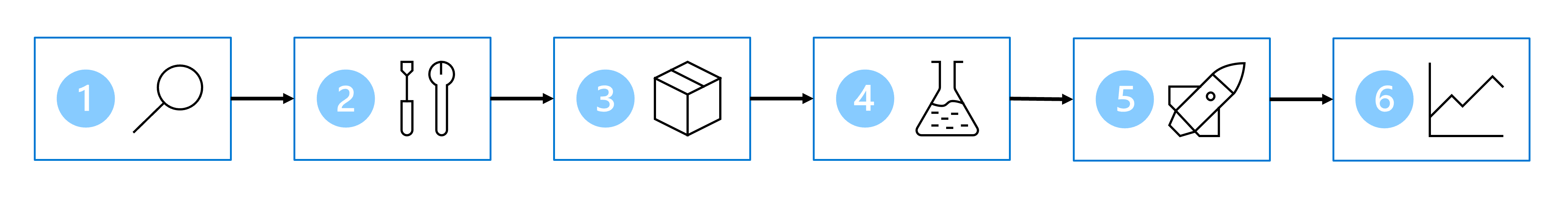
Module 3: Get started w/ ML in Azure

# ML Framework

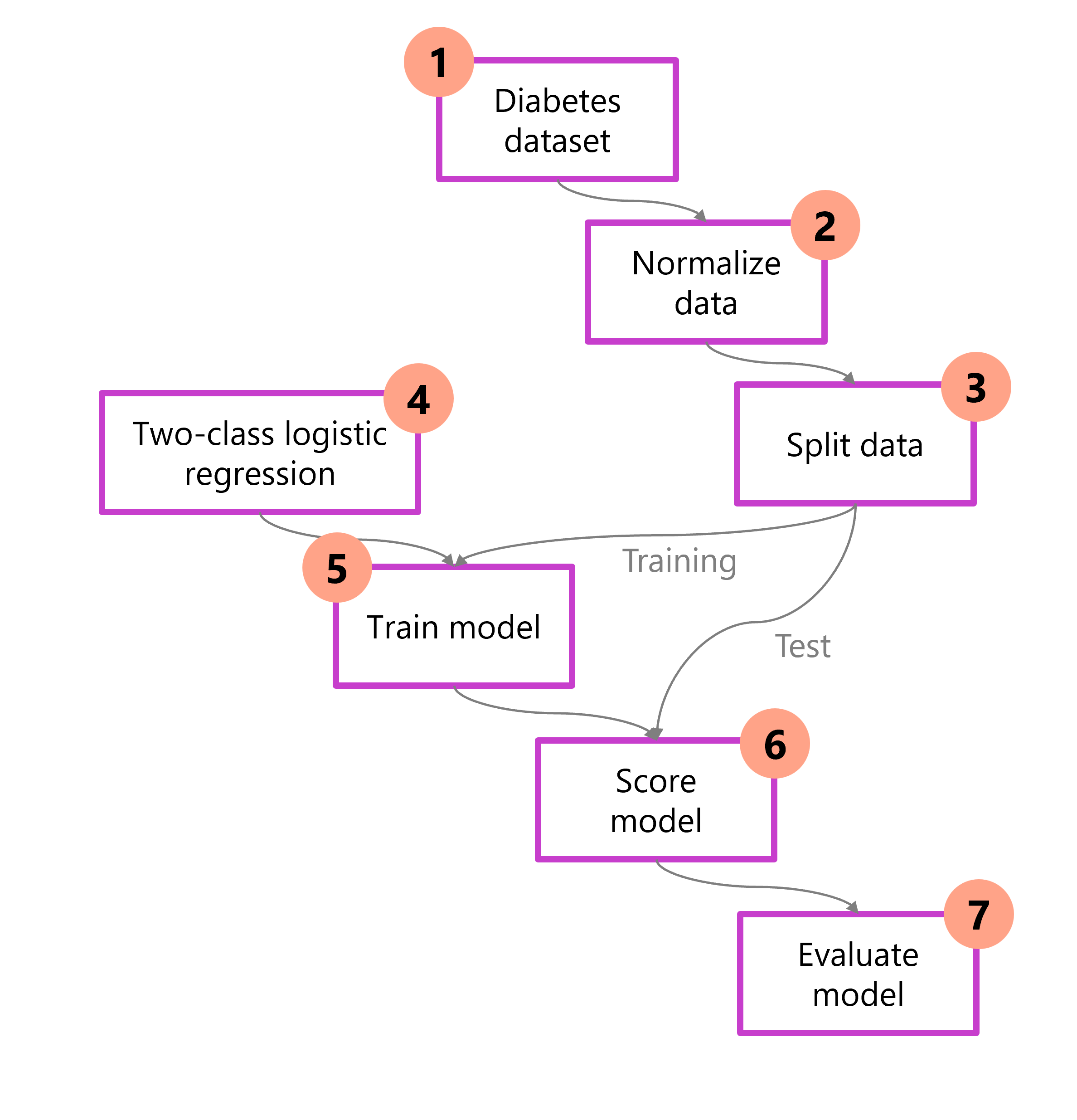


1. **Define the problem**: Decide on what model should predict and when it's successful.
2. **Get the data** from data sources.
3. **Prepare the data** through ETL based on the model's requirements.
4. **Train the model** by choosing right algorithm and hyper-parameter values based on trial and error.
5. **Integrate the model**: Deploy the model to an endpoint to generate predictions.
6. **Monitor the model’s** performance.

# Define the Problem

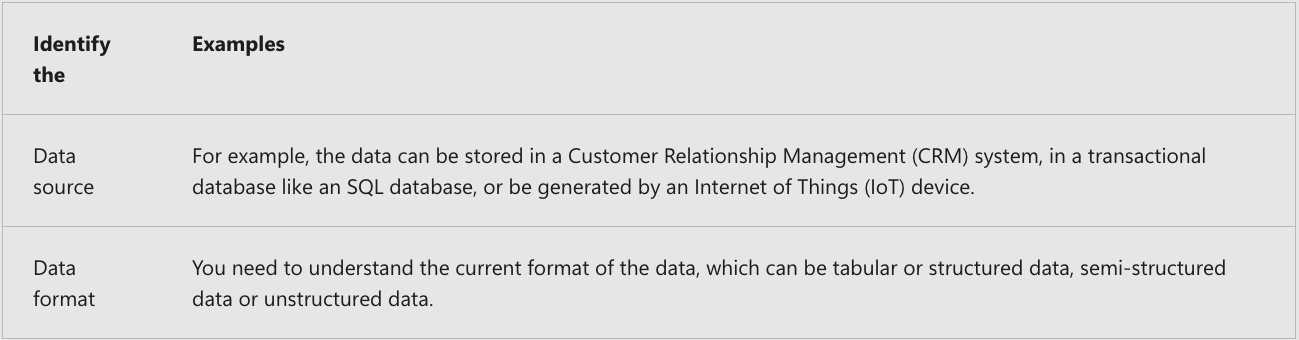
* **Criteria**:
  + What the model’s output should be.
  + What type of machine learning task you use.
  + What criteria make a model successful.
* Use Cases:
  + **Classification**: Predict a categorical value.
  + **Regression**: Predict a numerical value.
  + **Time-series forecasting**: Predict future numerical values based on time-series data.
  + **Computer vision**: Classify images or detect objects in images.
  + **Natural language processing** (NLP): Extract insights from text.

**Example**: *Determine if Patients have Diabetes*

* **Framework**: *Problem you're trying to solve and the type of data available determines the machine learning task you choose*.
  + In this case, the available data are other health data points from patients. We can represent the output we want as categorical information that either the patient has diabetes or doesn't have diabetes. Thus, the machine learning task is classification.
* 
  1. **Load data**: Import / inspect the dataset.
  2. **Preprocess data**: Normalize / clean.
  3. **Split data**: Separate into training and test sets.
  4. **Choose model / Algorithm**
  5. **Train model** to learn patterns from the training data.
  6. **Score model**: Generate predictions in test.
  7. **Evaluate**: Calculate performance metrics.

# Get and Prep Data

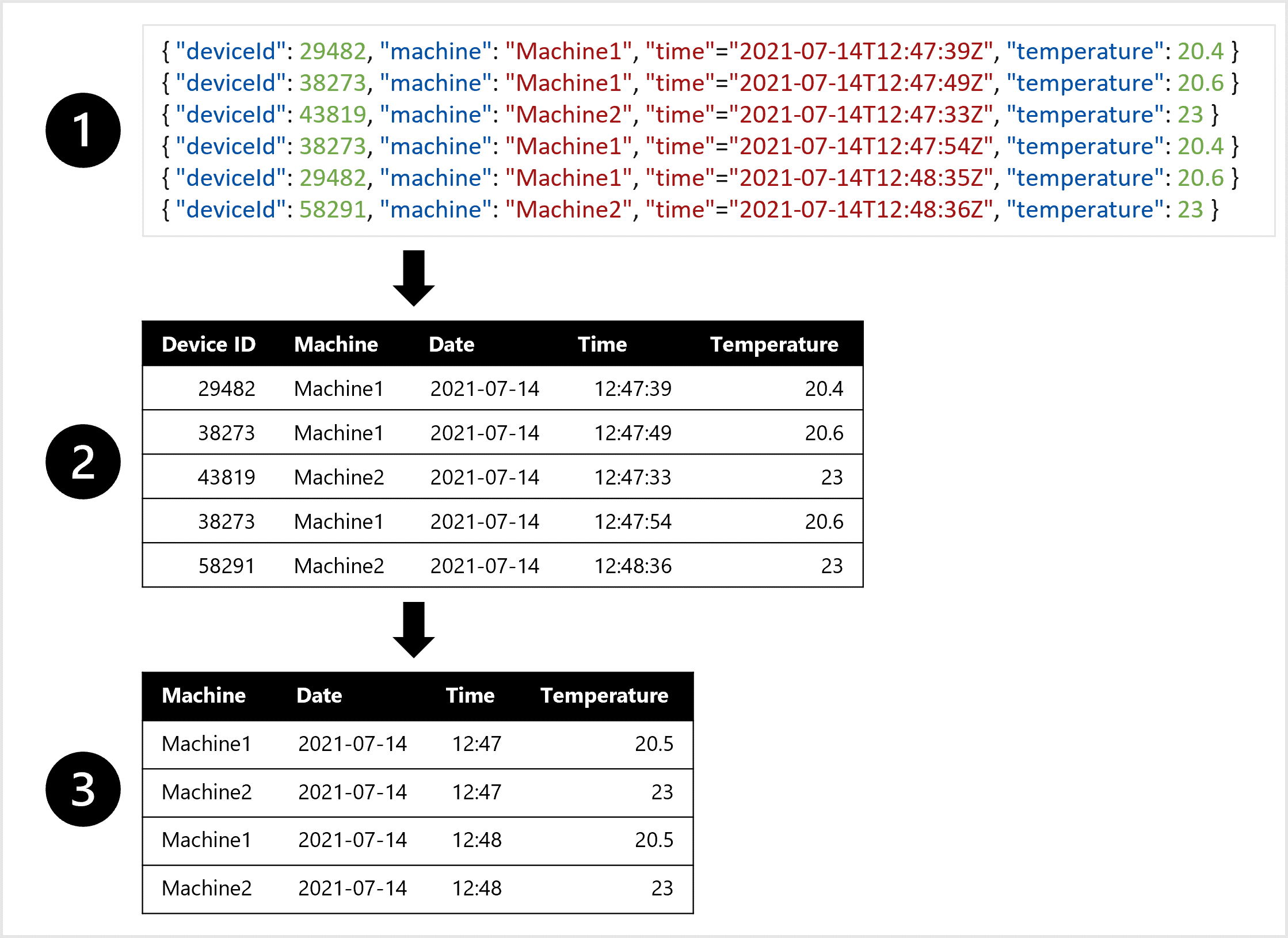
## Identity Data Source/Format

* 

# Design a data ingestion solution

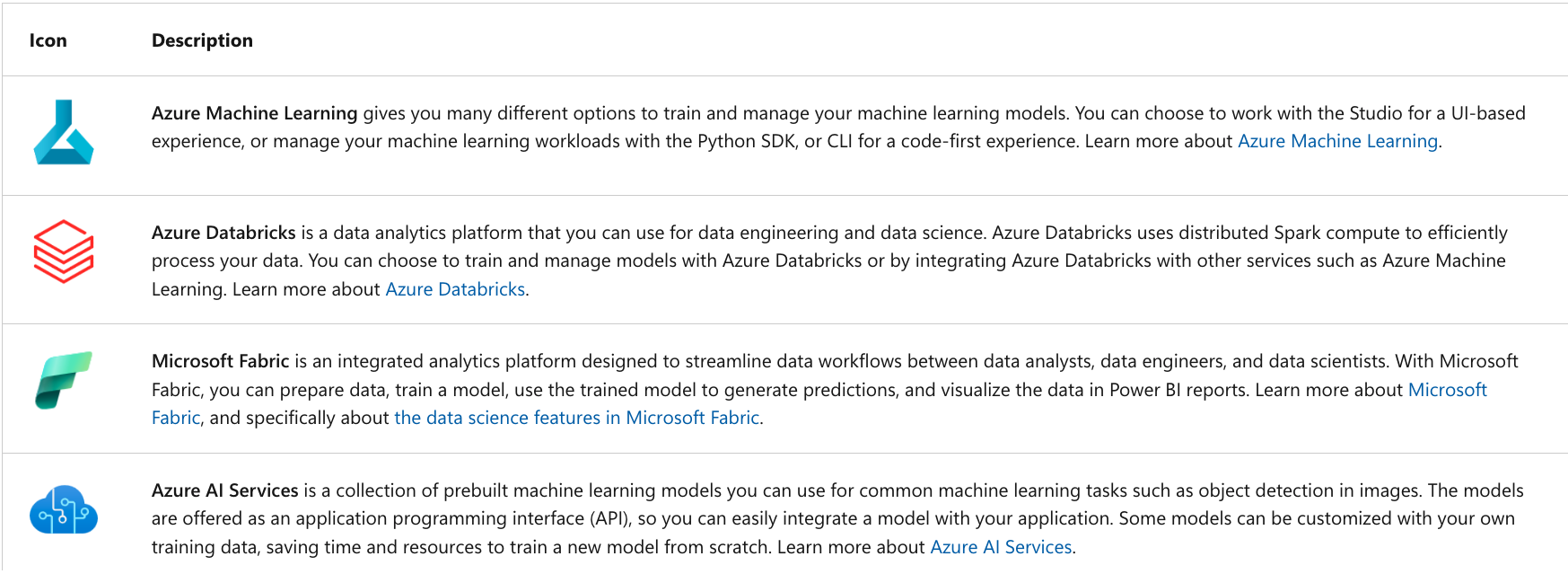
* Use Data Ingestion Pipeline to Extract, Transform, and Load (ETL or ELT).
  + Such pipelines can be created with Azure Synapse Analytics, Azure Databricks, and Azure ML.
  + A common approach for a data ingestion solution is to:
    - **Extract** raw data from its source (like a CRM system or IoT device).
    - **Copy** and **transform** the data with Azure Synapse Analytics.
    - **Store** the prepared data in an Azure Blob Storage.
    - **Train** the model with Azure Machine Learning.

**Example**: *Train a weather forecasting model*.

* 
* You can create a dataset for training the forecasting model by:
  + **Extract** data measurements as JSON objects from the IoT devices.
  + **Convert** the JSON objects to a table.
  + **Transform** the data to get the temperature per machine per minute.

# Train the model

Choose Azure Service based on:

* What type of model you need to train,
* Whether you need full control over model training,
* How much time you want to invest in model training,
* Which services are already within your organization,
* Which programming language you’re comfortable with.
* 

## Features and capabilities of Azure Machine Learning

* Azure ML Supports Tasks:
  + Exploring data and preparing it for modeling.
  + Training and evaluating machine learning models.
  + Registering and managing trained models.
  + Deploying trained models for use by applications and services.
  + Reviewing and applying responsible AI principles and practices
* Azure ML features and capabilities:
  + Centralized storage and management of datasets for model training and evaluation.
  + On-demand compute resources on which you can run machine learning jobs, such as training a model.
  + Automated machine learning (AutoML), which makes it easy to run multiple training jobs with different algorithms and parameters to find the best model for your data.
  + Visual tools to define orchestrated pipelines for processes such as model training or inferencing.
  + Integration with common machine learning frameworks such as MLflow, which make it easier to manage model training, evaluation, and deployment at scale.
  + Built-in support for visualizing and evaluating metrics for responsible AI, including model explainability, fairness assessment, and others.