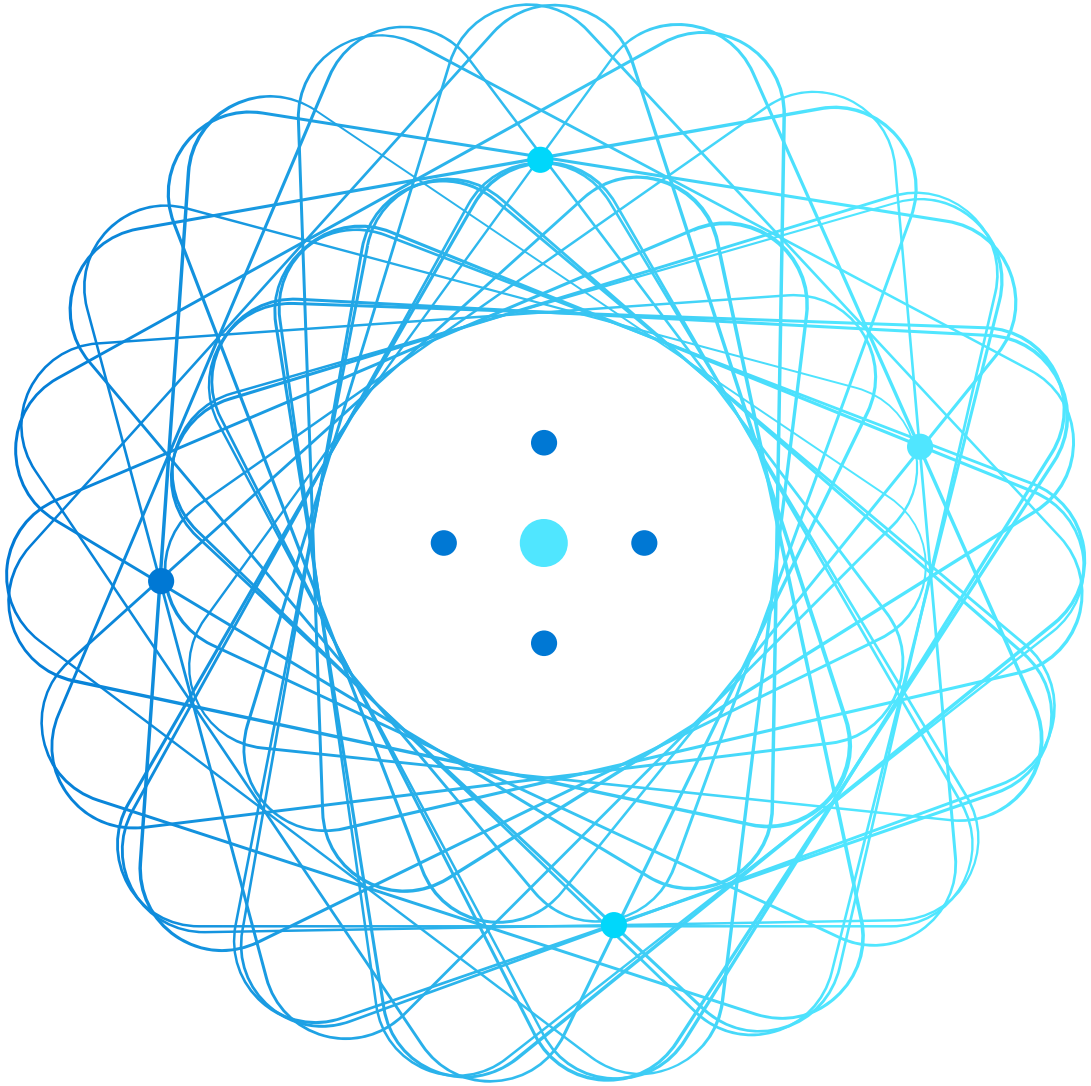
Microsoft Azure

Design a machine learning solution



Module Agenda

Design a data ingestion strategy for machine learning projects

Design a machine learning model training solution

Design a model deployment solution

Design a data ingestion strategy for machine learning projects

Identify your data source and format

Identify the data source:

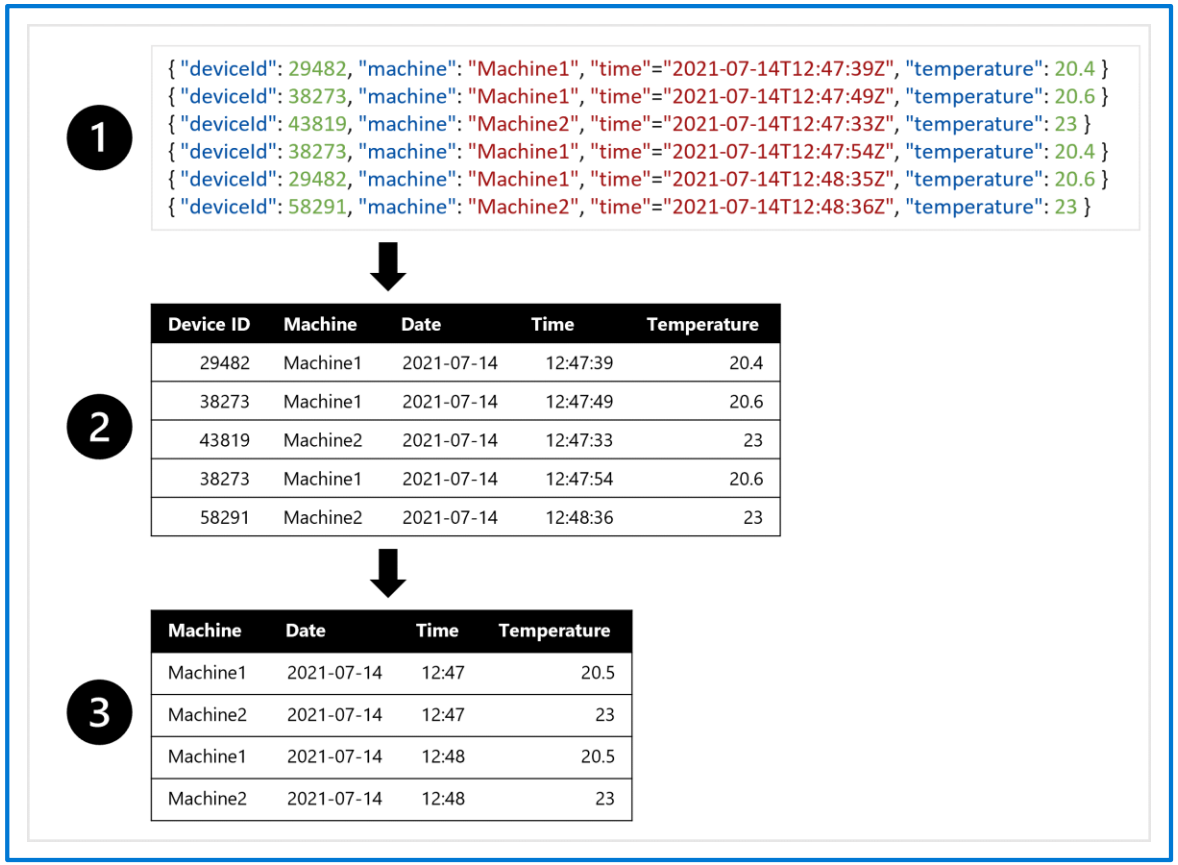
- First identify where the data you want to use is stored.

Identify the current data format:

- Tabular or structured data
- Semi-structured data
- Unstructured data

Identify the desired data format:

- Transform the data to change the data format and make it more suitable for model training.



Choose how to serve data to machine learning workflows

Separate compute from storage:

One of the benefits of the cloud is the ability to scale compute up or down according to your demands.

In addition, you can shut down compute when you don’t need it and restart it when you want to use it again.

Store data for model training workloads:

When you use Azure Machine Learning, Azure Databricks, or Azure Synapse Analytics for model training, there are three common options for storing data:

- Azure Blob Storage
- Azure Data Lake Storage (Gen 2)
- Azure SQL Database

Design a data ingestion solution

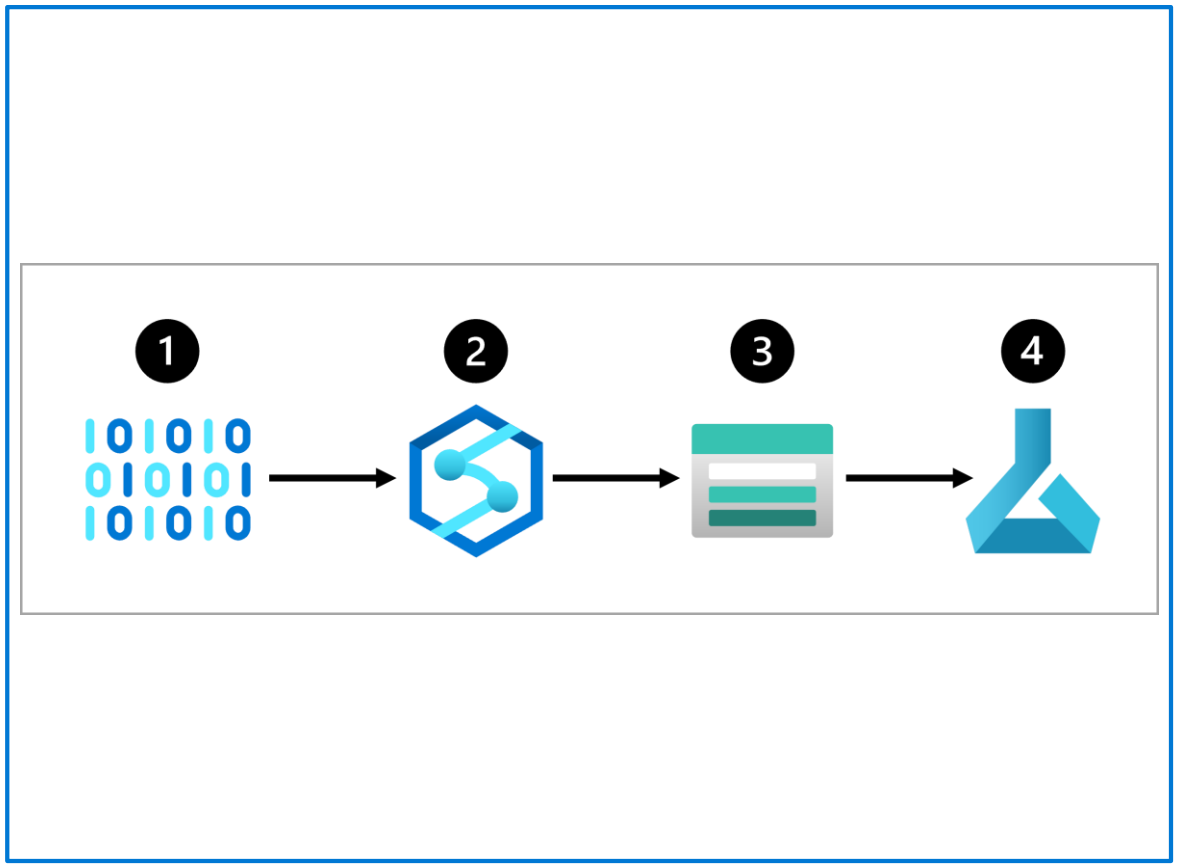
Create a data ingestion pipeline:

- Use Azure Synapse Analytics, Azure Databricks or Azure Machine Learning.

Design a data ingestion solution:

A common approach for a data ingestion solution is to:

1. Extract raw data from its source (like a CRM system or IoT device).
2. Copy and transform the data with Azure Synapse Analytics.
3. Store the prepared data in an Azure Blob Storage.
4. Train the model with Azure Machine Learning.



Case study - Design a data ingestion strategy（1/3）

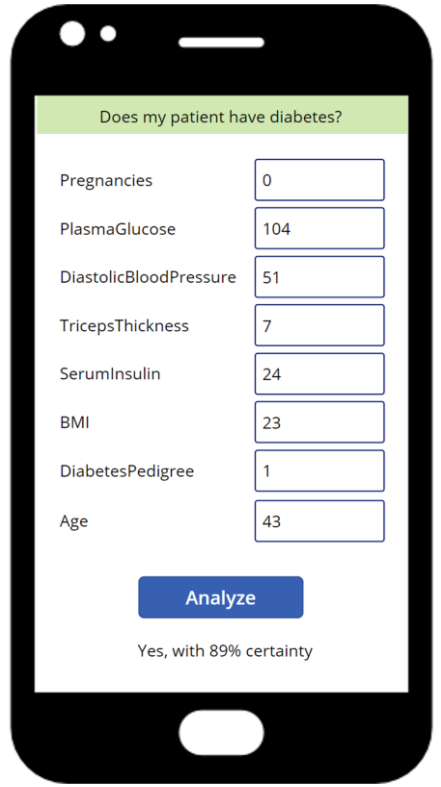
Welcome to Proseware! You've been hired as the lead data scientist to help us design a data ingestion solution.

At Proseware, we're developing a mobile application that will help doctors diagnose diseases in patients faster. A doctor can enter the patient's medical data into the app to get a diagnosis on the patient.

Our first planned feature is that the app will tell the doctor whether the patient should be further screened or treated for diabetes.

We need your help deciding how to extract, load, and transform the data we need to train a model.

We're looking forward to your advice on how to design the data ingestion solution!



Case study - Design a data ingestion strategy（2/3）

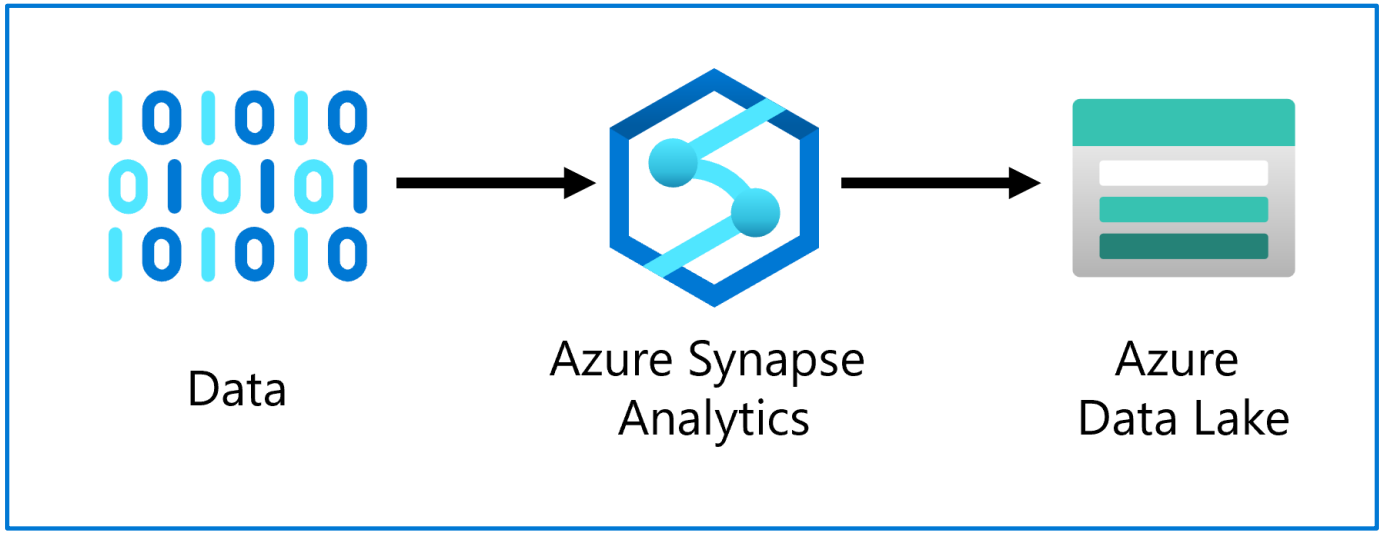
Consider the requirements:

- **Consider the current data type**: We have already collected data that correlates with diabetes, such as the number of pregnancies, age, and BMI in a patient database.
- **Consider the desired data type**: Our data scientists are used to working with Python and want the data as CSV files.
- **Consider the data access**: We want our design to be future-proof and ready for scale. We want to extract the privacy-sensitive data from the patient database and store the data in an Azure data storage solution.

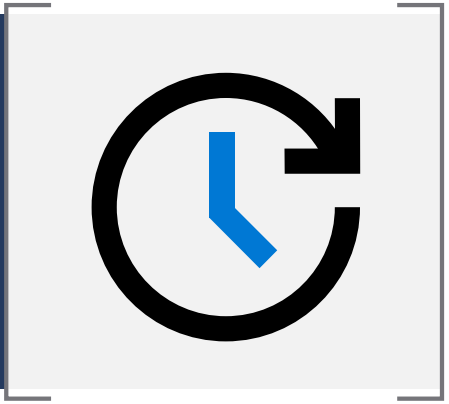
Case study - Design a data ingestion solution (3/3)

Which storage solution would you recommend to store the data?

Which tool would you recommend we use to move the data?



Design a machine learning model training solution



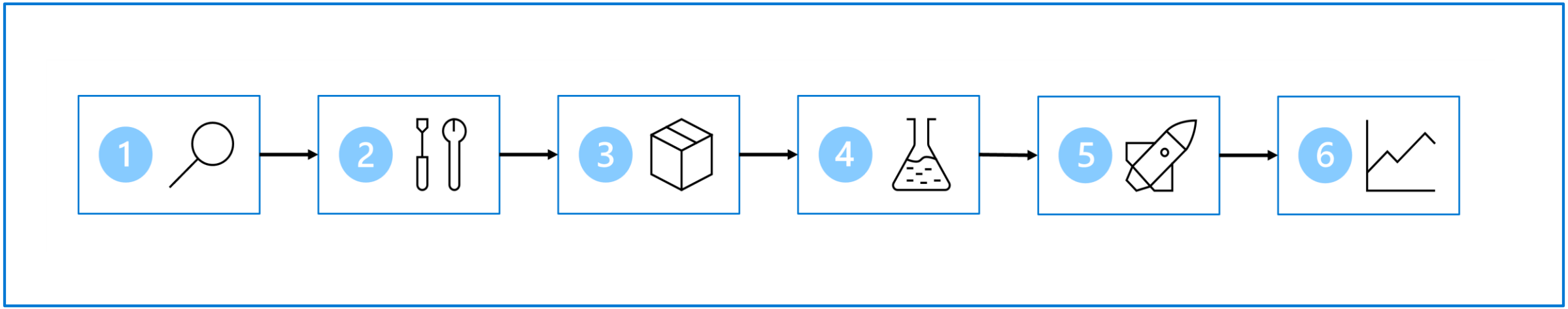
Introduction

In this module, you'll learn how to:

- Identify machine learning tasks.
- Choose a service to train a model.
- Choose between compute options.

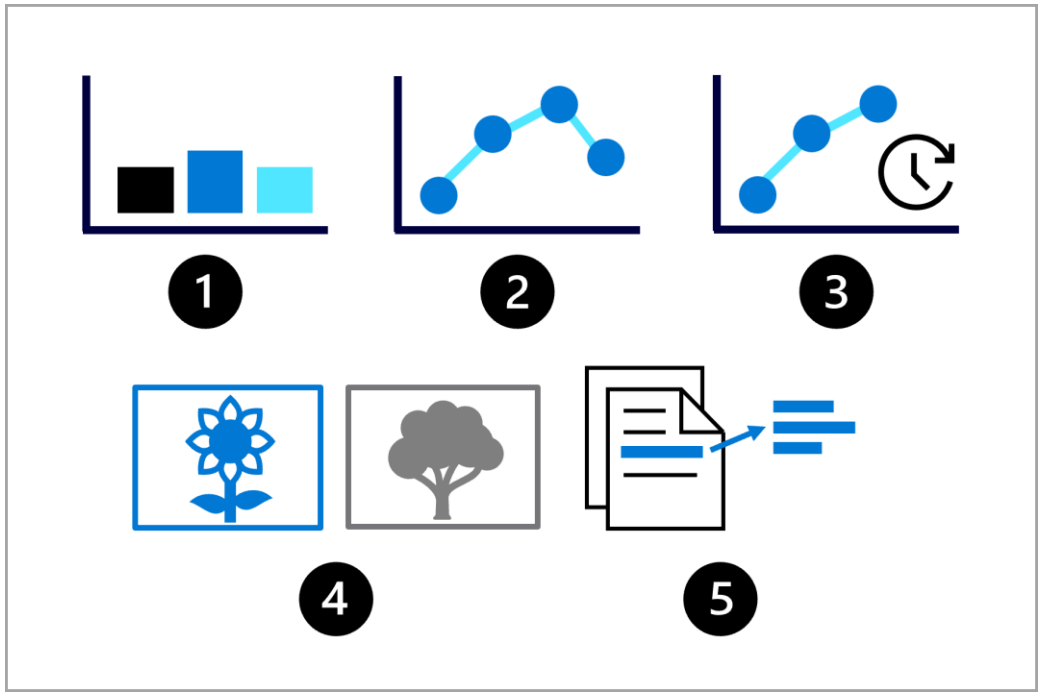
Understand the machine learning process

1. Define the problem
2. Get the data
3. Prepare the data
4. Train the model
5. Integrate the model
6. Monitor the model.



Identify machine learning tasks

1. Classification
2. Regression
3. Time-series forecasting
4. Computer vision
5. Natural language processing (NLP)



Choose a service to train a machine learning model

Understand the difference between services:

- Azure Cognitive Services
- Azure Synapse Analytics
- Azure Databricks
- Azure Machine Learning




Choose a service to train a machine learning model

Azure Cognitive Services

Customize or consume prebuilt models.

Save time and effort.

Azure Synapse Analytics

One platform for all data engineering and data science projects at scale.

Offers an easy to use UI, notebooks, and scripts.

Azure Databricks

Use notebooks for data engineering and data science at scale.

Offers distributed compute (PySpark).

Azure Machine Learning

Manage machine learning models from development, to testing, to production.

Use Python in notebooks and scripts.

Decide between compute options

CPU or GPU

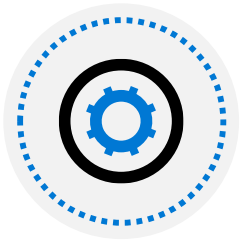
CPU will be sufficient and cheaper to use for smaller tabular datasets

Whenever working with unstructured data like images or text, GPUs will be more powerful and effective.

General purpose or memory optimized

General purpose: Have a balanced CPU-to-memory ratio. Ideal for testing and development with smaller datasets.

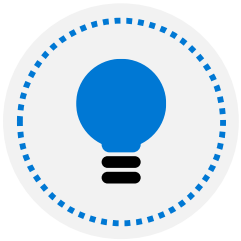
Memory optimized: Have a high memory-to-CPU ratio. Great for in-memory analytics.



Spark

Services like Azure Synapse Analytics and Azure Databricks offer Spark compute.

A Spark cluster consists of a driver node and worker nodes.



Monitor the compute utilization

Configuring your compute resources for training a machine learning a model is an iterative process.

Case study - Design a model training strategy（1/3）

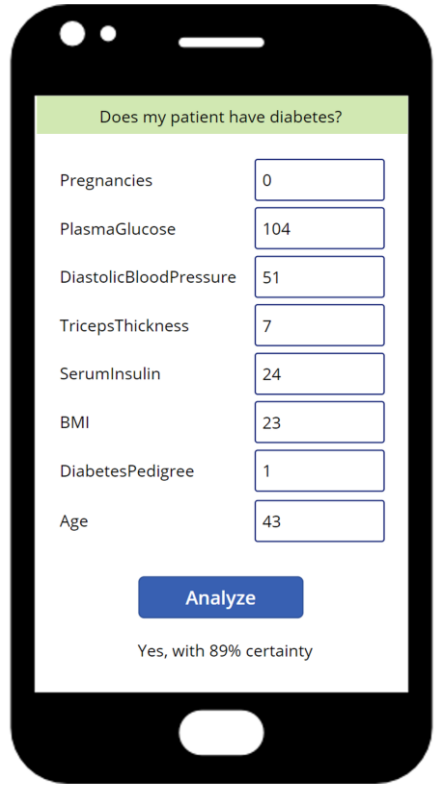
Welcome to Proseware! You've been hired as the lead data scientist to help us design a model training solution.

At Proseware, we're developing a mobile application that will help doctors diagnose diseases in patients faster. A doctor can enter the patient's medical data into the app to get a diagnosis on the patient.

Our first planned feature is that the app will tell the doctor whether the patient should be further screened or treated for diabetes.

We need your help deciding how to train a model which can detect diabetes.

We're looking forward to your advice on how to design the model training solution!



Case study - Design a model training strategy（2/3）

Consider the requirements:

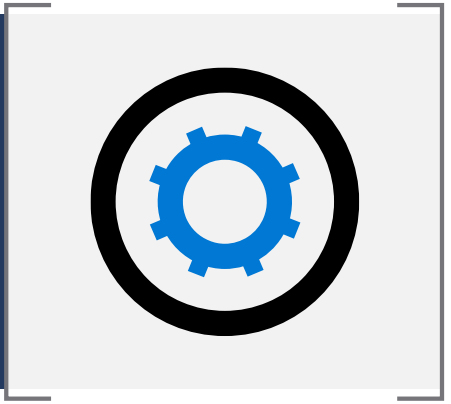
- **Consider the team**: We have a team of data scientists. They should all be able to train a classification model. They are used to working with Python and have no experience with SQL or Spark.
- **Consider the preferred tooling**: We prefer not to use a UI. We want our data scientists to train the model with notebooks and scripts. When we get audited, we need to show exactly how a model is trained. We also want our data scientists to have full control over how a model is trained.
- **Consider the compute**: We want our data scientists to get started with Jupyter notebooks, which is what they're familiar with.

Case study - Design a model training solution (3/3)

How should we train the model to predict diabetes?

Which compute would you recommend to train the model?

Design a model deployment solution



Introduction

In this module, you'll learn how to:

- Understand how a model will be consumed.
- Decide whether to deploy your model to a real-time or batch endpoint.

Understand how model will be consumed

Deploy a model to an endpoint

When you train a model, the goal is often to integrate the model into an application.

To easily integrate a model into an application, you can use endpoints.

When you deploy a model to an endpoint, you have two options:

Get real-time predictions:

If you want the model to score any new data as it comes in, you need predictions in real-time.

Real-time predictions are often needed when a model is used by an application such as a mobile app or a website.

Get batch predictions:

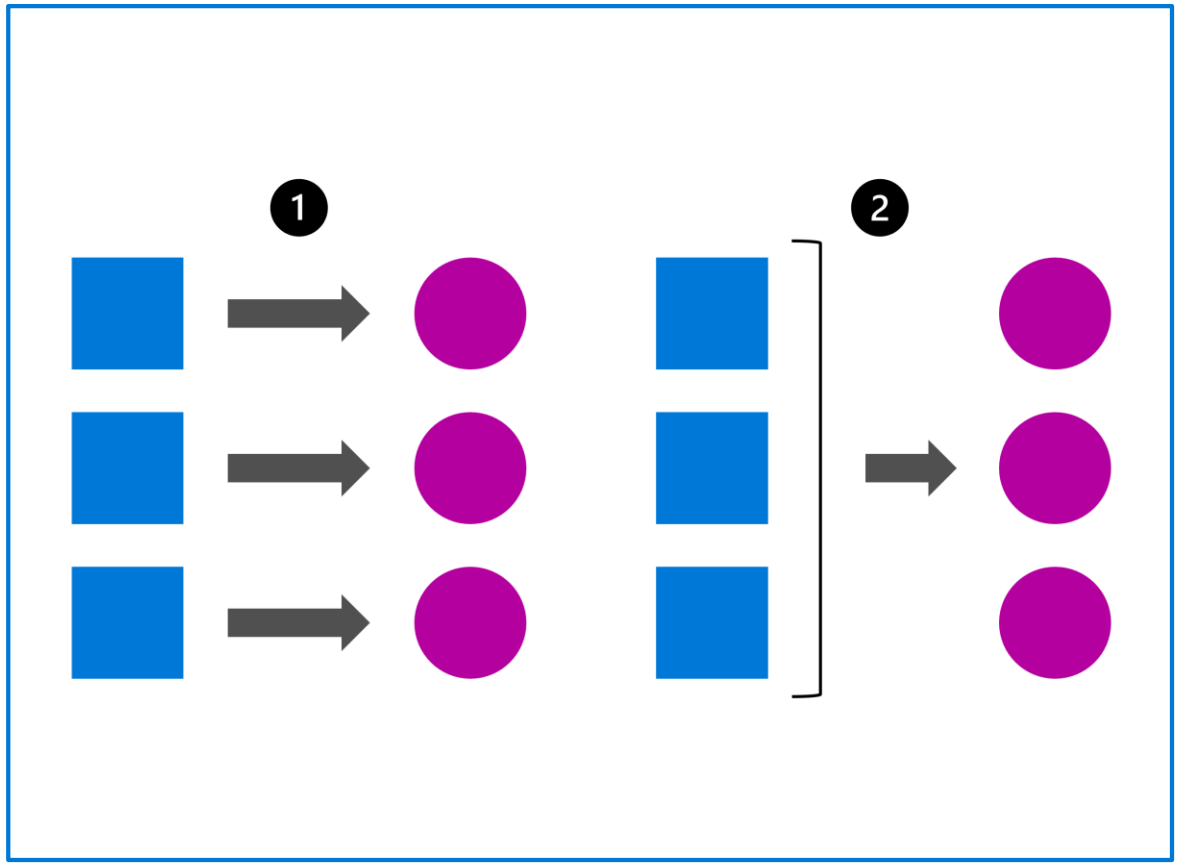
If you want the model to score new data in batches and save the results as a file or in a database, you need batch predictions.

Decide on real-time or batch deployment（1/2）

Identify the necessary frequency of scoring:

Generally, there are two types of use cases:

1. You need the model to score the new data as soon as it comes in.
2. You can schedule or trigger the model to score the new data that you've collected over time.



Decide on real-time or batch deployment（2/2）

Decide on the number of predictions:

To ask yourself is whether you need the predictions to be generated individually or in batches.

Consider the cost of compute:

To decide whether to deploy your model to a real-time or batch endpoint, you must consider the cost of each type of compute.

Decide on real-time or batch deployment:

In general, if you need individual predictions immediately when new data is collected, you need real-time predictions.

If you need the model to score new data when a batch of data is available, you should get batch predictions.

Case study - Design an model deployment solution（1/3）

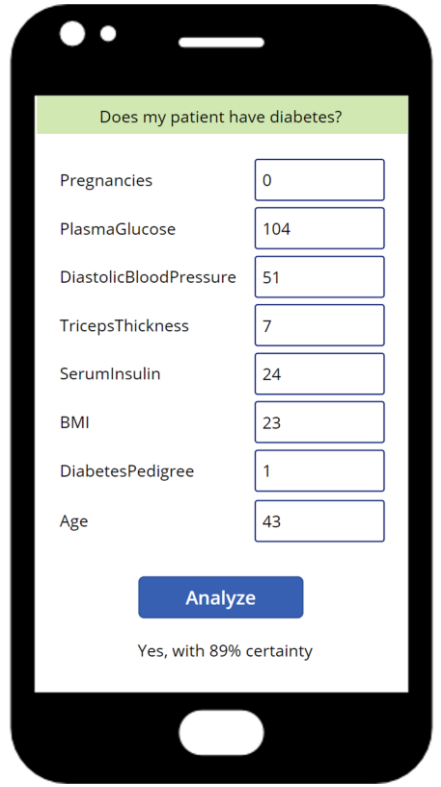
Welcome to Proseware! You've been hired as the lead data scientist to help us design a machine learning deployment solution.

At Proseware, we're developing a mobile application that will help doctors diagnose diseases in patients faster. A doctor can enter the patient's medical data into the app to get a diagnosis on the patient.

Our first planned feature is that the app will tell the doctor whether the patient should be further screened or treated for diabetes.

We need your help deciding how to deploy the model to integrate it with our mobile application.

We're looking forward to your advice on how to design the model's deployment solution!



Case study - Design a model deployment solution（2/2）

Consider the requirements:

- **Consider the frequency**: The plan is that a doctor enters a patient's information into the app, like their age and BMI. After entering, a doctor can select the Analyze button, after which the model should predict whether a patient is likely to have diabetes.
- **Consider the compute**: A doctor consultation typically takes less than 10 minutes. If we want doctors to use this app, we need the answers to be returned as quickly as possible. The deployed model should always be available as we don't know when a doctor may use it.
- **Consider the size**: A doctor will only use the app to get a prediction on an individual's situation. There's no need for generating the predictions of multiple patients at once.

Case study - Design an model deployment solution (3/3)

What type of predictions are needed by the mobile application?

What kind of compute should be used by the deployed model?

Microsoft Azure