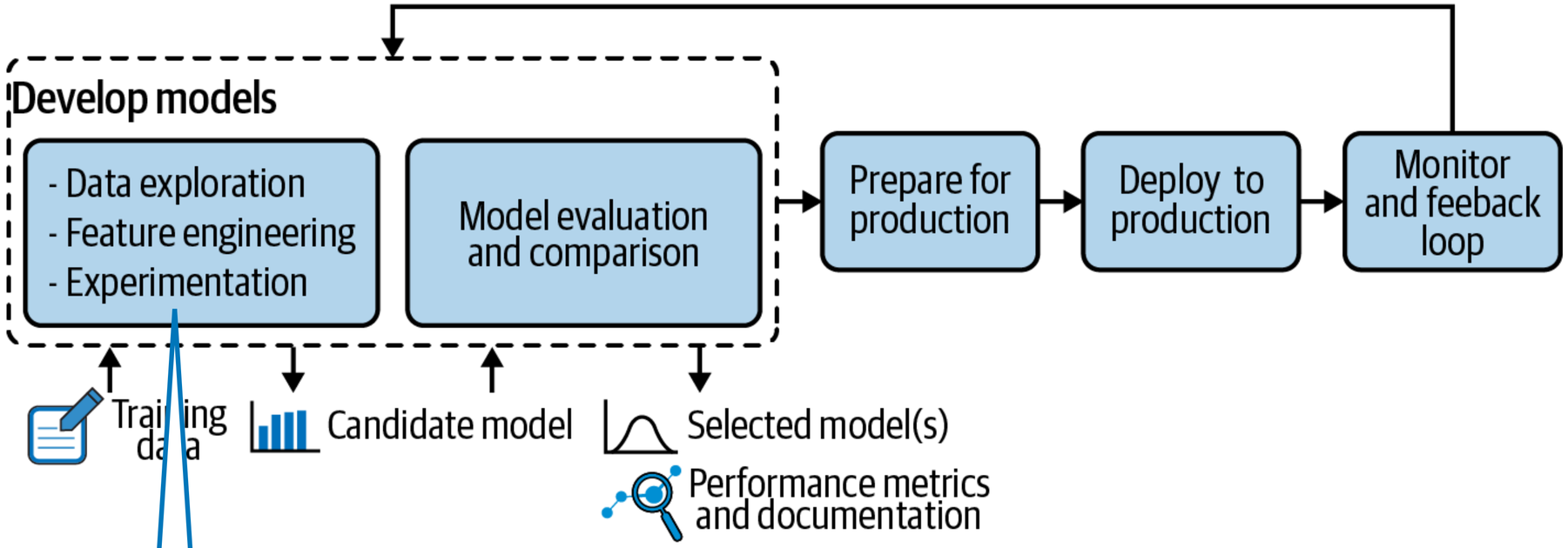
Data Exploration

- Data Scientists consider where the information came and evaluate inconsistencies
- What are some mistakes?
- Look at the distribution
- Compare data with other models, are there items missing?



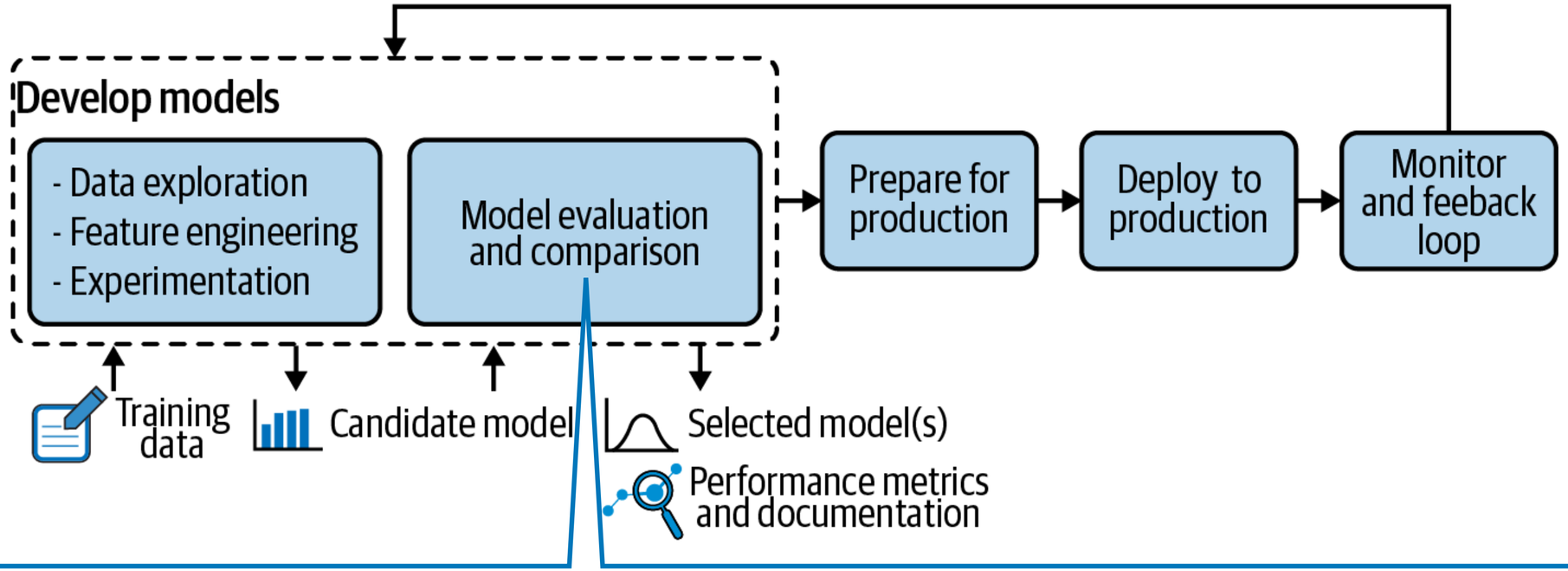
Feature Engineering

- Can we see into the columns and extract more information?
  - Day of the Week
  - Public Holiday
  - Weekday or Weekend
  - Link Features Together



Experimentation

- Assessing how useful or how good of a model can be built given the elements
- Finding the best modeling parameters (algorithms, hyperparameters, feature preprocessing, etc.).
- Tuning the bias/variance trade-off for a given training cost to fit that definition of “best.”

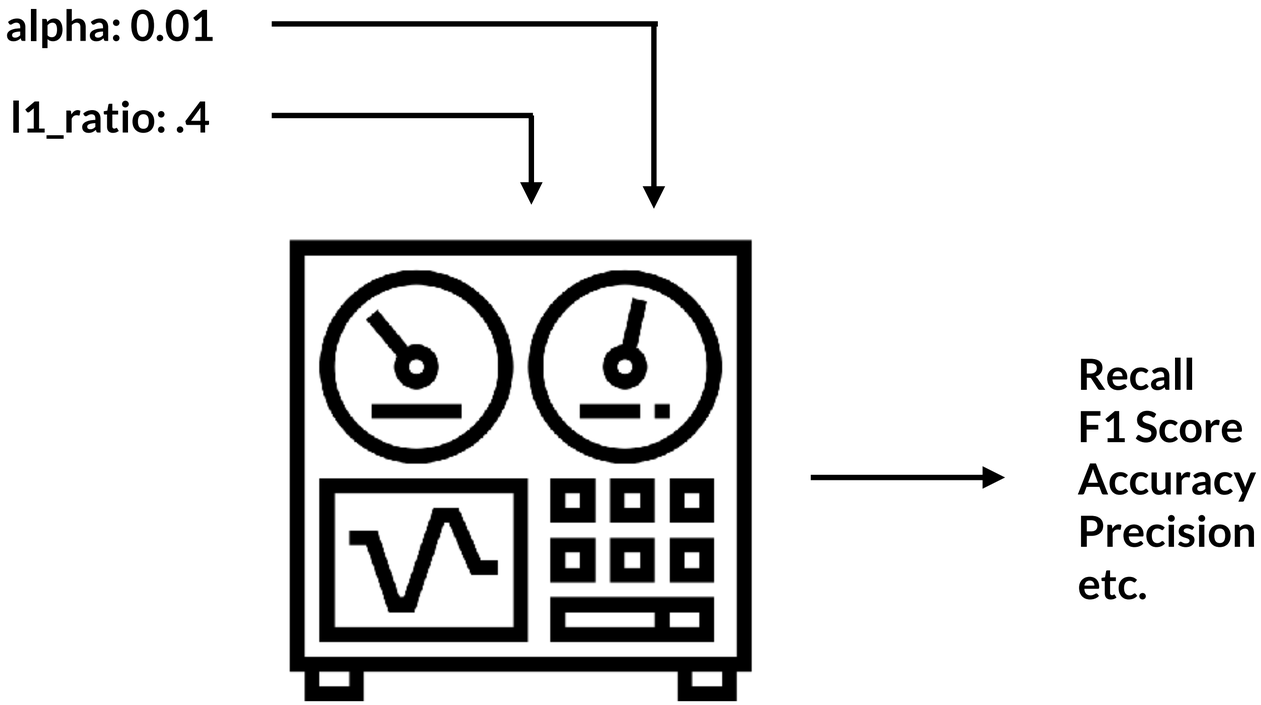


Model Evaluation and Comparison

- Select a Model based on data, what shape it holds, can it be simplified (logistic, linear)
- Choose a metric that fits well for your problem
- Compare model scores using cross-validation techniques

Hyperparameter Tuning

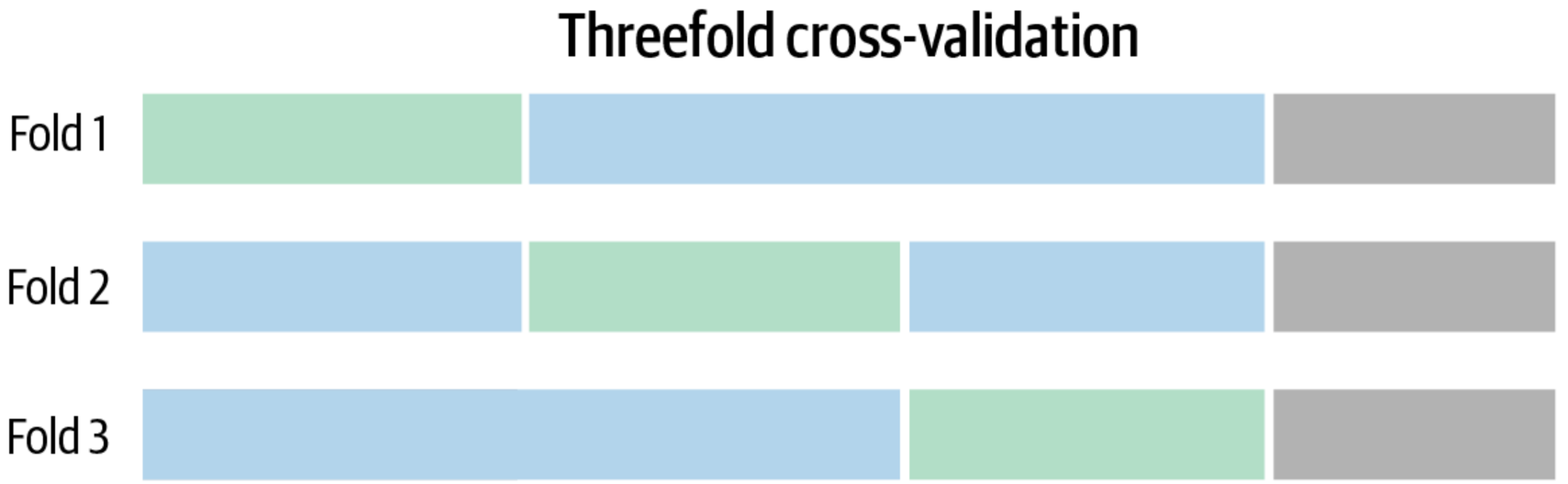
Hyperparameters are adjustable parameters that let you control the model training process.



Cross Validation

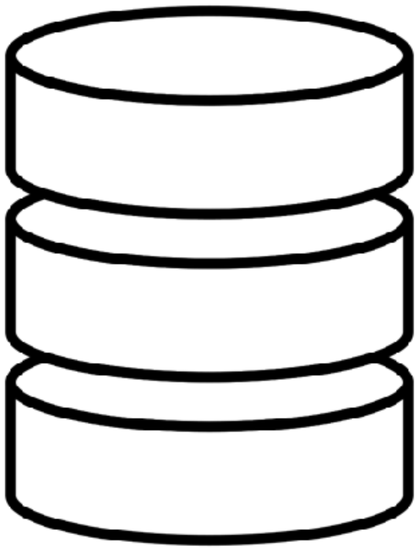
- The validation set is separate from the training and test sets and is used during the model training process to tune hyperparameters
- There are techniques like k-fold cross-validation where you can shuffle data, split the data into groups
- Each group and either select as a hold out or test data set, take the remainder and use if for training.
- This iterative process is how we can get close to the score that we need

Cross Validation



- Grey is the holdout for Test Data
- Remaining Data is split into three to find the best combination
- Green is used for validation
- Blue is for training

Model Registry



- A Model Registry in MLOps is a centralized repository where machine learning models are stored, versioned, and managed throughout their lifecycle. Key aspects include:
  - **Versioning:** Tracks different versions of models, including changes in parameters, data, and code.
  - **Stage Transitions:** Defines stages such as development, staging, and production, allowing models to move through testing and deployment phases systematically.
  - **Metadata:** Stores important information like metrics, training datasets, and environment configurations for each model.
  - **Collaboration:** Enables teams to collaborate by sharing and reviewing models across the organization.
  - **Governance:** Ensures proper auditing, tracking, and control of model versions in production environments.

Model Versioning

- Model Versioning in MLOps is the process of tracking and managing different versions of machine learning models throughout their lifecycle. Key features include:
  - **Version Control:** Assigns unique identifiers to models, allowing teams to track and reference specific versions.
  - **Reproducibility:** Ensures that every version of a model can be reproduced with the exact same data, code, and parameters used during training.
  - **Experiment Comparison:** Enables comparisons between different versions of models to assess performance improvements or changes.
  - **Deployment Management:** Facilitates the deployment of specific model versions to different environments (e.g., testing, staging, production).
  - **Rollback:** Allows easy rollback to previous model versions in case of issues or performance degradation.

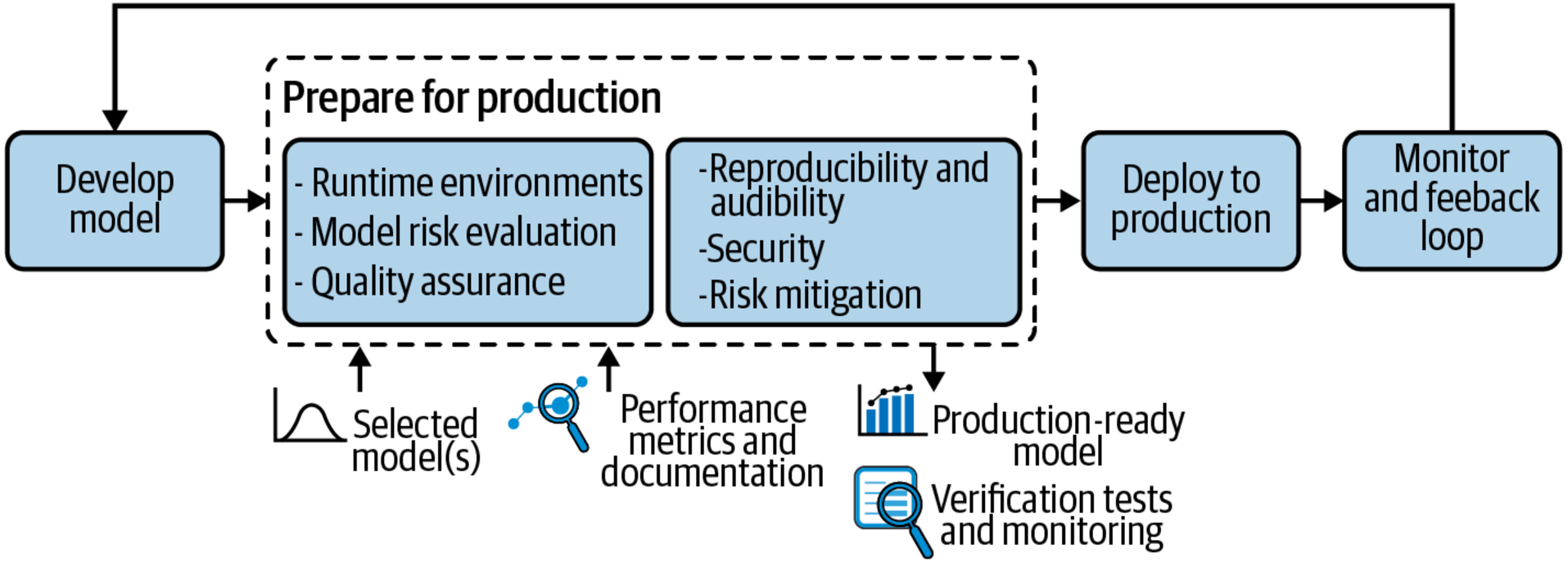
Model Versioning Strategies

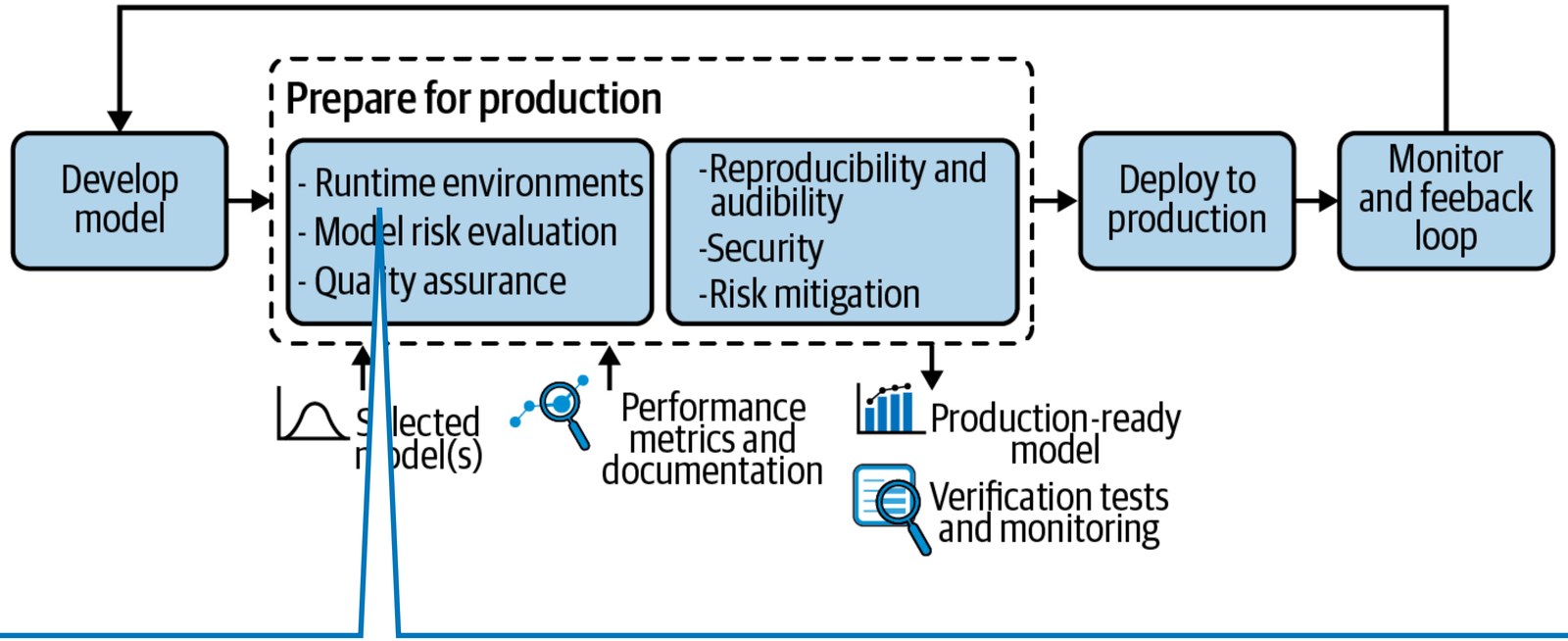
- Model Versioning can be done in a variety of ways:
  - Based on your Model Registry
  - Source Control
  - Containerization Versioning: 1.0 or SHA
  - File-Based Versioning in Object Storage Systems
  - Semantic Versioning

Data Cataloging

- Data Cataloging in MLOps is the process of organizing, indexing, and managing metadata about datasets used in machine learning workflows. Key aspects include:
  - **Centralized Repository:** Provides a single source of truth for all datasets, making it easy to locate and access data across teams.
  - **Metadata Management:** Stores detailed information about datasets, such as schema, source, quality metrics, lineage, and usage history.
  - **Searchability:** Enables users to quickly search and discover datasets based on keywords, tags, or filters like date, owner, or type.
  - **Data Governance:** Supports data compliance by tracking how datasets are used and by whom, ensuring adherence to regulations and policies.
  - **Collaboration:** Facilitates collaboration by making data assets discoverable and reusable across teams, improving consistency and productivity.

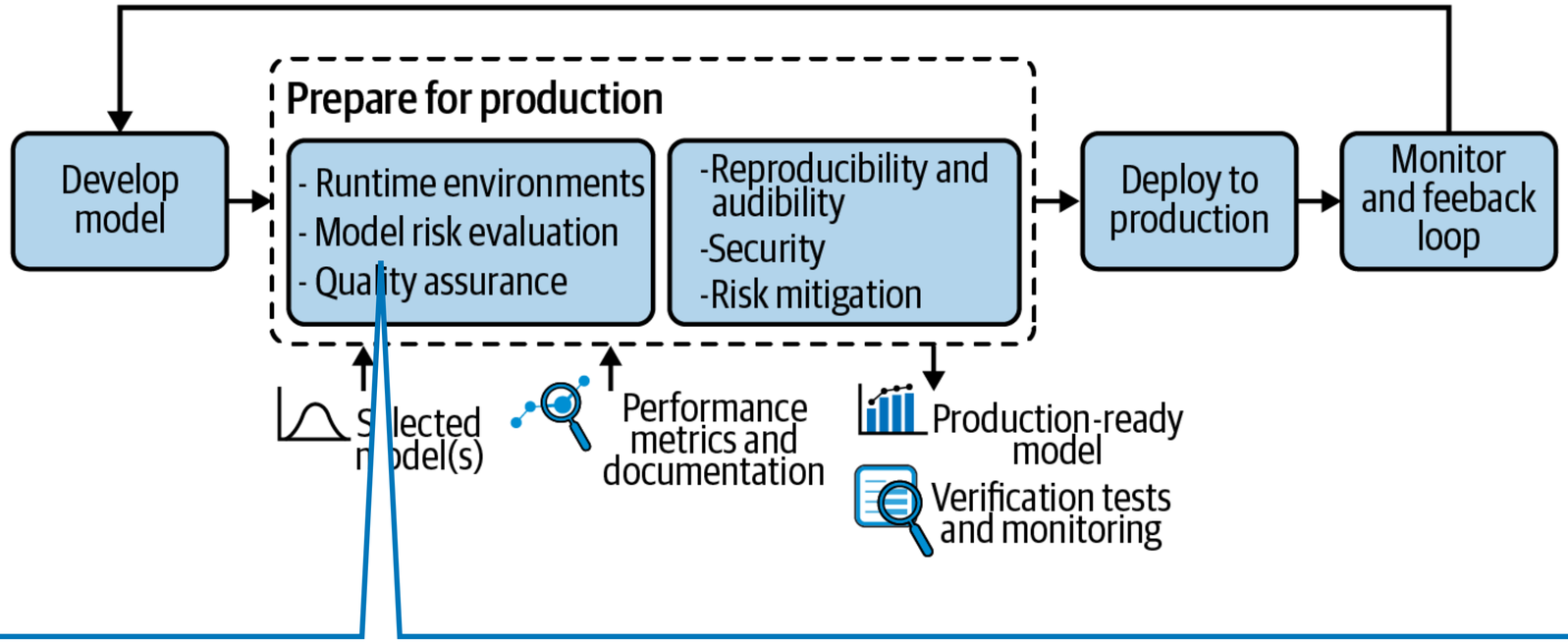
Preparing for Production


Prepare for production
Develop model
- Runtime environments
- Model risk evaluation
- Quality assurance
-Reproducibility and audibility
-Security
-Risk mitigation
Deploy to production
Monitor and feeback loop
Selected model(s)
Performance metrics and documentation
Production-ready model
Verification tests and monitoring



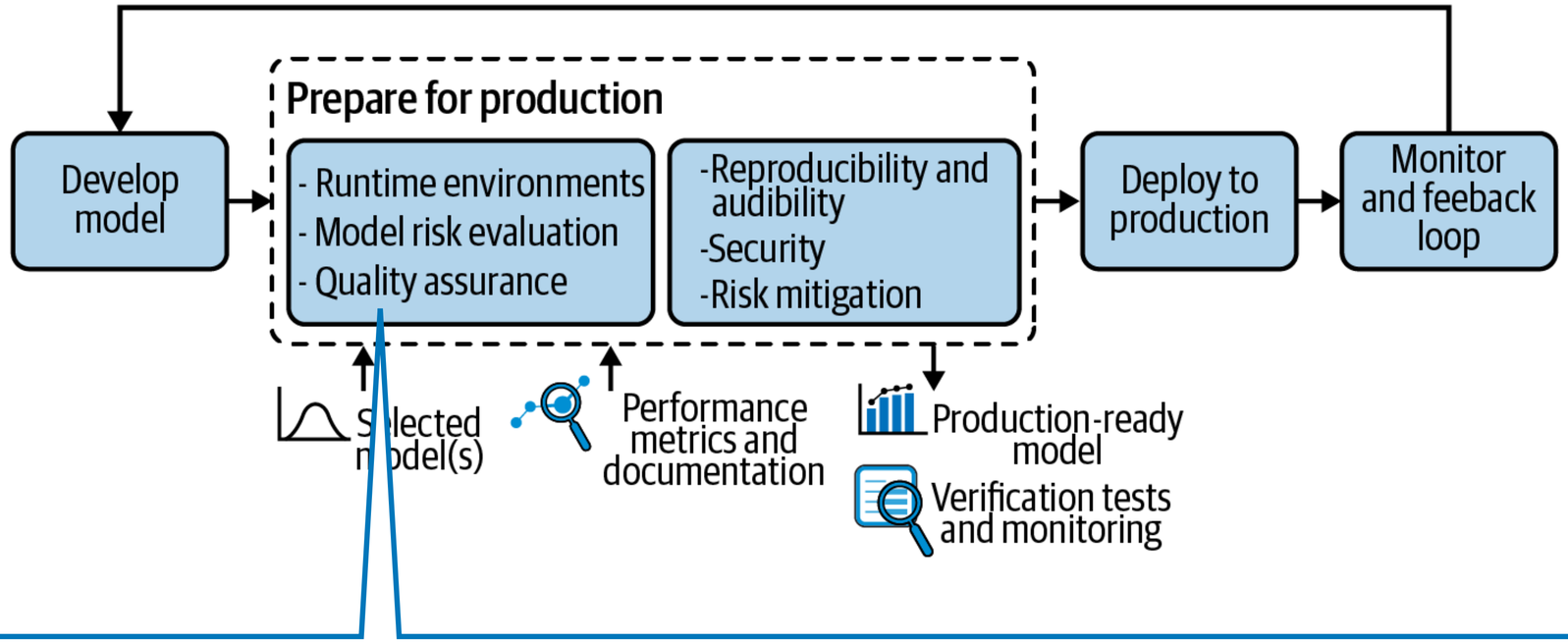
Runtime Environments

- Is it technically possible to run? There have been huge efforts and costs only to find out that they cannot deploy their model
- What tools will you use?
- What will data access be like to continually feed your model?



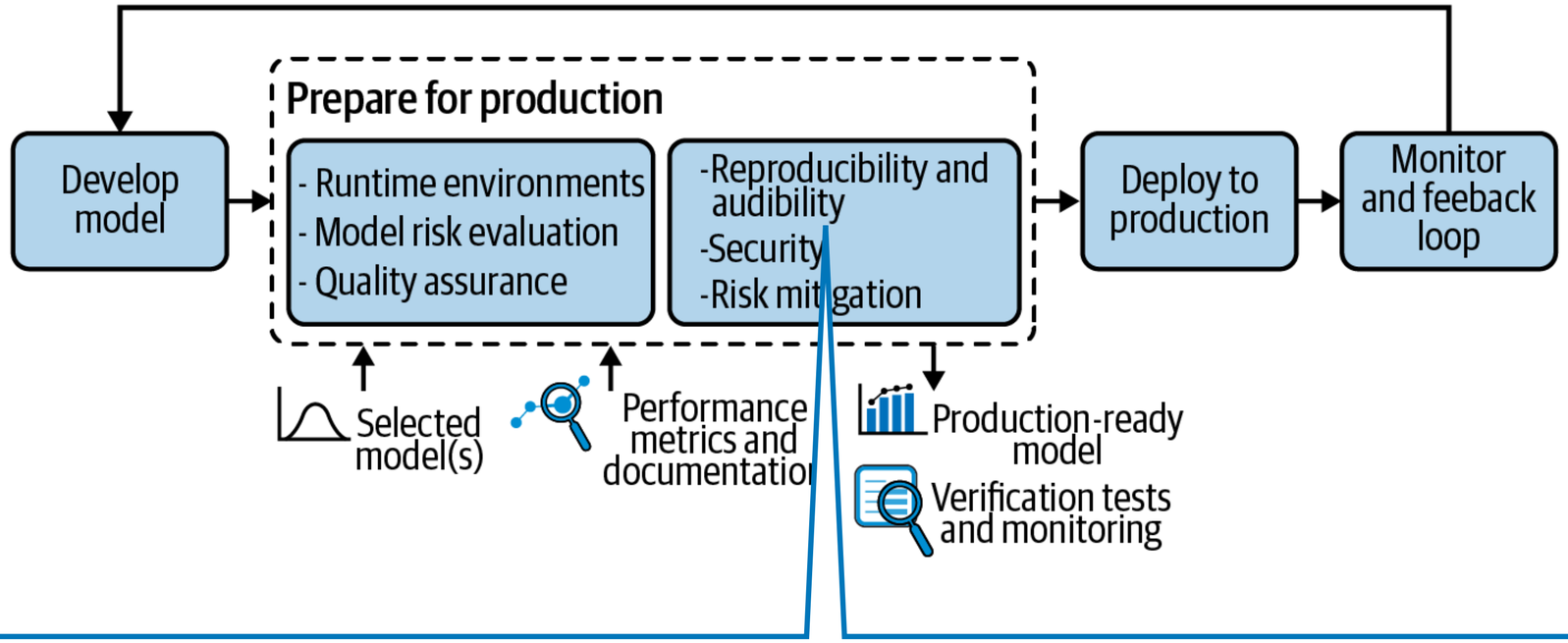
Model Risk Evaluation

- What if the model works in the worst possible way?
- What are security concerns of running this model?
- What if the model is exposed?
- What are financial, business, legal, and other risks by proceeding?



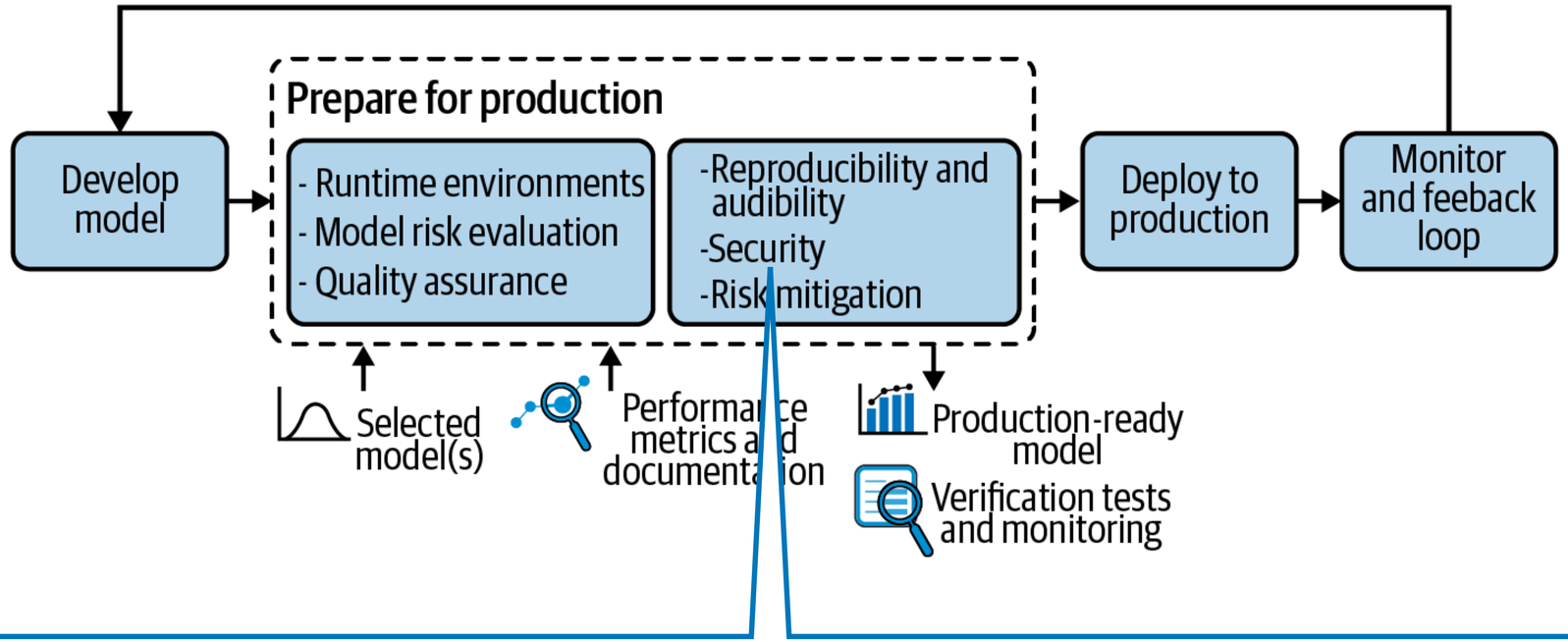
Quality Assurance

- Create documentation and validate the model against organizational guidelines
- Origin of all input datasets, pretrained models, or other assets should be known, as they could be subject to regulations or copyrights



Reproducibility and Audibility

- Reproducibility in MLOps also involves the ability to easily rerun the exact same experiment
- For a model to be auditable, it must be possible to access the full history of the ML pipeline from a central and reliable storage and to easily fetch metadata on all model versions

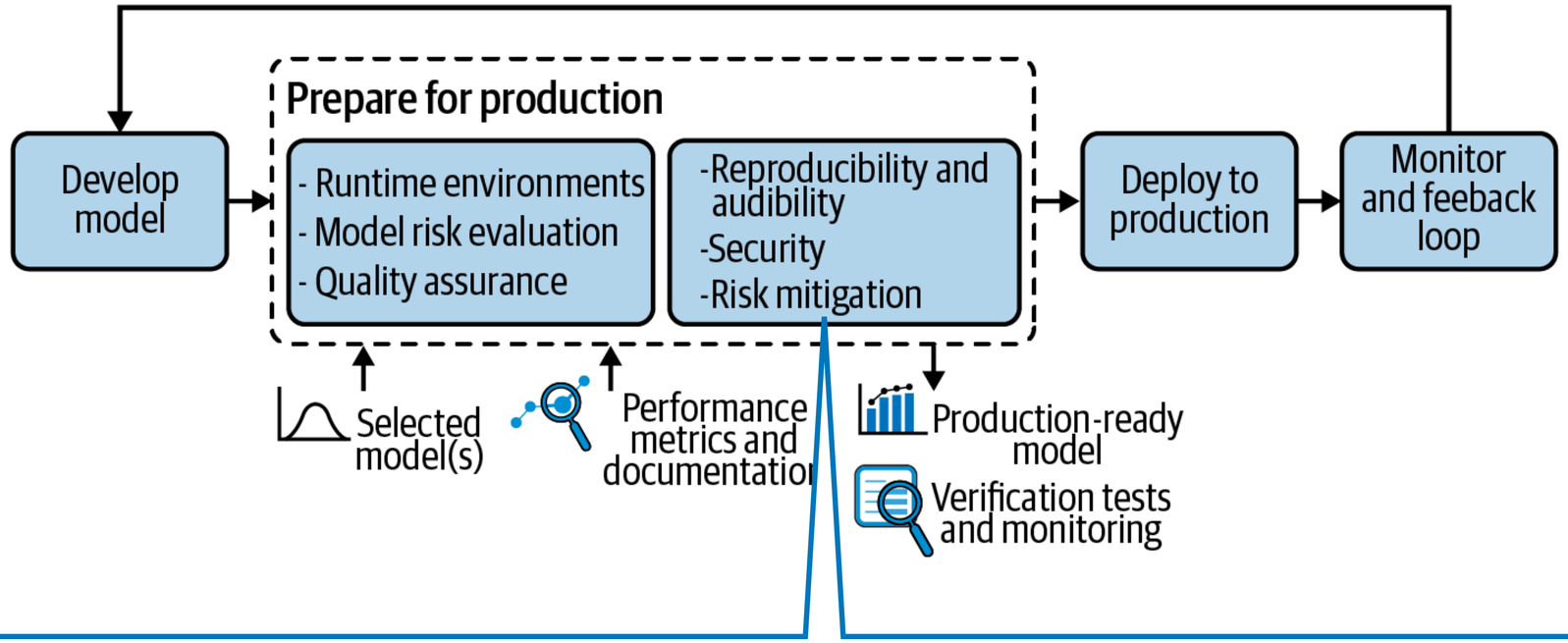


Security

- Perform a threat model of what can happen with things go wrong with your model
- Logistic Regression (S-Curve) models are immune to many attacks
- Neural Networks are expose due to the ability to corrupt the inputs

Researchers hack a self-driving car by putting stickers on street signs

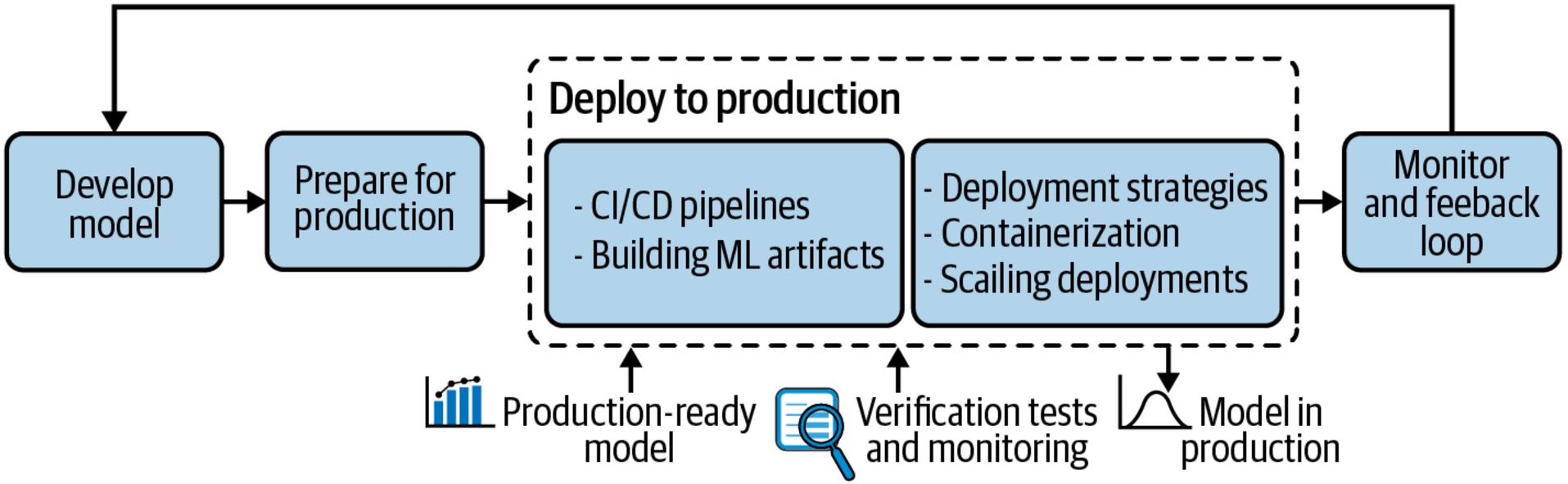
https://www.autoblog.com/news/self-driving-car-sign-hack-stickers

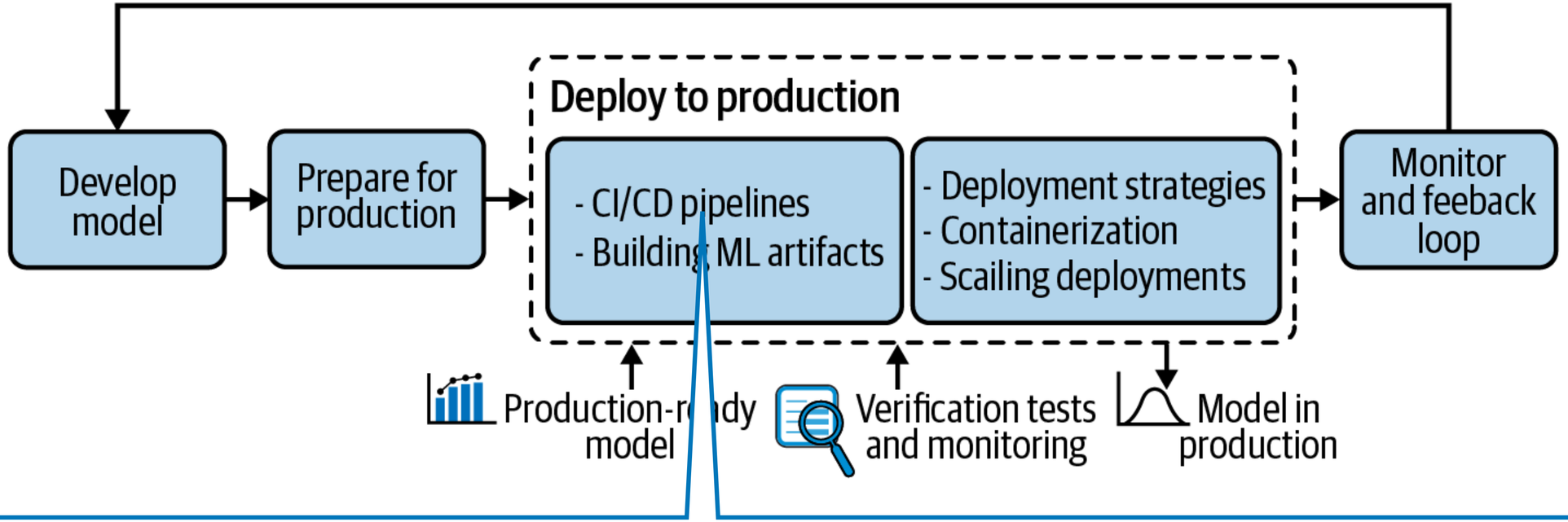


Risk Mitigation

- Evaluate how you are going to “pull-out” when things go wrong
- The more complex systems are the more riskier this rollout may become
- Understand the unpredictability of a model

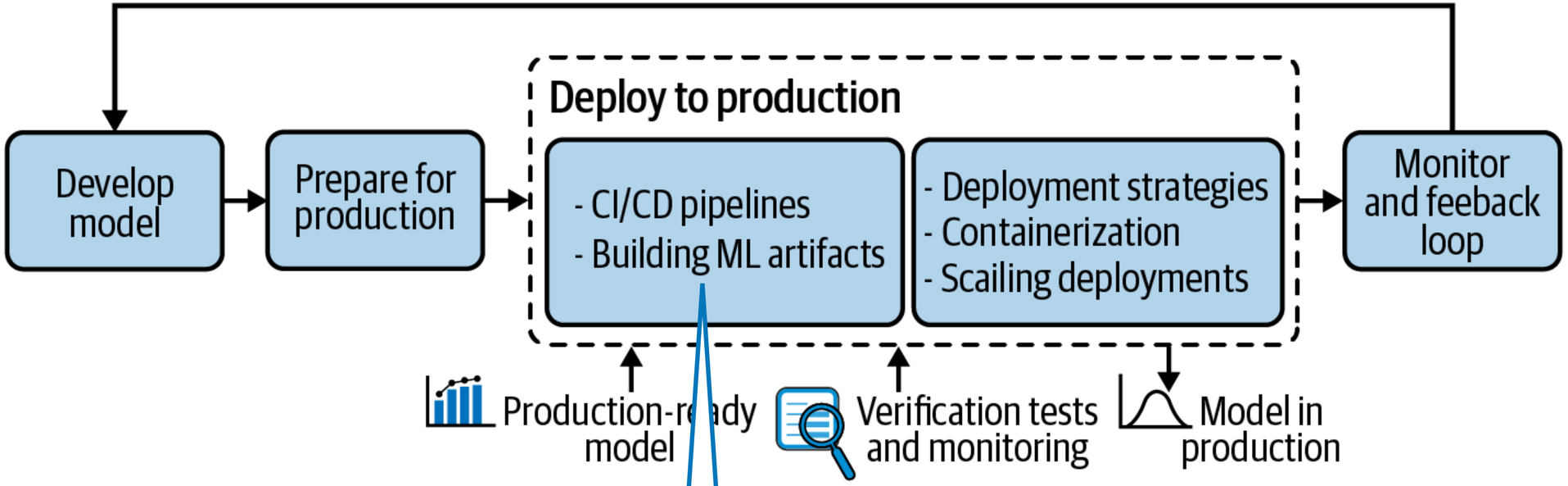
Deploy to Production


Develop model
Prepare for production
Deploy to production
- CI/CD pipelines
- Building ML artifacts
- Deployment strategies
- Containerization
- Scailing deployments
Monitor and feeback loop
Production-ready model
Verification tests and monitoring
Model in production



CI/CD Pipelines

- After successfully developing a model, a data scientist should push the code, metadata, and documentation to a central repository and trigger a CI/CD pipeline.
- Start simple for your needs and then expand your pipelines for what you require
- There are also MLOps Pipelines that can be used in place of CI/CD pipelines

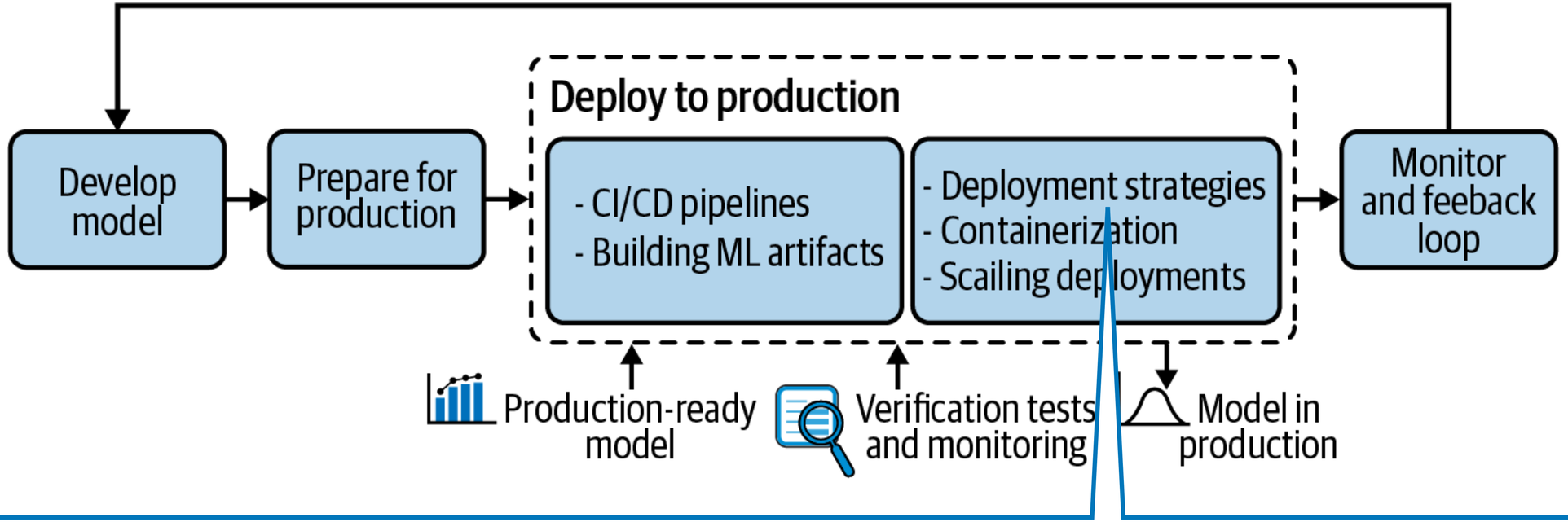


Building ML Artifacts

- An ML Artifact is a testable and deployable bundle of the project must be built once the data and code are in a repository
- Include all materials that led up to building a model

What is in an ML Artifact?

- Code for the model and its preprocessing
- Hyperparameters and configuration
- Training and validation data
- Trained model in its runnable form
- An environment including libraries with specific versions, environment variables, etc.
- Documentation
- Code and data for testing scenarios



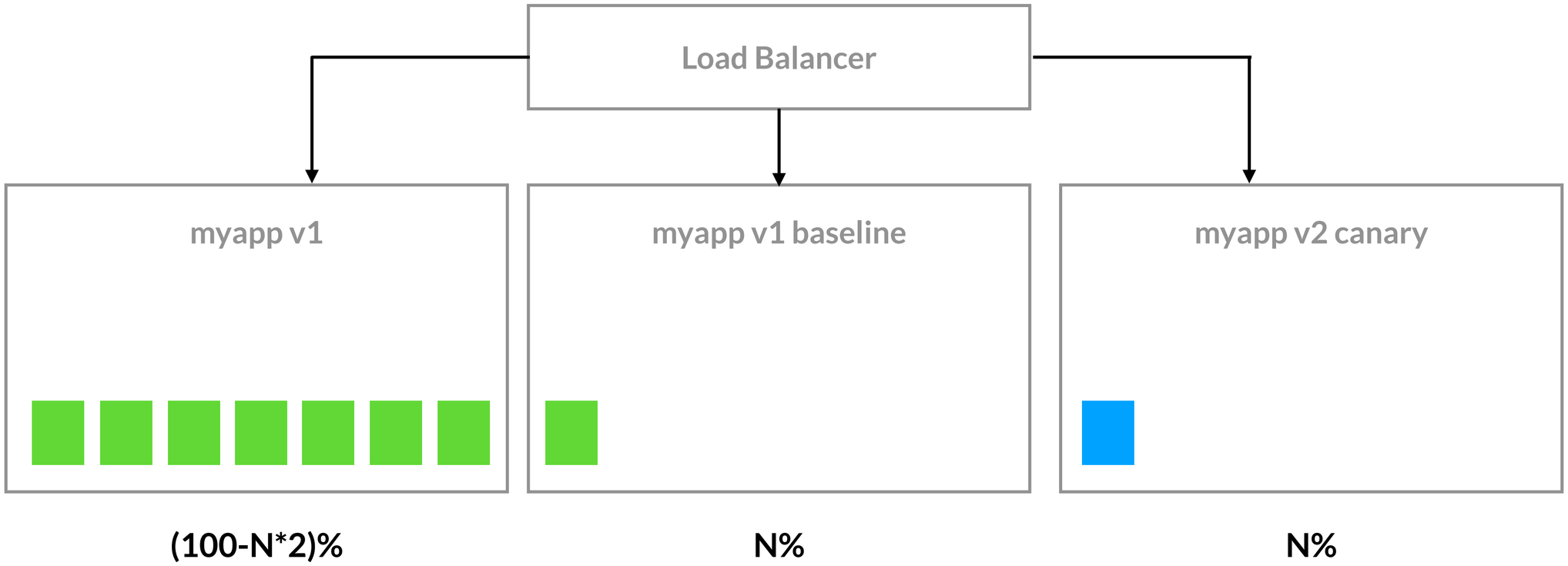
Deployment Strategies

- How are you going to deploy the model? What is the process of merging?
- How will run the model on a target infrastructure ?
- What is release strategy to onboard a new model?

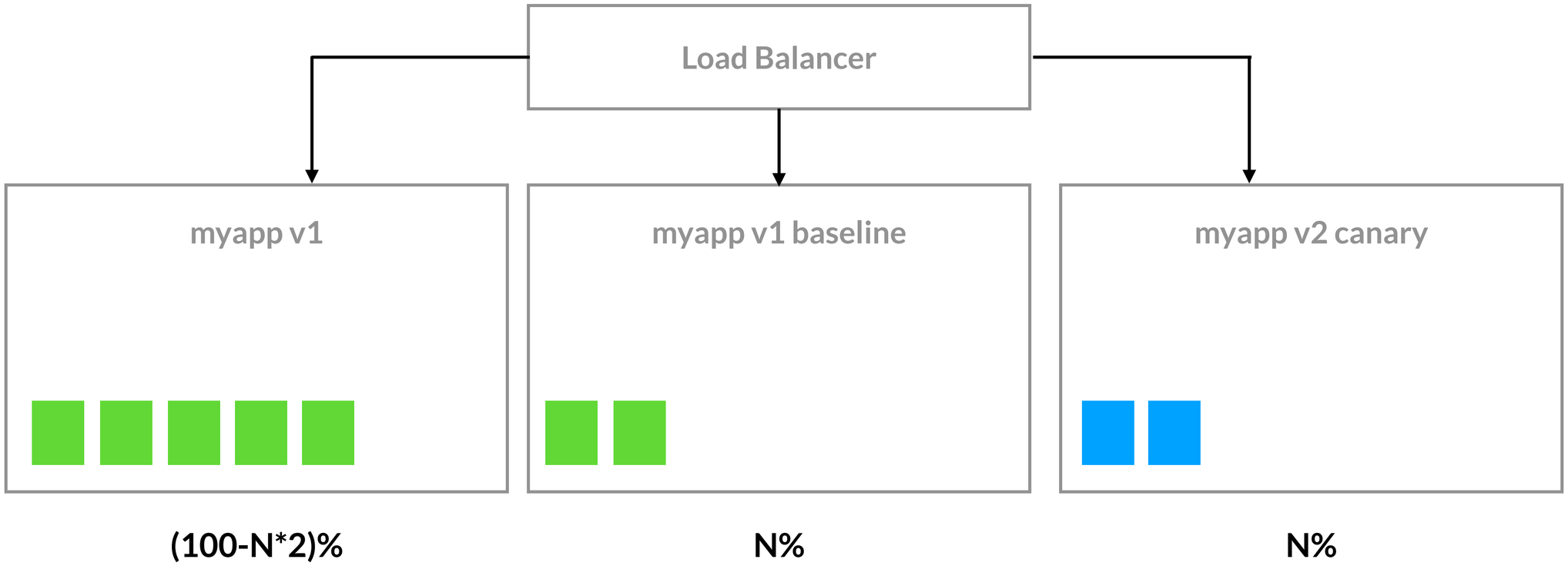
Scoring

- **Batch scoring:** where whole datasets are processed using a model, such as in daily scheduled jobs.
- **Real-time scoring:** where one or a small number of records are scored, such as when an ad is displayed on a website and a user session is scored by models to decide what to display.

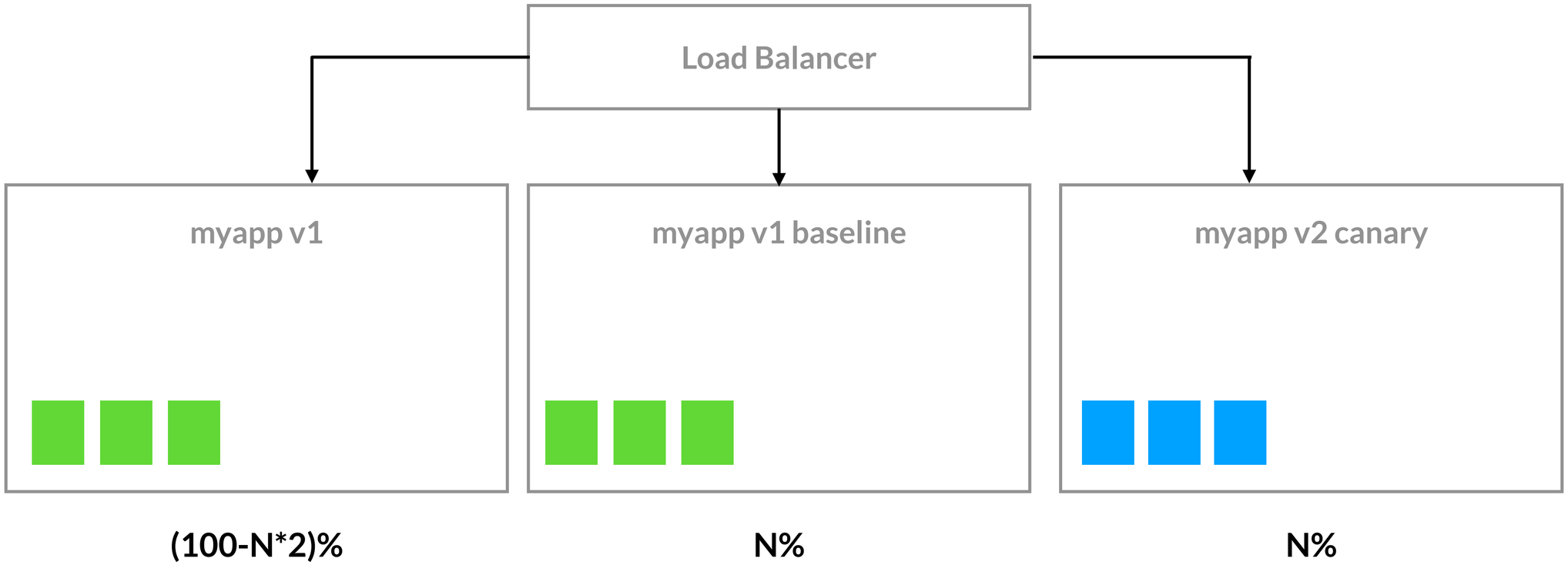
Canary Deployments



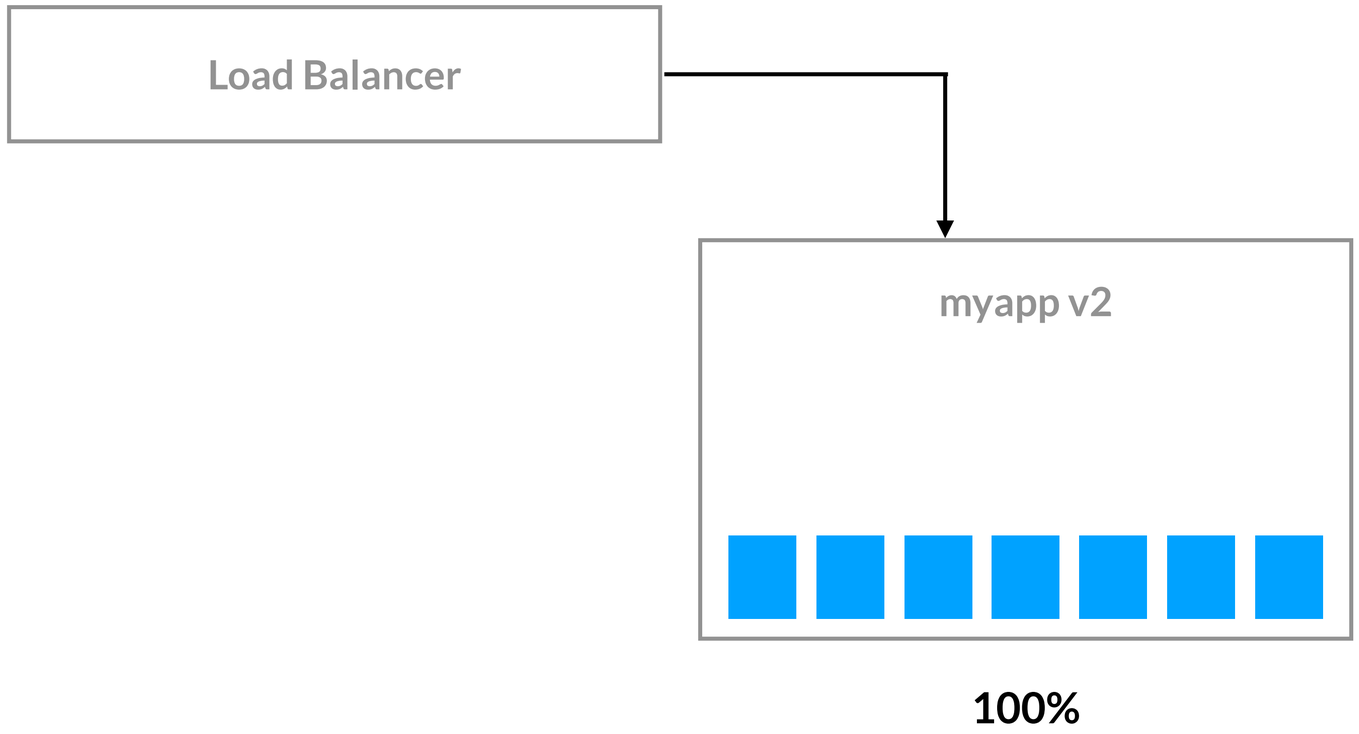
Canary Deployments

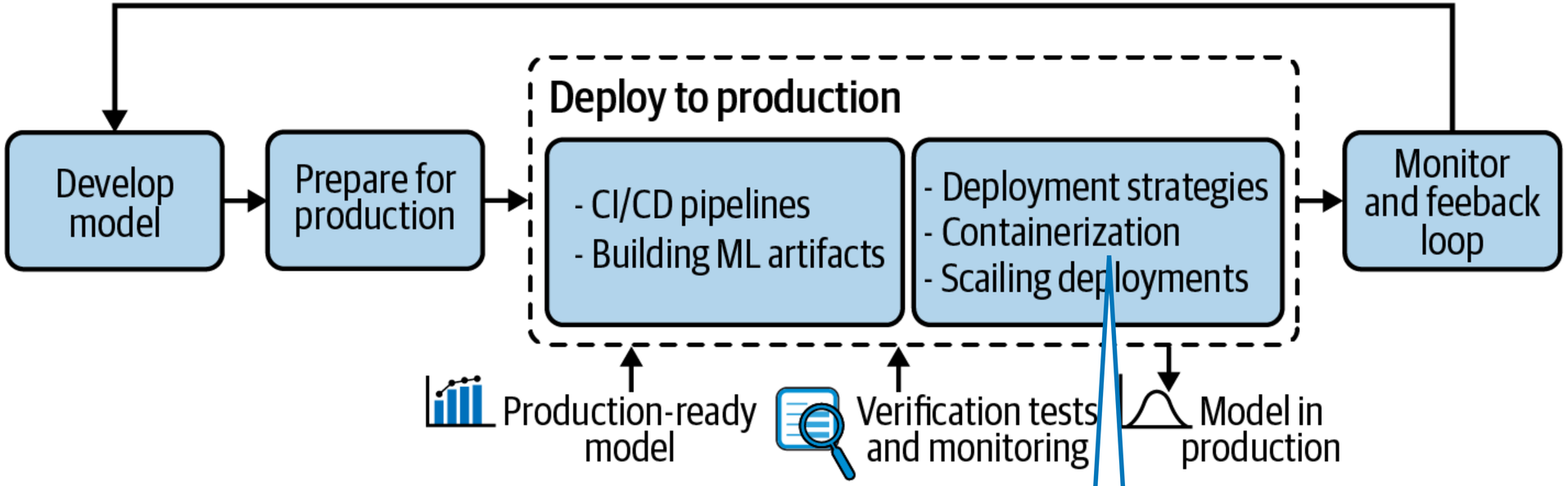


Canary Deployments



Canary Deployments



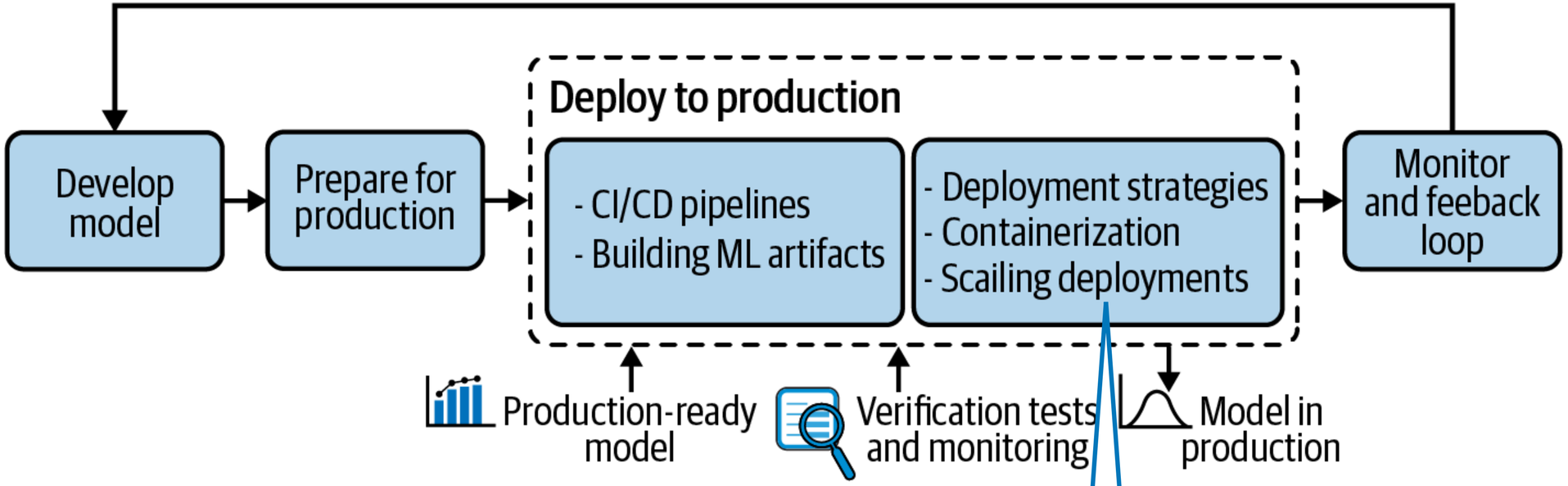


Containerization

- Create a model artifact to be deployed onto a platform
- The can either be done as a virtual machine, or a container like Docker or Podman
- Using containers make things portable and can be used in platforms like Kubernetes

Serving

- Serving is taking a model and creating containers that will run the models through an endpoint
- The models will be served in your choice of protocol: REST, GRPC, anything else proprietary
- Wide variety of choices TFServing, KServe, Seldon Core, Torch Serve
- Cloud Providers available: AWS Sagemaker, Google AI Platform Prediction, Azure Maching Learning Model Deployment



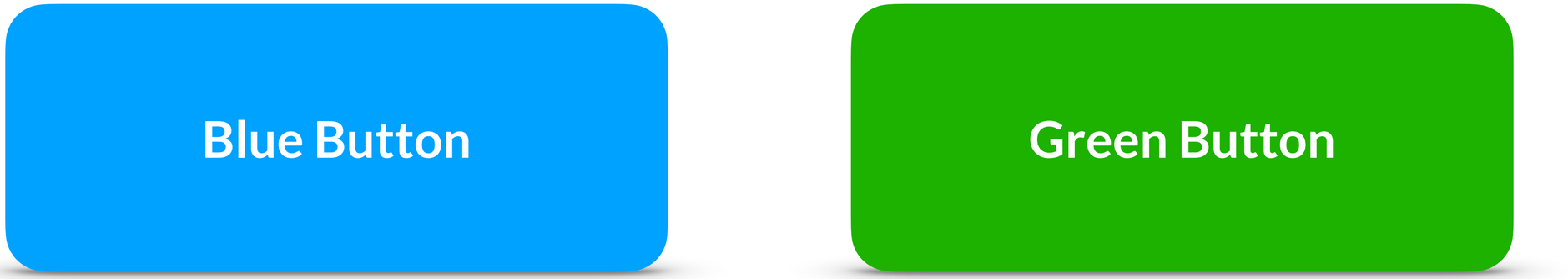
Scaling Deployments

- Once items are rolled out, questions should be known prior:
  - How do we deal with our model with high scale data?
  - How do we train larger and larger number of models
- How do maximize technologies like Spark and Kubernetes?

A/B Testing

- When we have a new release and an old release we can perform A/B Testing.
- Where the new release, or canary is the "experiment" group and the baseline is the "control" group
- We can then test to see if a new feature is equal, less, or more effective that what it currently is.
- For the results of an A/B test to be reliable, they must be statistically significant.
  - This means the differences observed between A and B are unlikely to have occurred by chance.
  - Statistical tools and techniques like p-values, z-values, confidence variables and other statistical functions that would be necessary here to determine if the results are significant enough

A/B Testing



We hypothesize that changing the “Buy Now” button color from blue to green will lead to higher conversions”

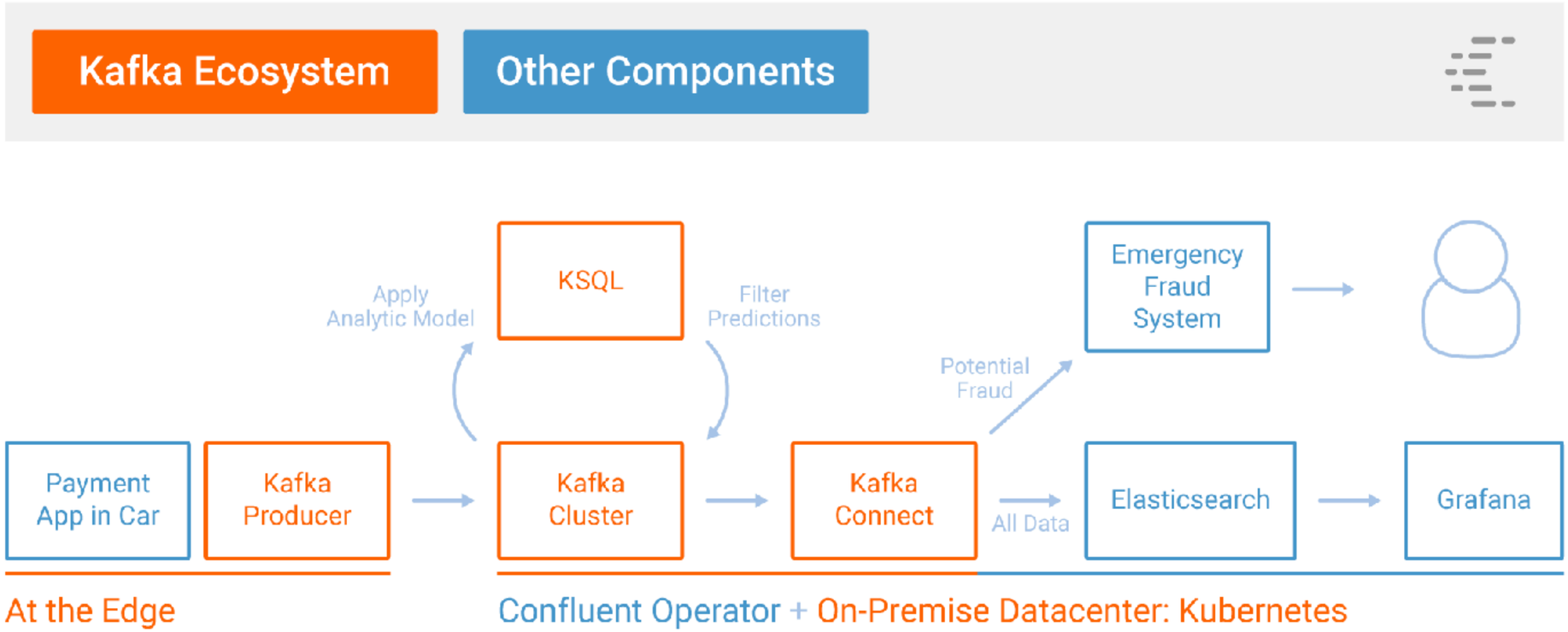
Streaming

- Once a model has been deployed we can stream the data using one of your favorite streaming frameworks: Flink, Kafka Streams, Spark Streaming
- When creating a container with your model you can choose to create a side car with your streaming application so that it can retrieve data, process it with your machine learning model, and then post the results into another topic


kafka

KSQLDB User Defined Functions

```java
package com.example.ksql.functions;

import io.confluent.ksql.function.udf.Udf;
import io.confluent.ksql.function.udf.UdfDescription;
import io.confluent.ksql.function.udf.UdfParameter;

@UdfDescription(
    name = "reverse",
    description = "Example UDF that reverses an object",
    version = "0.1.0",
    author = ""
)
public class ReverseUdf {
  @Udf(description = "Reverse a string")
  public String reverseString(
      @UdfParameter(value = "source", description = "the value to reverse")
      final String source
  ) {
    return new StringBuilder(source).reverse().toString();
  }
}
```



https://www.confluent.io/blog/build-udf-udaf-ksql-5-0/

Creating a UDF Model with H2O

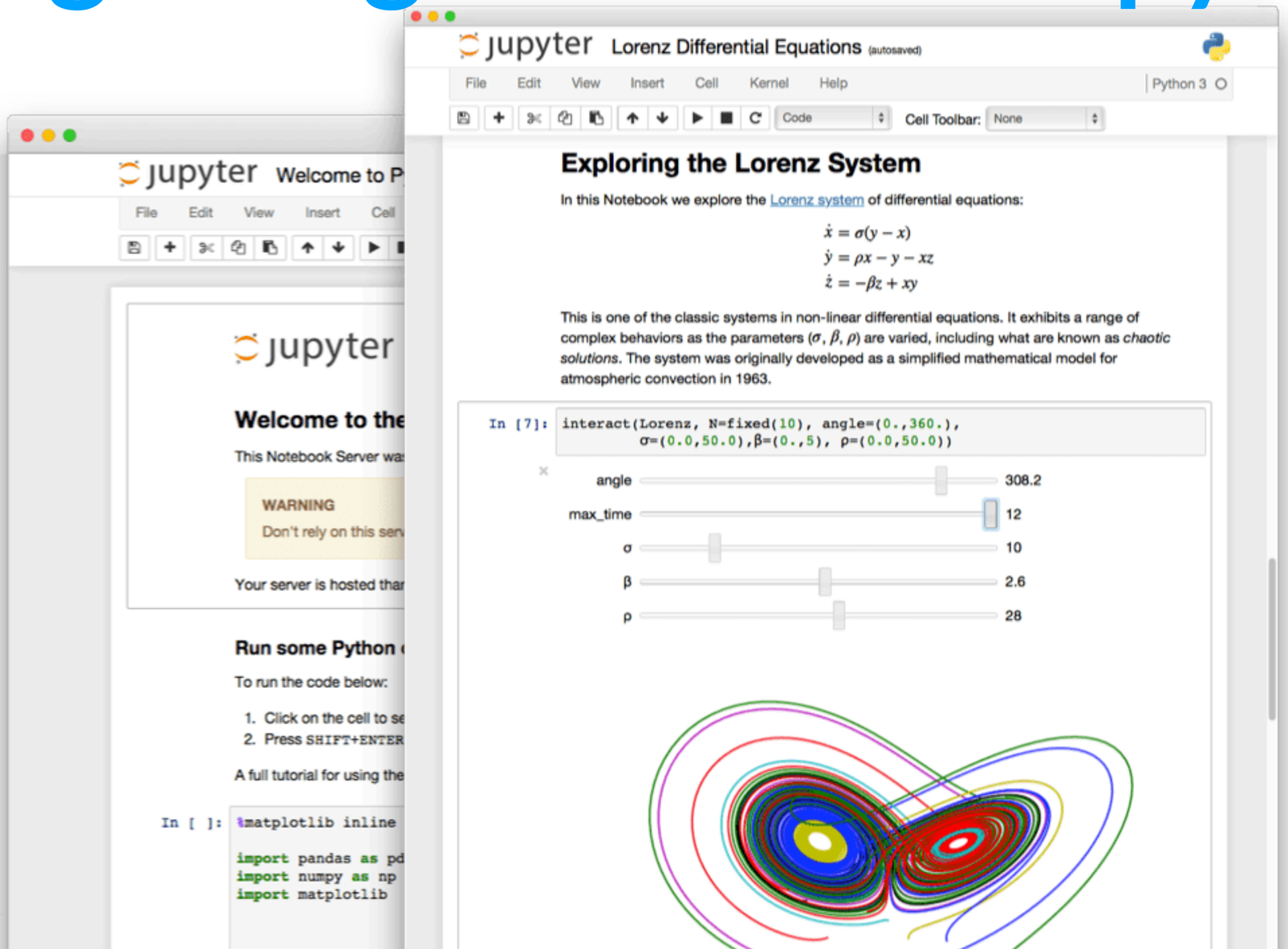
```
GenModel rawModel;

rawModel = (hex.genmodel.GenModel)
Class.forName(modelClassName).newInstance();

EasyPredictModelWrapper model = new
   EasyPredictModelWrapper(rawModel);
```

H2O.ai

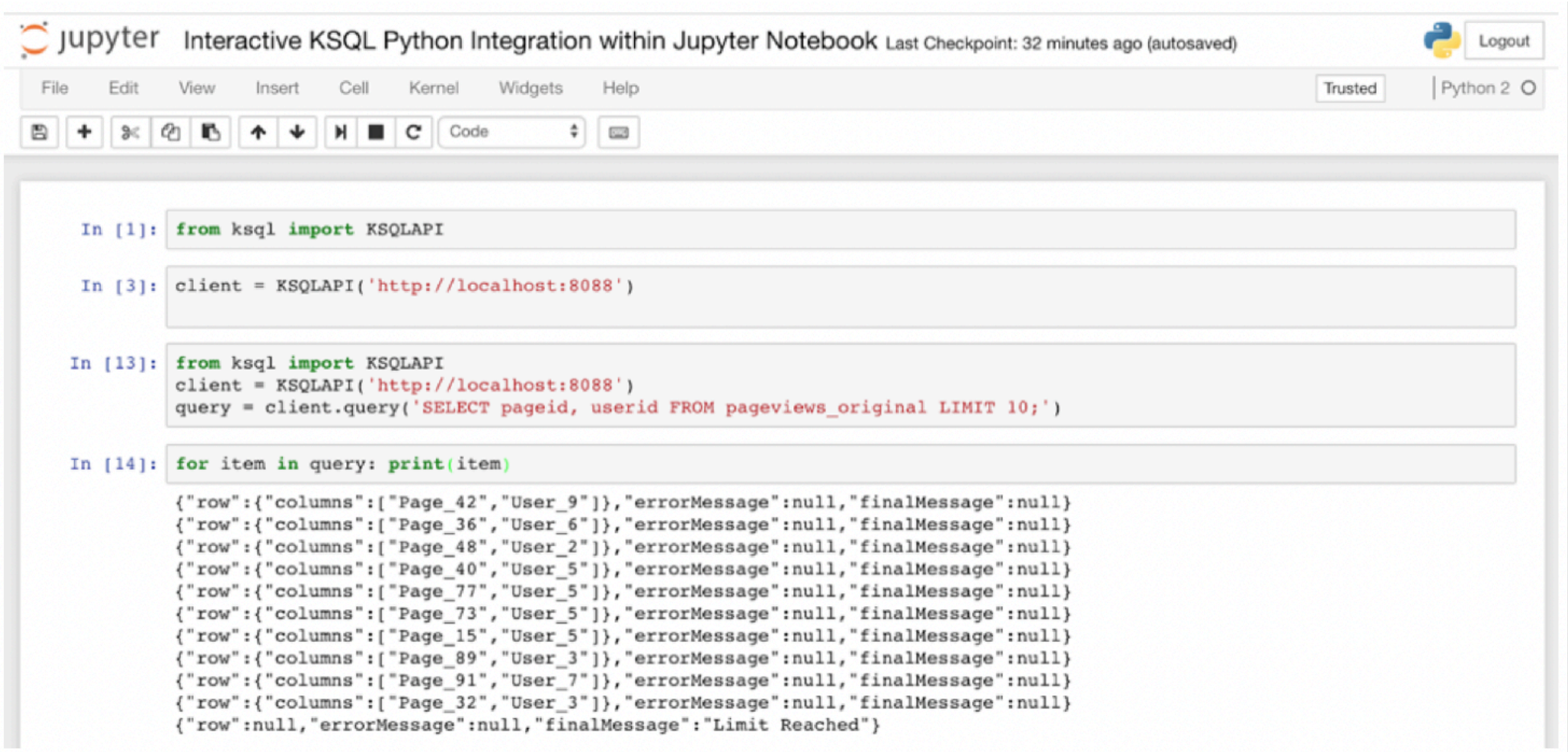
```
SELECT car_id, event_id, ANOMALY(sensor_input) FROM car_sensor;
```

Source: https://bit.ly/32zEgB0

Integrating Results to Jupyter

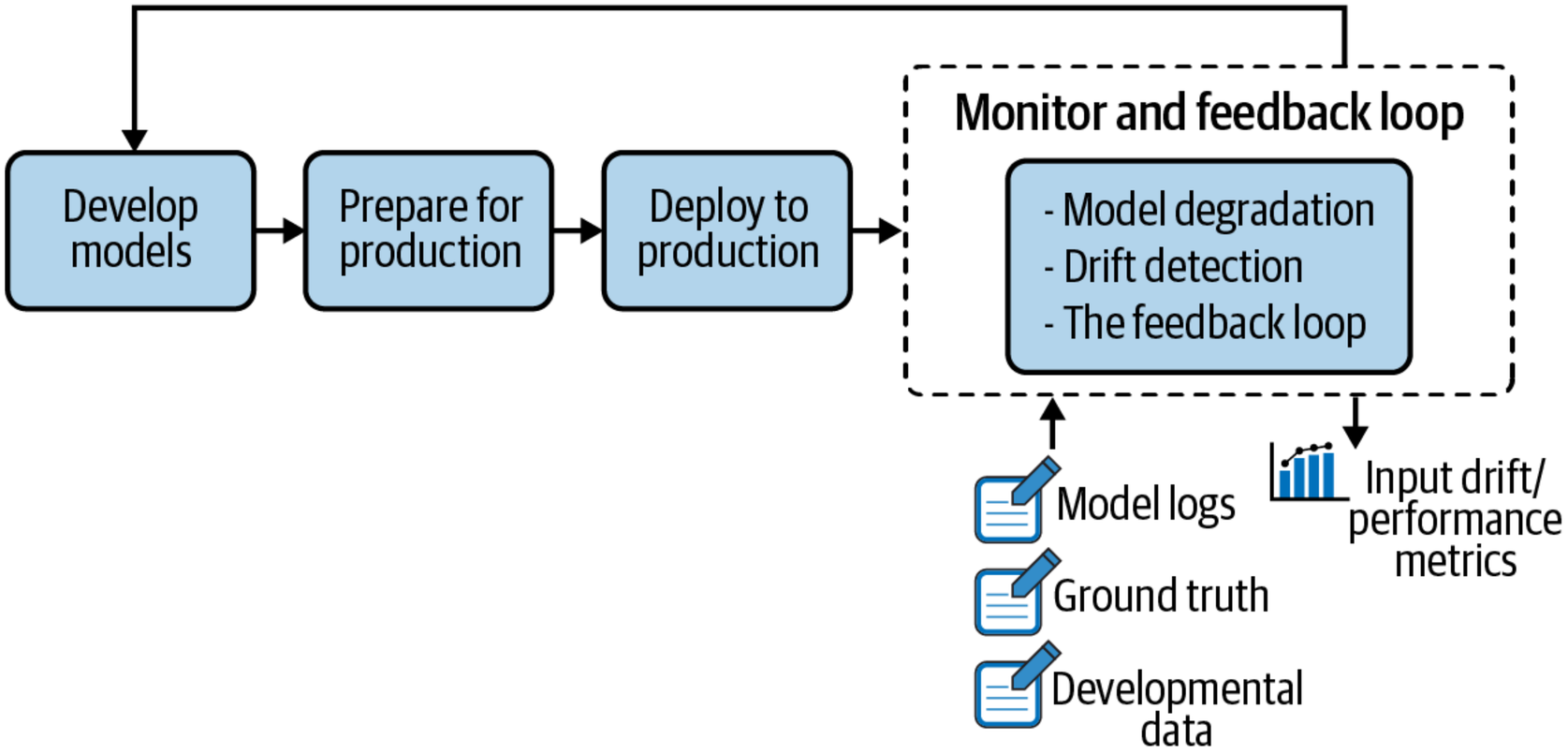


Viewing Real-time results in KSQLDB



This enables live use of data through the notebook to experiment with possible modelling

Monitoring a Feedback


Develop models
Prepare for production
Deploy to production
Monitor and feedback loop
- Model degradation
- Drift detection
- The feedback loop
Model logs
Ground truth
Developmental data
Input drift/ performance metrics

Monitoring Levels

dev ops

- At the **resource level**, including ensuring the model is running correctly in the production environment.
    - **Key questions include:** Is the system alive? Are the CPU, RAM, network usage, and disk space as expected? Are requests being processed at the expected rate?

ml ops

- At the **performance level**, meaning monitoring the relevance of the model over time.
    - **Key questions include:** Is the model still an accurate representation of the pattern of new incoming data? Is it performing as well as it did during the design phase?

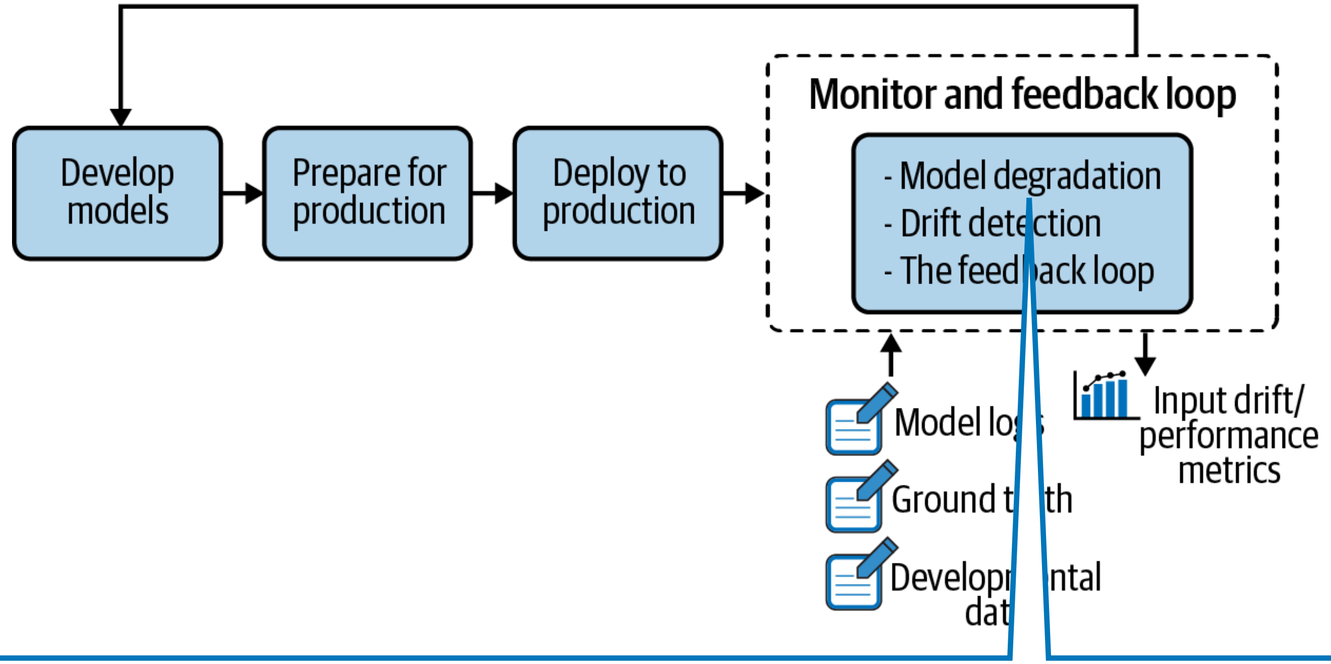
Retraining

- Retraining in MLOps is the process of updating a machine learning model by training it on new or updated data to maintain or improve its performance over time. Key elements include:
  - **Continuous Improvement:** Regularly retraining models helps address issues like data drift, where changes in input data affect model accuracy.
  - **Automated Pipelines:** MLOps enables automated retraining pipelines, where models are periodically retrained based on new data, predefined schedules, or performance thresholds.
  - **Performance Monitoring:** Retraining is often triggered when monitoring detects performance degradation or shifts in data patterns.
  - **Deployment:** Retrained models are deployed back into production to ensure they can adapt to changing real-world conditions.

When to retrain?

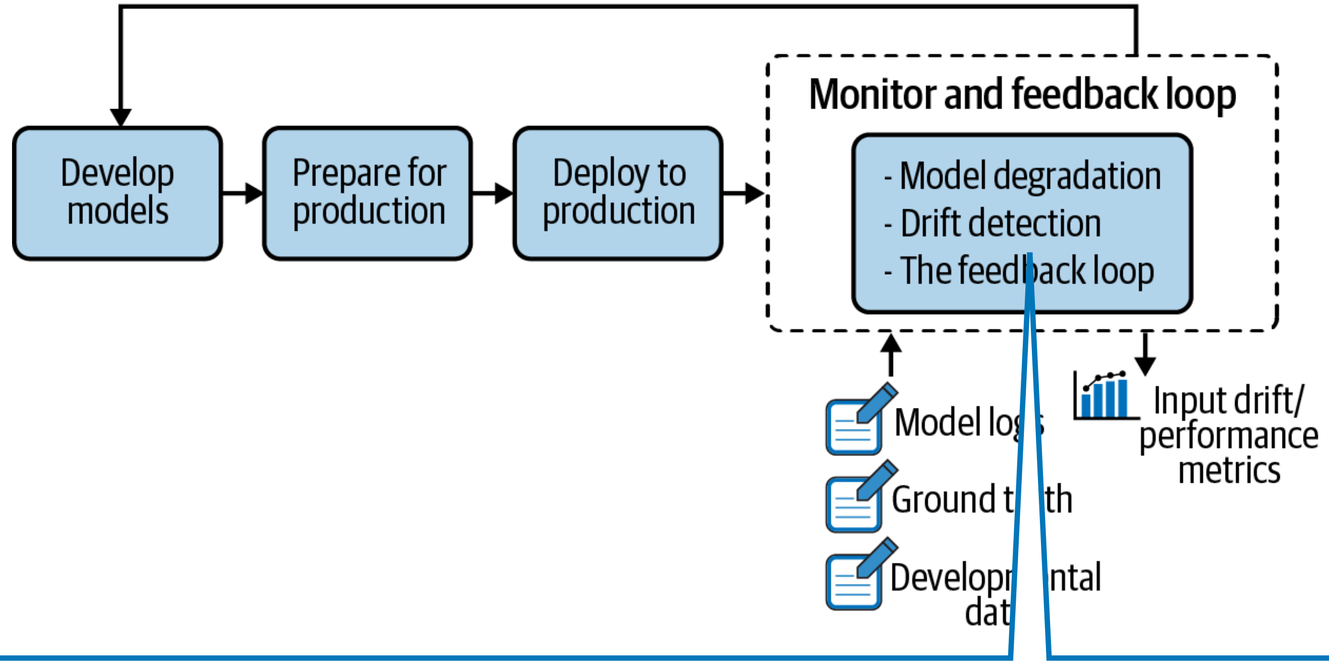
- **Upper Bound:** It is better to perform retraining once every year to ensure that the team in charge has the skills to do it.

- **Lower Bound:** In case of models with instantaneous feedback on whether the model works as expected (clicks as opposed to credit card approval) an A/B Testing or statistical valuation can help determine if retraining is required



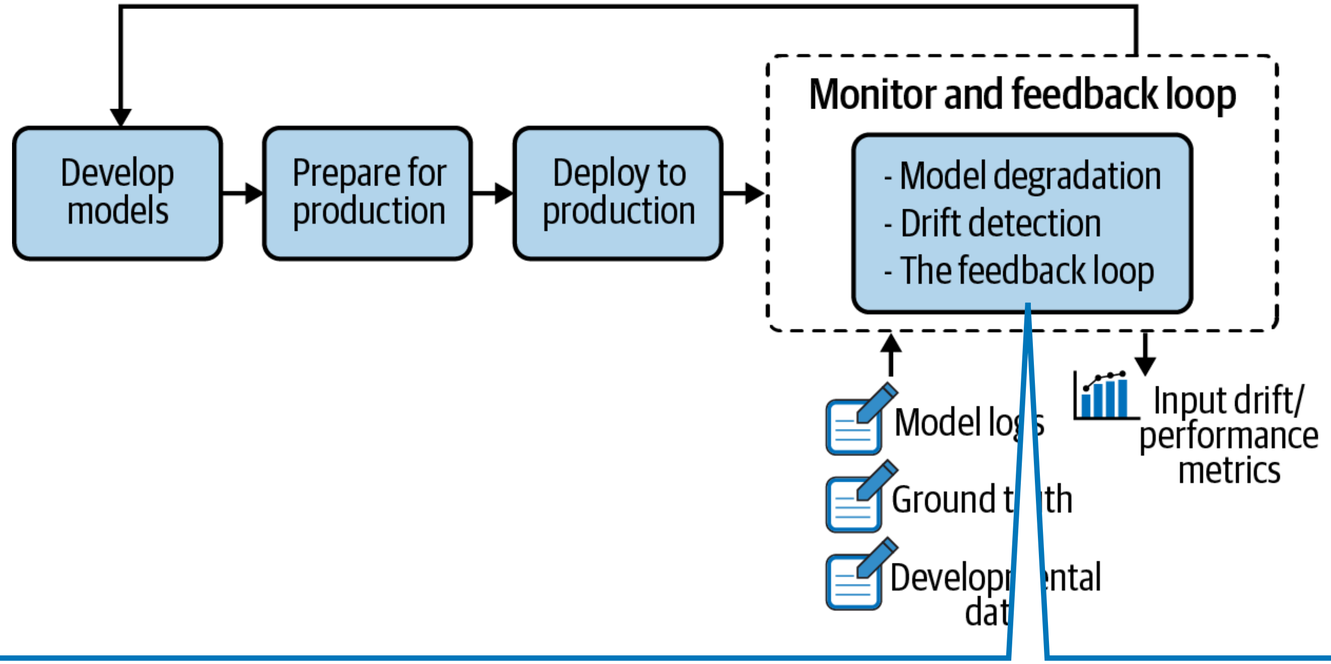
Model Degradation

- Monitor the ground truth - waiting for the label event, once the event is available, we can then calculate the performance, if it surpasses a threshold then retraining is in order
- Use ROC AUC, Log Loss or other Specific Metrics like p-values



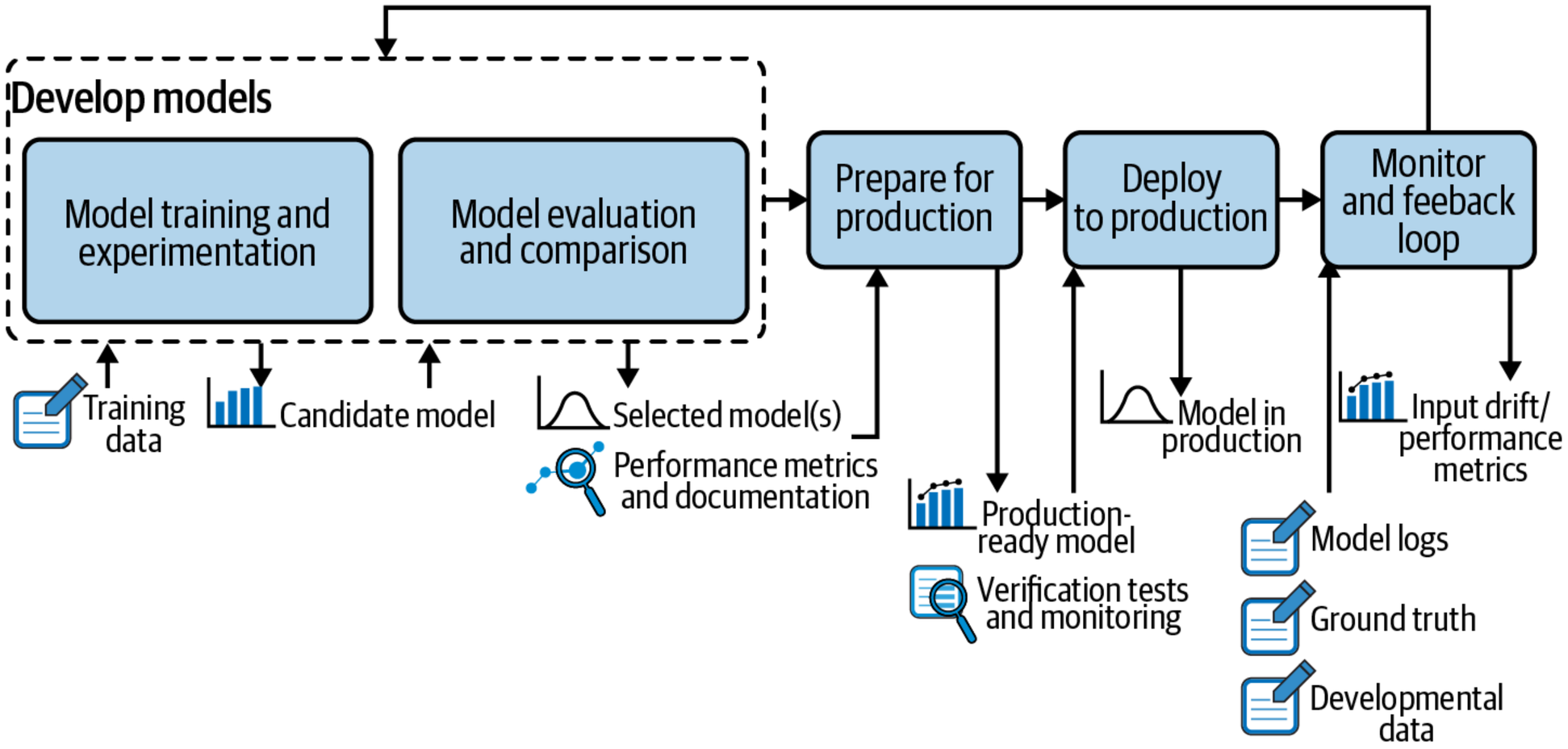
Input Drift Detection

- Are inputs going to change the results drastically, are inputs identically distributed?
- Are there duplicates? If samples are duplicated that can lead to erroneous results



Feedback Loop

- Information from the production environment flows back to the model prototyping environment for further improvement.
- Possibility of retraining the model with new labeled data or developing a new model with additional features.


Develop models
Model training and experimentation
Model evaluation and comparison
Prepare for production
Deploy to production
Monitor and feeback loop
Training data
Candidate model
Selected model(s)
Performance metrics and documentation
Production-ready model
Verification tests and monitoring
Model in production
Input drift/ performance metrics
Model logs
Ground truth
Developmental data

What technologies can be considered?

Apache Airflow

What does Apache Airflow bring to MLOps?

- Airflow excels at orchestrating and scheduling workflows and it can help automate:
  - Data collection and preprocessing
  - Training ML Models
  - Model Evaluation and Hypertuning
  - Model Deployment
  - Monitoring

What does Apache Airflow not bring to MLOps?

- Airflow is not suitable for :
  - Experiment Tracking
  - Model Registry
  - Model Serving
  - Advanced ML Workflows
- Airflow is not specifically designed for ML management

Can they combined with other tools?

- Airflow can compliment other tools like MLFlow, Kubeflow, or Seldon Core by orchestrating vairous stages of ML Lifecycle

- **MLFlow + Airflow:** Airflow can trigger different stages of an ML pipeline (e.g., data processing, model training, and deployment), while MLFlow handles experiment tracking, model versioning, and serving.

- **KubeFlow + Airflow:** KubeFlow can manage the end-to-end ML lifecycle (including model training, pipelines, and serving), and Airflow can be used to schedule and trigger those pipelines at appropriate intervals.


SELDON

What does Seldon Core bring to MLOps?

- Seldon Core brings MLOps capabilities by providing a Kubernetes native platform for common MLOps tasks
  - Model Versioning
  - Canary Deployments
  - A/B Testing
  - Monitoring for Outliers, Drift, and Explainability

What does Seldon Core not bring to MLOps?

- Seldon Core does not provide feature for managing full machine learning lifecycle
  - Experiment tracking, model training, data preprocessing, or hyperparameter tuning
  - Lacks tools for data versioning, pipeline orchestration (e.g., like Kubeflow Pipelines or Apache Airflow), automatic model retraining
- These tasks require integration with other tools (e.g., MLFlow, Kubeflow, or Airflow) to cover the complete MLOps workflow beyond just model deployment and serving.


mlflow™

What does MLFlow bring to MLOps?

- MLFlow brings key functionalities to MLOps by providing tools for managing the entire machine learning lifecycle:
  - Experiment Tracking
  - Model Versioning
  - Packaging Models
  - Centralized Model Registry
  - Logs parameters and metrics

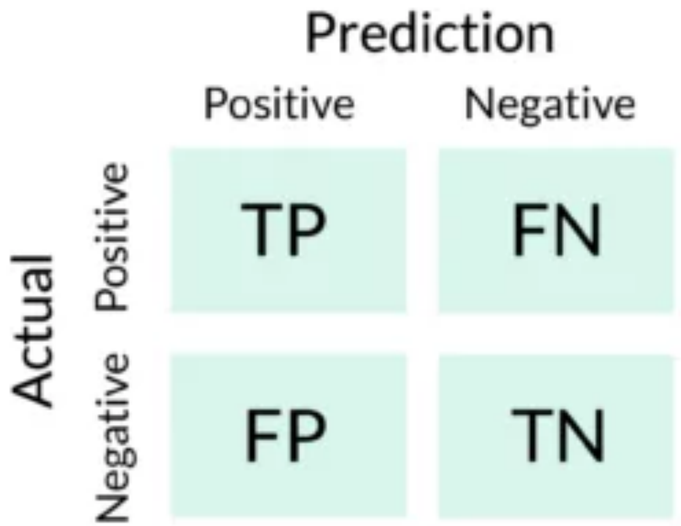
What does MLFlow not bring to MLOps?

- MLFlow does not provide built-in support for distributed model training, pipeline orchestration, or automated workflow management Experiment Tracking
  - Model Versioning
  - Packaging Models
  - Centralized Model Registry
  - Logs parameters and metrics

Model Scoring

Accuracy

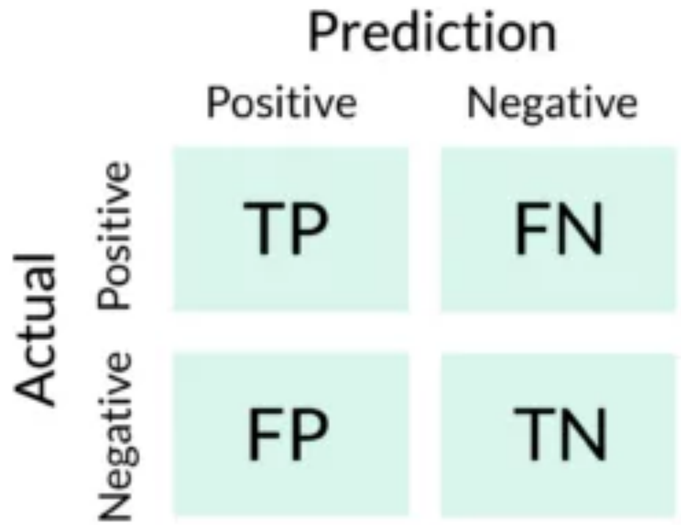
- Measures the percentage of correct predictions over the total predictions.
- (True Positives + True Negatives) / Total Instances



Precision

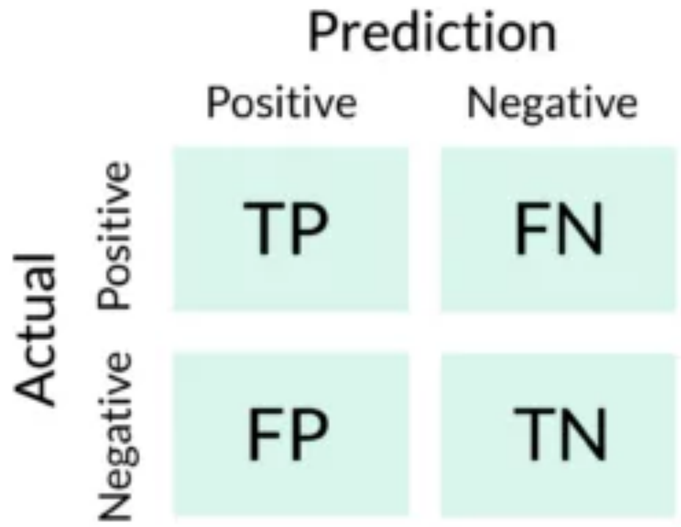
- **Indicates the proportion of true positive predictions among all positive predictions.**

- **True Positives / (True Positives + False Positives)**



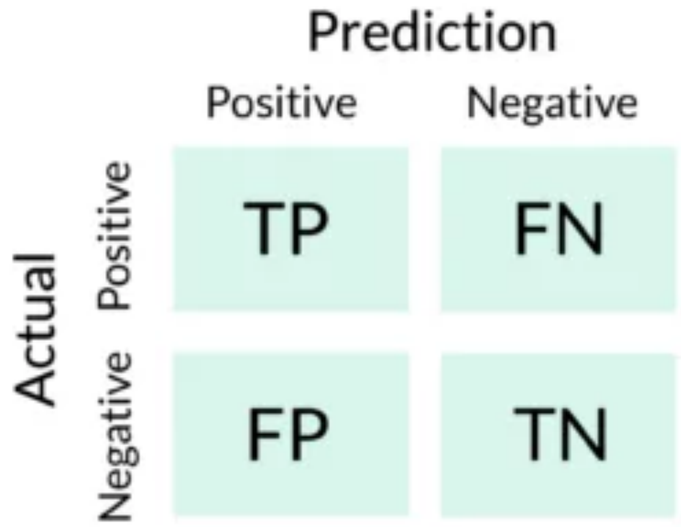
Recall

- **Measures the ability to identify all actual positive cases.**
- **True Positives / (True Positives + False Negatives)**



F1-Score

- **Harmonic mean of precision and recall, balancing their trade-offs.**
- **2 * (Precision * Recall) / (Precision + Recall)**



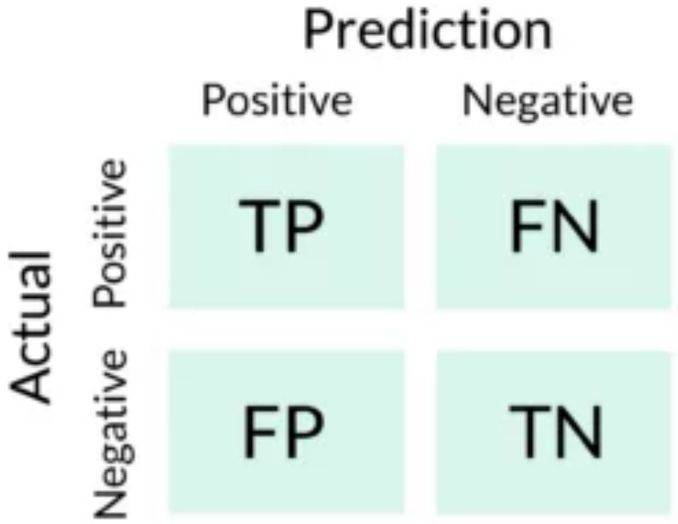
ROC-AUC

ROC

- **Receiver Operating Characteristic**
- **The ROC curve** is a popular metric for evaluating the performance of classification models, particularly in binary classification problems.
- **The ROC curve** was first used during World War II for the analysis of radar signals before it was employed in signal detection theory.

$$TPR = TP / (TP + FN)$$

$$FPR = FP / (FP + TN)$$



AUC

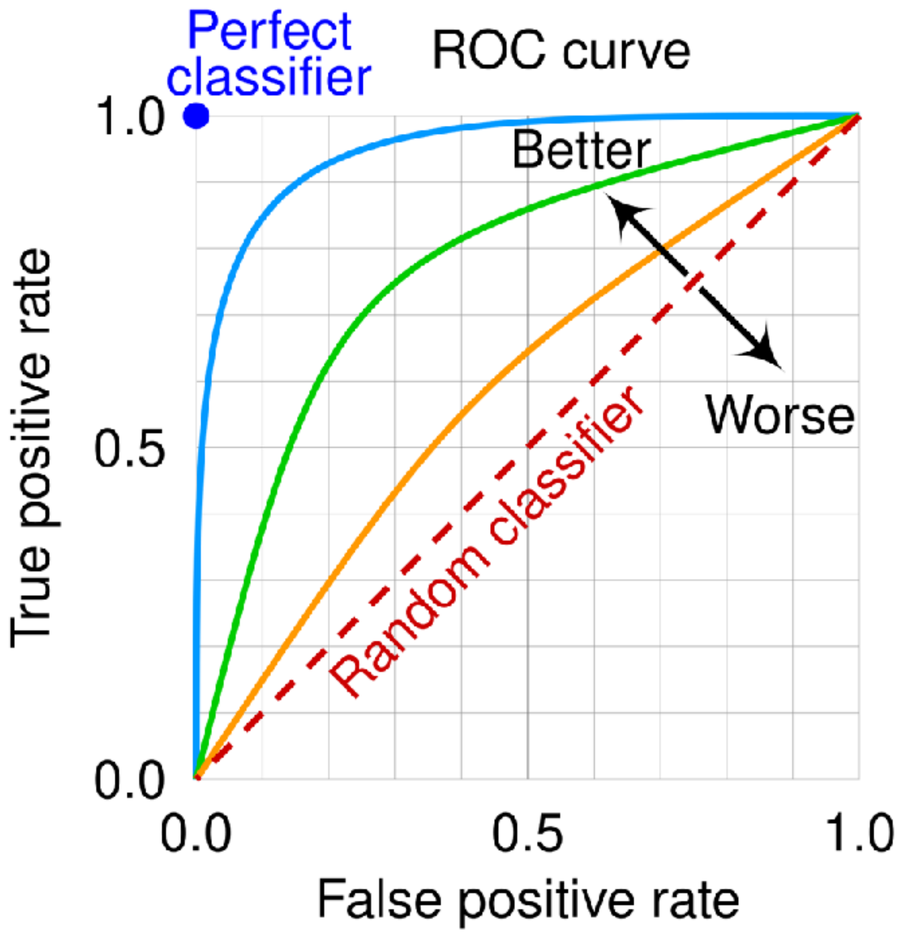
- **Area Under the Curve (AUC)** measures the area under the ROC curve, providing a single scalar value to summarize the model's performance

- AUC ranges from 0 to 1:

  - **1.0:** Perfect classifier.

  - **0.5:** No discriminative ability (random guessing).

  - **< 0.5:** Worse than random guessing (rare and suggests issues with the model).



Lab: MLFlow

- Let’s try some of MLFlow
- MLFlow is lightweight and perfect for docker-compose

Kubeflow

What does KubeFlow bring to MLOps?

- Kubeflow brings a comprehensive, Kubernetes-native platform to MLOps, offering a set of tools for building, deploying, and managing machine learning workflows at scale. Its key contributions to MLOps include:
  - Kubeflow Pipelines and Scaling using Kubernetes
  - Distributed Training
  - Hyperparameter Tuning using Katib
  - Kubeflow Serving and Advanced Serving Features
  - Experiment Tracking
  - Framework Agnostic

What does KubeFlow not bring to MLOps?

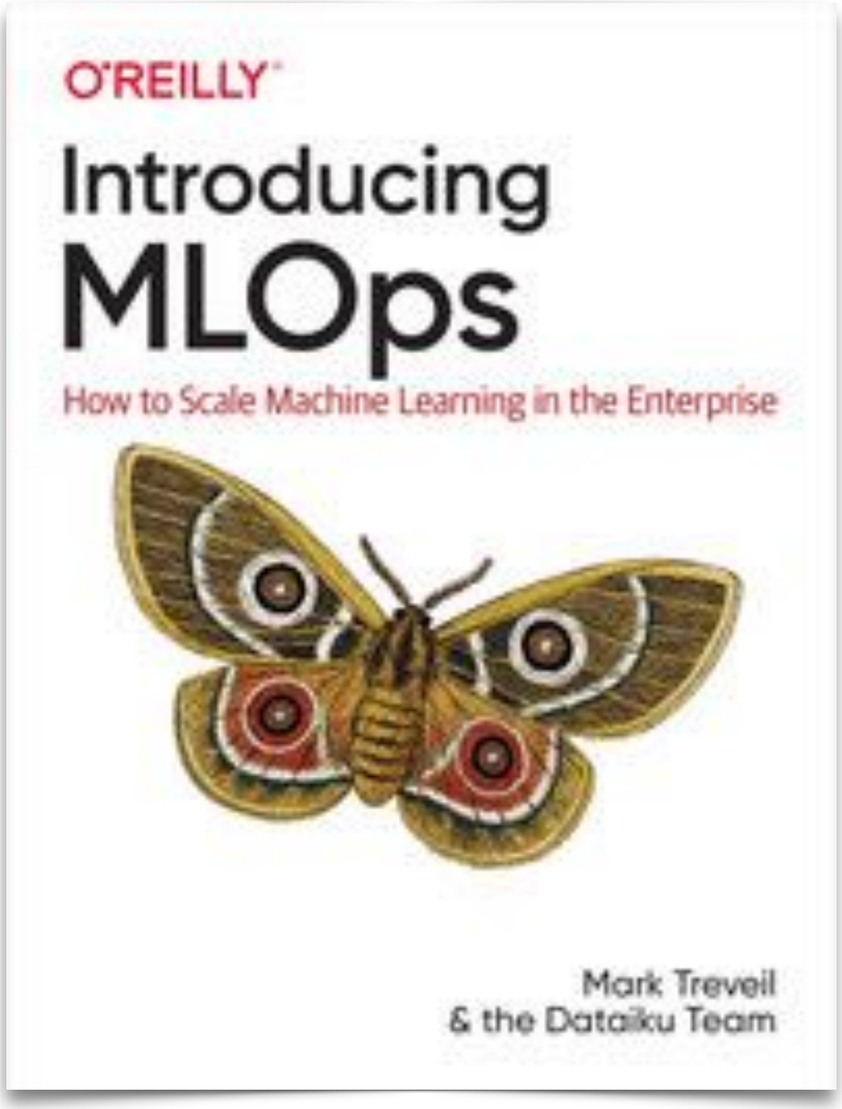
- Data Versioning - Kubeflow doesn’t offer built-in data version control. You’ll need to integrate other tools for managing datasets.
- Lightweight Setup - Kubeflow can be complex to install and configure due to its Kubernetes dependencies and the number of integrated components.
- Experiment Simplicity - While powerful, Kubeflow can be overkill for small, simple machine learning experiments. Tools like MLFlow or DVC might be more appropriate for smaller-scale use cases.

Grafana

What does Prometheus and Grafana bring to MLOps?

- In MLOps both are used to track performance and health of ML Models and Infrastructure
- Specific Roles are:
  - **Prometheus** - Monitoring metrics, Time-Series Data, Alerting
  - **Grafana -** Data Visualization, Custom Dashboards, Correlation of Metrics

What books and resources are available?

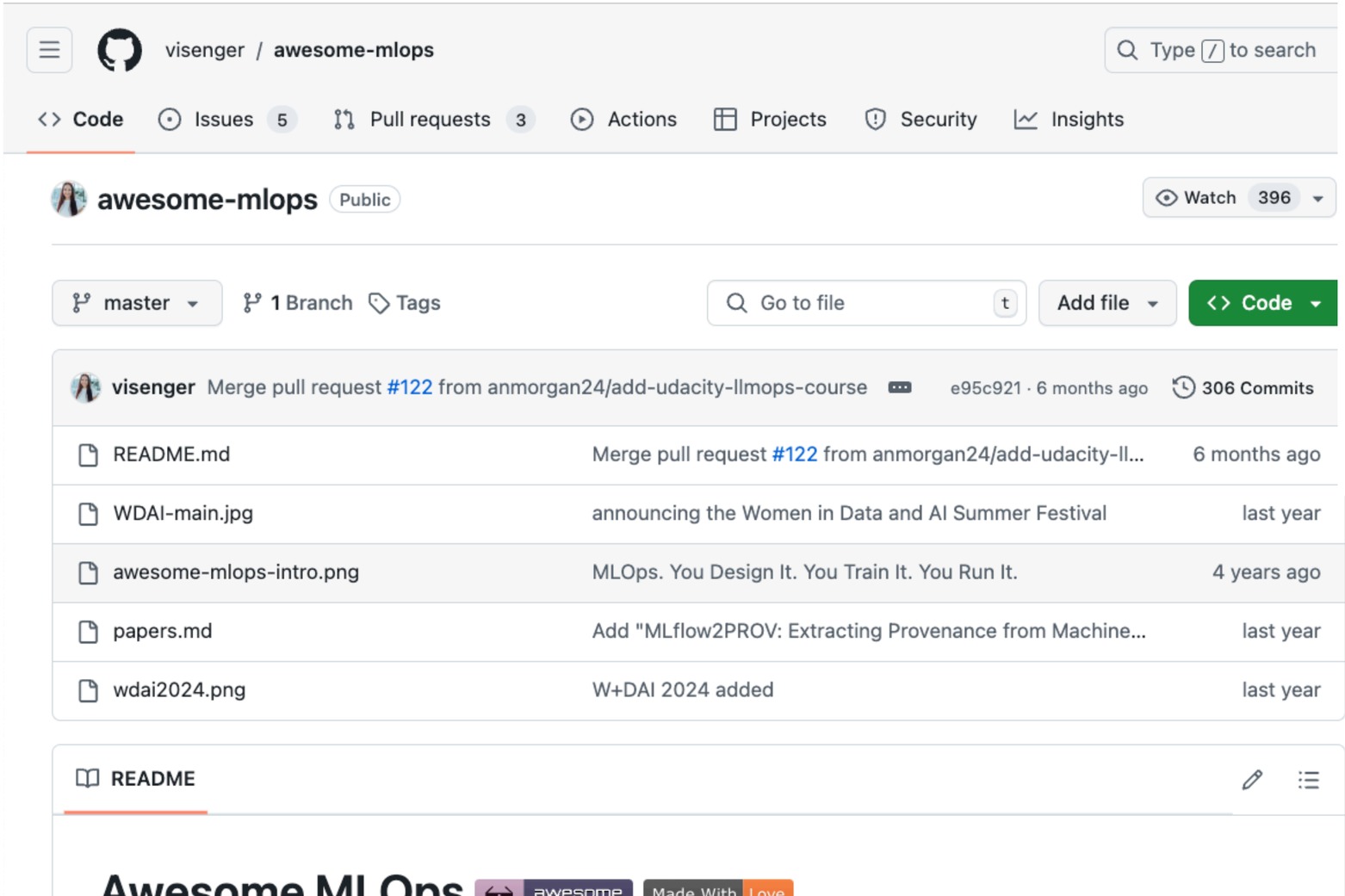


Introducing MLOps

Mark Treveil, Nicolas Omont, Clément Stenac, Kenji Lefevre, Du Phan, Joachim Zentici, Adrien Lavoillotte, Makoto Miyazaki, Lynn Heidmann

Published by O'Reilly Media, Inc.

Awesome-MLOps



machine-learning ai ml software-engineering federated-learning mlops

Readme
Activity
12.6k stars
396 watching
1.9k forks
Report repository

Thank You

- Email: dhinojosa@evolutionnext.com
- Github: https://www.github.com/dhinojosa
- Twitter: http://twitter.com/dhinojosa
- Linked In: http://www.linkedin.com/in/dhevolutionnext