

# Production: Model Deployment

```
$ echo "Data Science Institute"
```

# Introduction

# Agenda

## 6.1 Model Deployment and Prediction Service

- ML Deployment Myths and Anti-Patterns
- Batch Prediction vs Online Prediction

## 6.2 Explainability Methods

- Partial Dependence Plots
- Permutation Importance
- Shap Values

# Slides, Notebooks, and Code

- These notes are based on Chapter 7 of *Designing Machine Learning Systems*, by [Chip Huyen](#).

## Notebooks

- `./01_materials/labs/production_5_model_development.ipynb`

## Code

- `./05_src/credit_experiment_*.py`

# Our Reference Architecture

# The Flock Reference Architecture

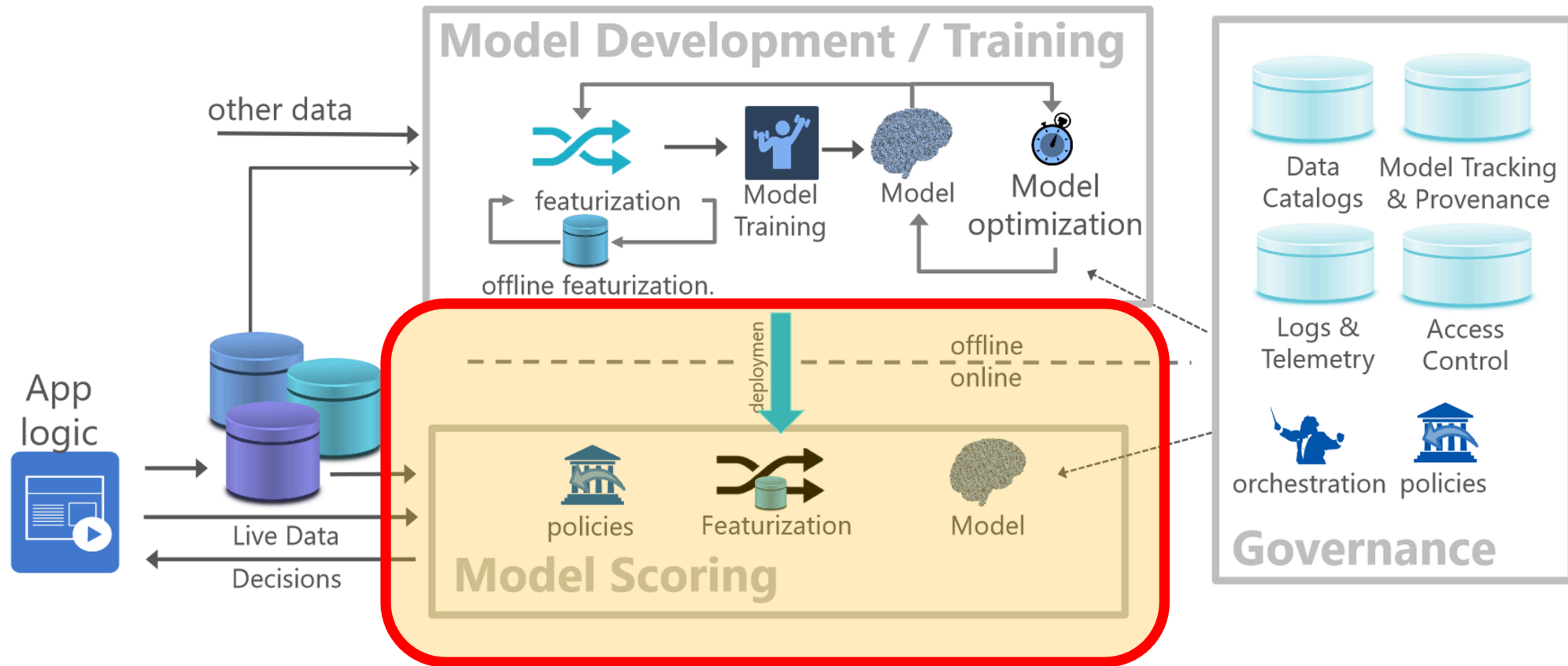


Figure 1: Flock reference architecture for a canonical data science lifecycle.

# Deployment

# Deployment

- Deploying a model is to make it useable by allowing users to interact with it through an app or by using its results for a purpose in a data product (BI visuals, reports, data views).
- Deployment is a transition of development to a production environment.
- There is a wide range of production environments, from BI to live applications serving millions of users.
- Engaging with users in formal or informal feedback conversations is helpful, although only sometimes possible.



# Deployment Myths and Anti-Patterns

## 1. You only deploy one or two ML models at a time

- Infrastructure should support many models, not only a few.
- Many models can interact, and we also need a way of mapping these interactions.
- Ride sharing app:
  - 10 models: ride demand, driver availability, estimated time of arrival, dynamic pricing, fraud, churn, etc.
  - 20 countries.

## 2. If we don't do anything, model performance stays the same

- Software does not age like fine wine.
- Data distribution shifts: when the data distribution in the trained model differs from the distribution during testing.

# Deployment Myths and Anti-Patterns (cont.)

## 3. You won't need to update your models as much

- Model performance decays over time.
- Deploy should be easy:
  - The development environment should resemble the production environment as closely as possible.
  - Infrastructure should be easier to rebuild than to repair.
  - Small incremental and frequent changes.

## 4. Most ML engineers don't need to worry about scale

- Scale means different things to different applications.
- Number of users, availability, speed or volume of data.

# Batch Prediction Vs Online Prediction

## Online Prediction

- Predictions are generated and returned as soon as requests for these predictions arrive.
- Also known as on-demand prediction.
- Traditionally, requests are made to a prediction service via a RESTful API.
- When requests are made via HTTP, online prediction is known as *synchronous prediction*.

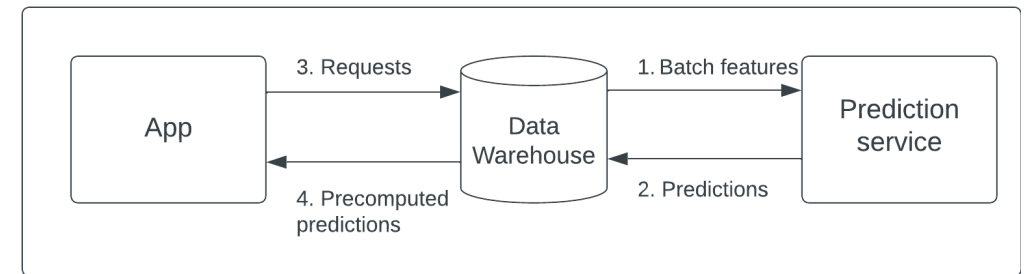
## Batch Prediction

- Predictions are generated periodically or whenever triggered.
- Predictions are stored in SQL tables or in memory. They are later retrieved as needed.

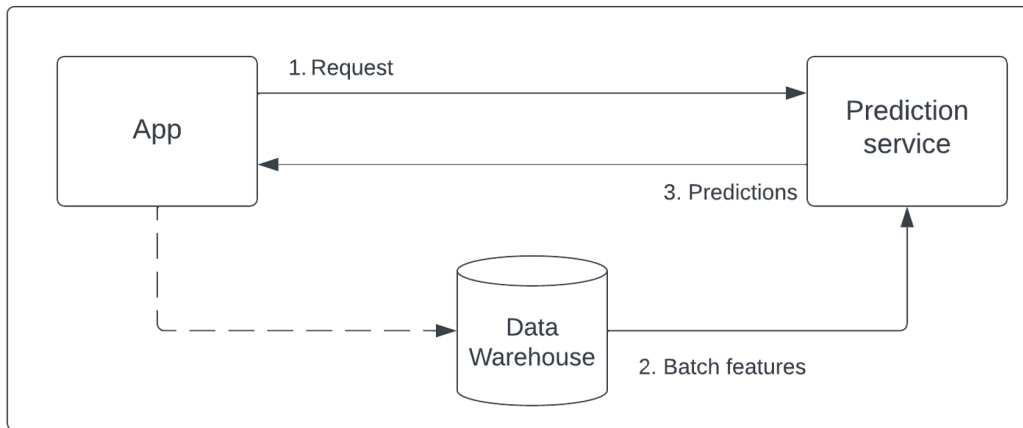
# Model Prediction Service

Three types of model prediction or inference service:

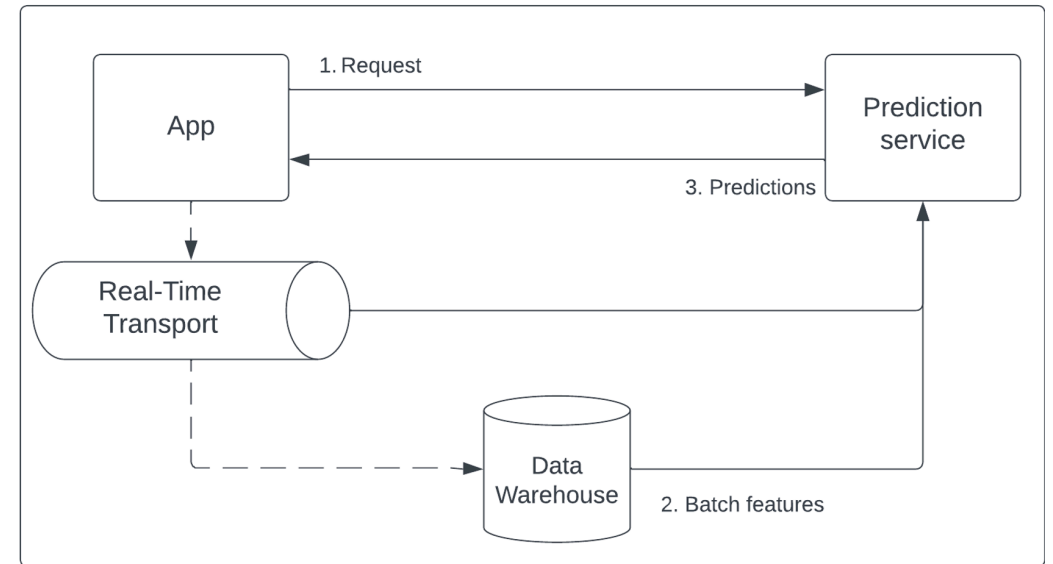
- Batch prediction: uses only batch features.
- Online prediction that uses only batch features (e.g., precomputed embeddings).
- Online streaming prediction: uses batch features and streaming features.



# Model Prediction Service (cont.)



Online Prediction (based on Huyen 2021)



Streaming Prediction (based on Huyen 2021)

# References

- Agrawal, A. et al. "Cloudy with a high chance of DBMS: A 10-year prediction for Enterprise-Grade ML." arXiv preprint arXiv:1909.00084 (2019).
- Huyen, Chip. "Designing machine learning systems." O'Reilly Media, Inc.(2021).