DATAMAX

Introduction to Machine Learning Engineering and MLOps

DATAMAX

ML IN PRODUCTION

Today's Agenda

1. **About me, About Data Max**
   Bujar Bakiu, CEO Data Max
2. **Hidden Technical Debt in Machine Learning**
   Hidden Technical Debt in ML, and why it is an issue
3. **Modern ML Platforms, MLOps**
   Components of ML Platforms, their purpose, role of MLOps
4. **ML Model for Email Classification**
   A ML Model for NLP Classification
5. **Hands-on: Deploying ML Model in Kubernetes**
   Deploying and scaling an ML Model in k8s

DATAMAX

BUJAR BAKIU

Bujar Bakiu

- CEO & Machine Learning Engineer, Data Max
- Graduated 2016, MSc, RWTH Aachen University
- Passionate about Machine Learning, Software Engineering and running
- Find me in Twitter: @bujarbakiu
  LinkedIn: linkedin.com/in/bbakiu
  Email: bujar@data-max.io

DATAMAX

Sadik Bakiu

PRINCIPAL ML ENGINEER

Bujar Bakiu

ML ENGINEER

Deni Myftiu

SYSTEMS ENGINEER

Igli Koxha

ML ENGINEER

We’ re hiring! 👩‍💻👨‍💻👨‍💻

Also, we offer paid internship! 👩‍🎓👨‍🎓👨‍🎓

Find us at: data-max.io

linkedin.com/company/datamax-io

info@data-max.io / bujar@data-max.io

DATAMAX

Hidden Technical Debt in Machine Learning

- **Technical Debt**
- **Why is it hidden?**
- **How to mitigate?**

DATAMAX

MACHINE LEARNING

Machine Learning Workflow

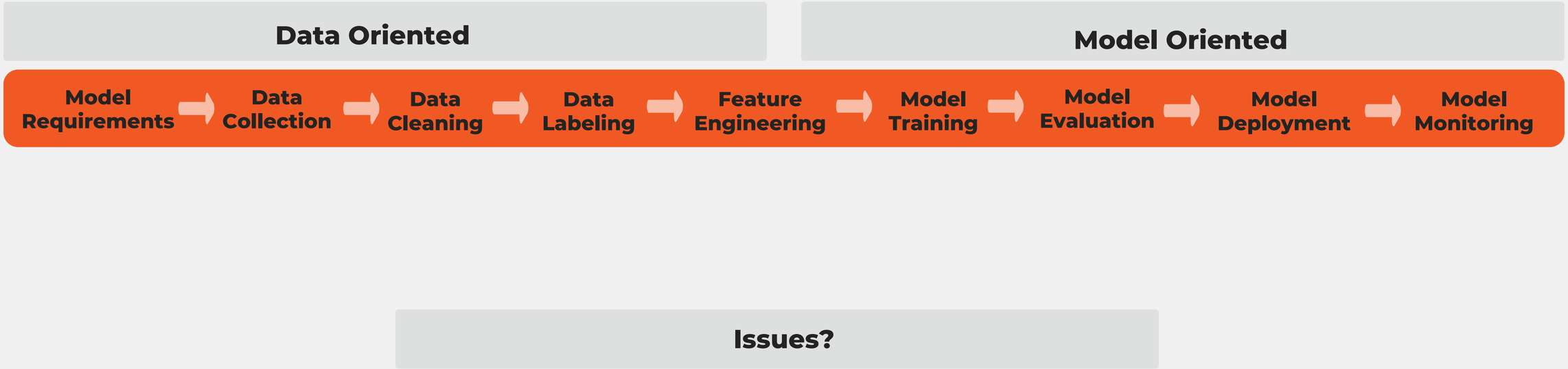
How does a machine learning workflow look like?

DATAMAX

MACHINE LEARNING

Machine Learning Workflow

The nine stages of machine learning workflow.



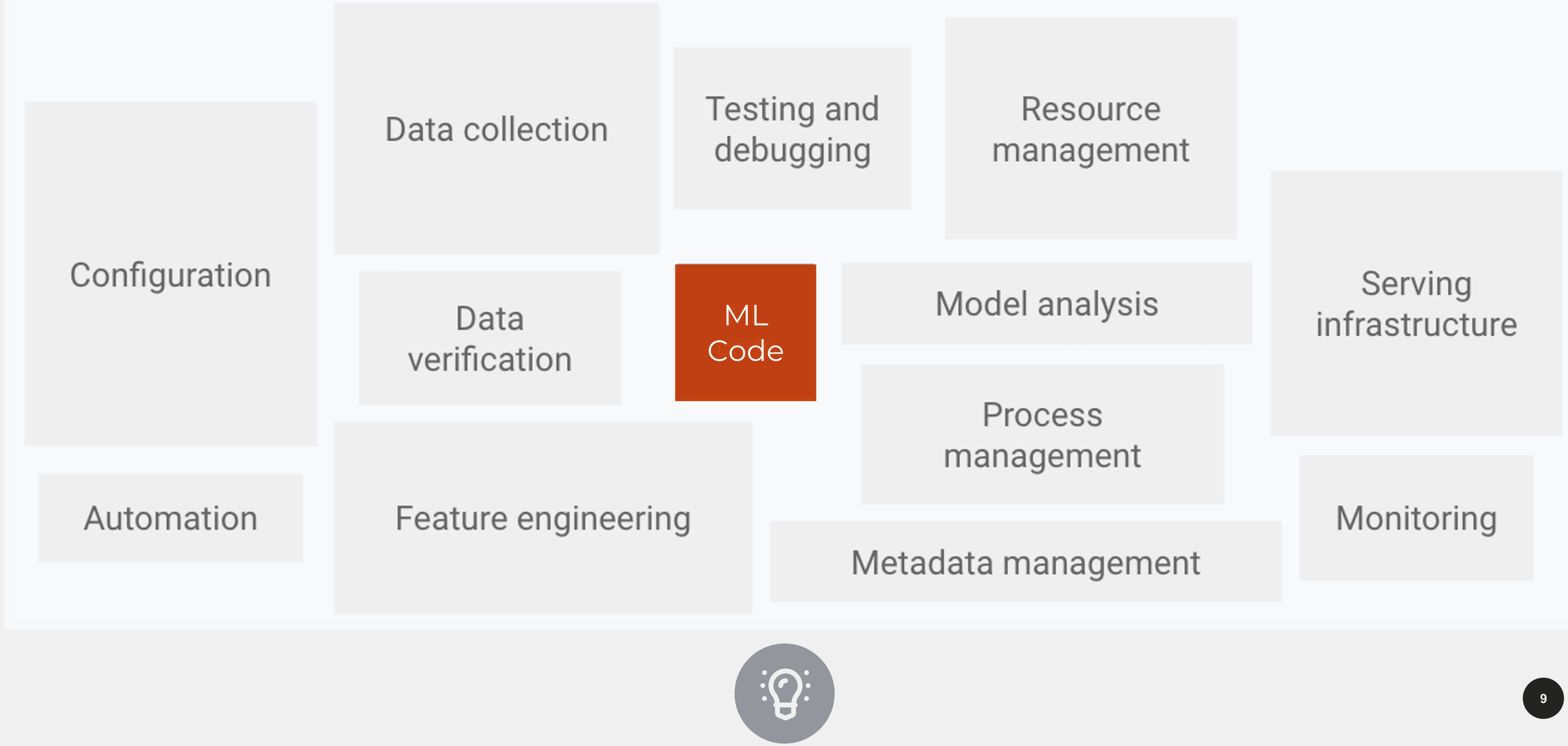
DATAMAX

TECHNICAL DEBT

Hidden Tech Debt

- Entanglement
- Undeclared Consumer
- Unstable Data Dependency
- Underutilized Data Dependency
- Direct Feedback Loop
- Hidden Feedback Loop

DATAMAX


Configuration
Data collection
Testing and debugging
Resource management
Serving infrastructure
Data verification
ML Code
Model analysis
Process management
Automation
Feature engineering
Metadata management
Monitoring

DATAMAX

Software Engineering has been dealing with these for a while now.

Questions so far?

DATAMAX

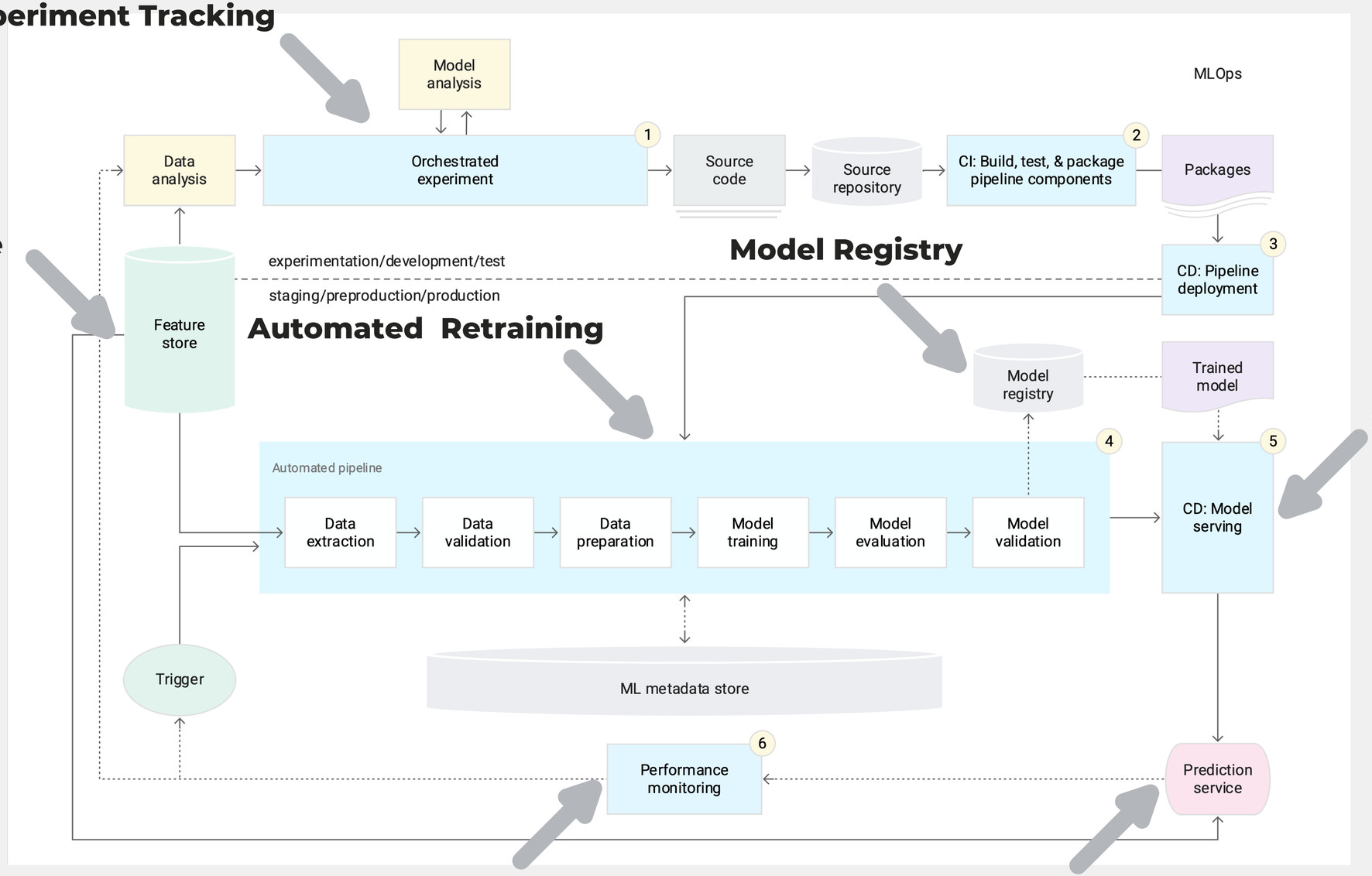
Modern Machine Learning Platforms

- **Components**
- **Deployment/Scaling**
- **Monitoring**

DATAMAX

Experiment Tracking

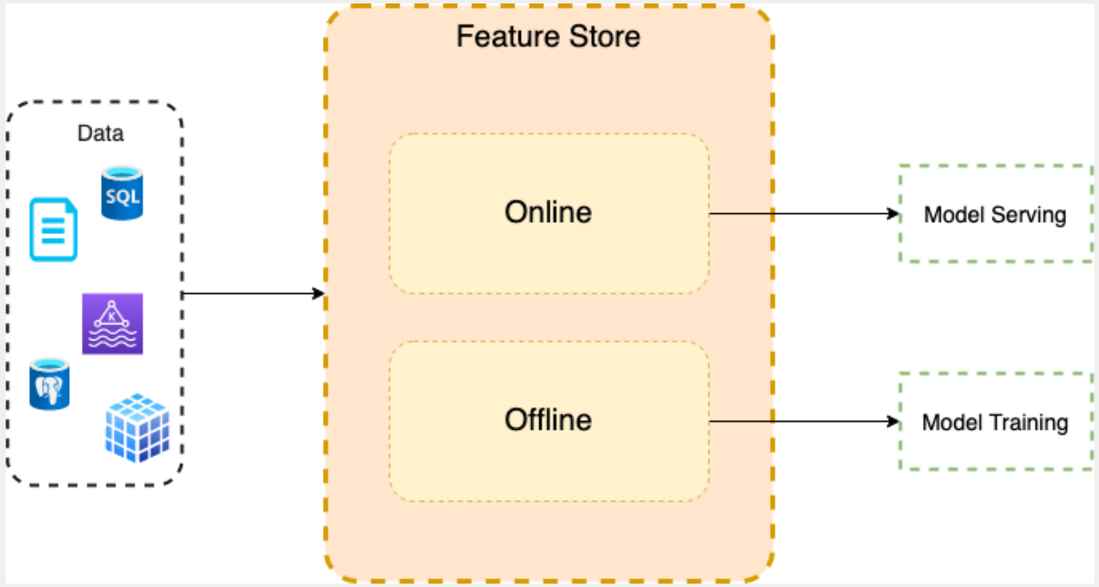
Feature Store



Monitoring

Inference

DATAMAX



ML WORKOFLOW

Feature Store

- Stores transformed data
- Offline, stored in big warehouse/database. Historic data used for batch processing and training
- Online, fast, responsive and usually only a small subset of all data. Used for data augmentation
- Data can be of different formats, or sources.

DATAMAX

Example Feature

```python
@new_feature_name(
    inputs={'name': 'input_table_name'},
    online=True,
    offline=True,
    feature_start_time=datetime(2021, 5, 20),
    batch_schedule='1d',
    ttl='30days',
    owner="name@email.ai",
    family='feature_family',
    description='A helpful description for the feature'
)
def new_feature_name_sql(inputs):
    return f'''
        SELECT
            colA,
            colB as col_B,
            colC
        FROM
            {inputs.name}
        '''
```

ML WORKOFLOW

Experiment Tracking

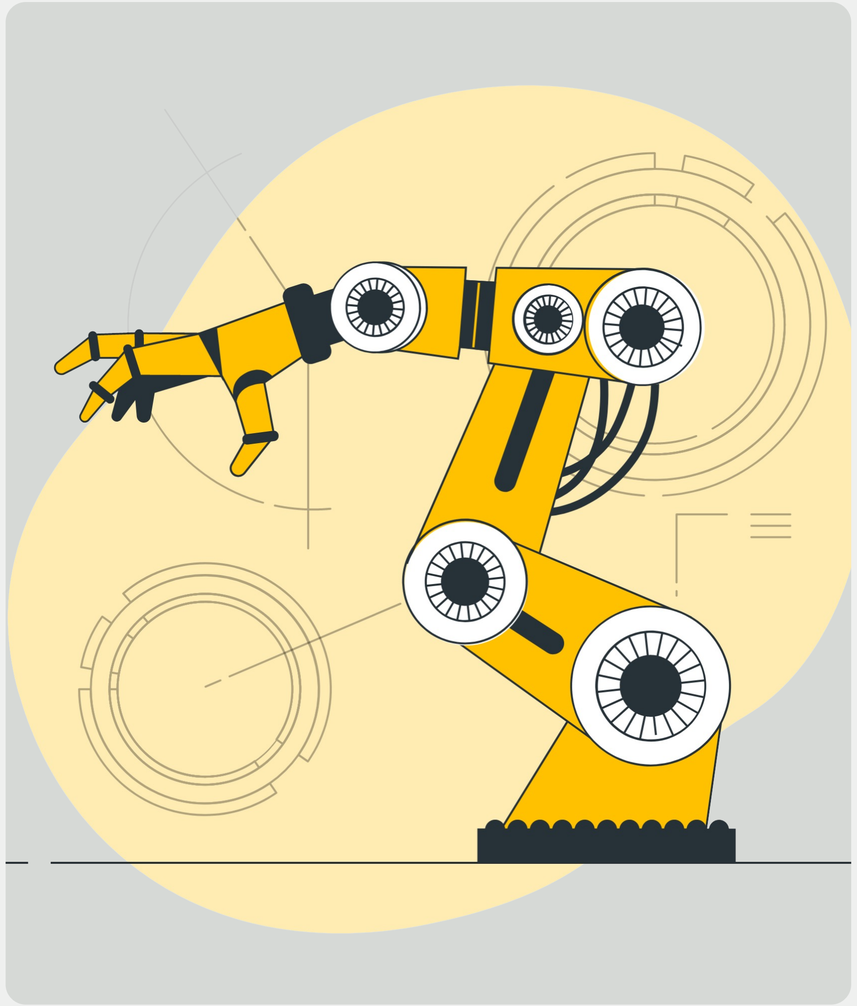
- Keep track of models and hyperparameters used
- Keep track of training and evaluation data used
- Environment configuration files
- Model weights, evaluation metrics

DATAMAX

ML WORKOFLOW

Automated Retraining

- Train and retrain models on trigger
- Metadata stored for each step in the process
- Artifacts of each step documented
- Automated checks before going in production/next stage



ML WORKOFLOW

Model Registry

- Trained model artifacts are stored
- Centralized storage
- Metadata of model
- Source for production model deployment

ML WORKOFLOW

Model Serving

- Serve model as API
- Take the latest version without downtime
- Automated scaling

ML WORKOFLOW

Model Monitoring

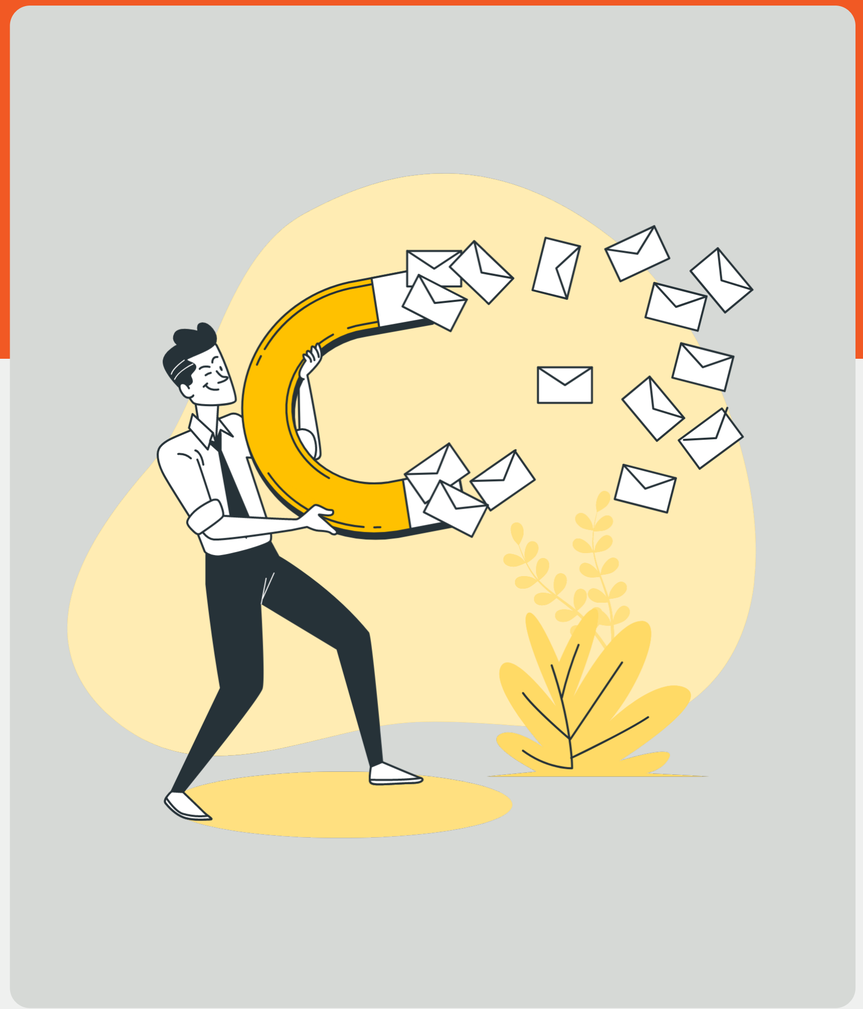
- Check for Drift
  - Training data not representing reality
  - Relationship between input and prediction changes
- Check for performance issues
- Monitor for bugs
- Monitor for fairness

Questions so far?

What was this all about?

Let’s go back to the diagram!

DATAMAX



Use Case – Email Classification

- **Enron Data Set**
- **Supervised Learning**
- **Naïve Bayes**

Email Classifier

Based on the content of an e-mail define on which department to forward it.

- CountVectorizer & Tfidf Transformation
- Manually labeled data from Enron dataset
- Naïve Bayes Estimator
- Not great accuracy but that is not the point

```
pipeline = Pipeline([
    ("vect", CountVectorizer()),
    ("tfidf", TfidfTransformer()),
    ("clf", MultinomialNB())
])

model = pipeline.fit(X_train, y_train)
```

DATAMAX

Email Classifier

- Serve model in a service with API

```python
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/predict', methods=['POST'])
def predict():
    message = req.get("message")
    logging.info(f"Received message: {message}")

    prediction = model.predict(message)
    logging.info(f"prediction: {prediction}")

    return {"label": prediction}
```

DATAMAX

Email Classifier

- Containerize the service

```
FROM python:3.8.10

WORKDIR /app

COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt

COPY app.py app.py
COPY model.pkl model.pkl

EXPOSE 5000

ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0

CMD ["flask", "run"]
```

DATAMAX

Email Classifier

- Build image
- Add to cluster
- Deploy and scale

```
# Build Docker image
docker build -t email_classifier_image .
# Add image to cluster
minikube image load email_classifier_image

# create kubernetes resources
kubectl apply -f deployment.yaml

# scale deployment
kubectl scale deploymemnt email-classifier-deployment
-n email-classifier-namespace --replicas=3
```

DATAMAX

Email Classifier

- Kubernetes resources

```
# Build Docker image
docker build -t email_classifier_image .
# Add image to cluster
minikube image load email_classifier_image

# create kubernetes resources
kubectl apply -f deployment.yaml

# scale deployment
kubectl scale deploymemnt email-classifier-deployment
-n email-classifier-namespace --replicas=3
```

DATAMAX

Email Classifier

- Kubernetes resources

```yaml
kind: Namespace
apiVersion: v1
metadata:
  name: email-classifier-namespace
  labels:
    name: email-classifier
---
apiVersion: v1
kind: Service
metadata:
  name: flask-service
  namespace: email-classifier-namespace
spec:
  selector:
    app: email-classifier
  ports:
    - protocol: "TCP"
      port: 5000
      targetPort: 5000
  type: ClusterIP

```

```yaml
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: email-classifier-deployment
  namespace: email-classifier-namespace
spec:
  selector:
    matchLabels:
      app: email-classifier
  replicas: 1
  template:
    metadata:
      labels:
        app: email-classifier
    spec:
      containers:
        - name: email-classifier
          image: email-classifier:latest
          imagePullPolicy: Never
          ports:
            - containerPort: 5000
```

THANK YOU!

- data-max.io
- info@data-max.io
- Linkedin.com/company/datamax-io
- bujar@data-max.io