

Get started with data science in Microsoft Fabric

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/get-started-data-science-fabric/1-introduction>

Introduction

Completed

Imagine you're working for a supermarket and you want to know how much bread you need to have in stock every week to meet customer demands while also avoiding food waste.

Or maybe you want to analyze your customers to understand how to best target them with personalized offers.

Whenever you want to make informed decisions within an organization, you can use data science to get insights from the data you have. **Data science** is a combination of mathematics, statistics, and computer engineering.

When you perform data science, you can analyze your data and identify *complicated patterns* that can provide you with meaningful insights for your organization. You can use data science to create **artificial intelligence (AI)** models that encompass the complicated patterns you find in your data. A common approach is to use data science to train **machine learning** models using libraries like `scikit-learn` in Python to achieve AI.

Taking a data science project from beginning to end can be a challenge. Microsoft Fabric offers one workspace to manage an end-to-end data science project.

In this module, you learn about a typical data science project. Additionally, you explore which Microsoft Fabric features you can use for each part of the data science process.

2. Understand the data science process

Understand the data science process

Completed

A common way to extract insights from data is to visualize the data. Whenever you have complex datasets, you may want to dive deeper and try to find intricate patterns in the data.

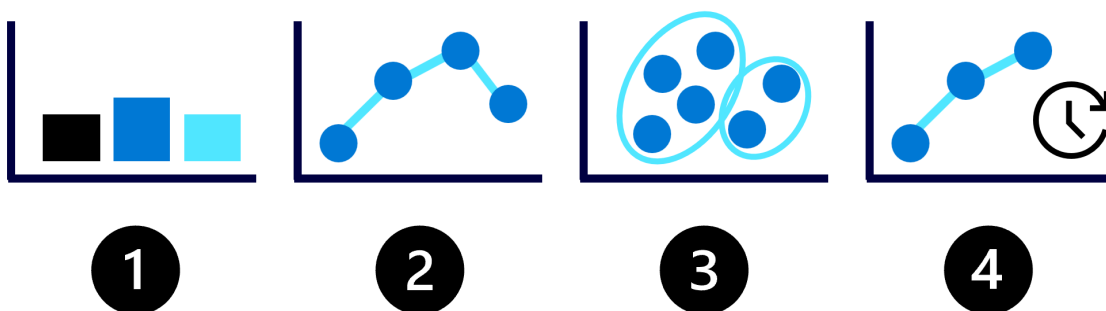
As a data scientist, you can train machine learning models to find patterns in your data. You can use those patterns to generate new insights, or predictions. For example, you can predict the expected number of products you expect to sell in the coming week.

Though training the model is important, it's not the only task in a data science project. Before exploring a typical data science process, let's explore common machine learning models you can train.

Explore common machine learning models

The purpose of machine learning is to train models that can identify patterns in large amounts of data. You can then use the patterns to make predictions that provide you with new insights on which you can take actions.

The possibilities with machine learning may appear endless, so let's begin by understanding the four common types of machine learning models:



1. **Classification:** Predict a categorical value like whether a customer may churn.
2. **Regression:** Predict a numerical value like the price of a product.
3. **Clustering:** Group similar data points into clusters or groups.
4. **Forecasting:** Predict future numerical values based on time-series data like the expected sales for the coming month.

To decide which type of machine learning model you need to train, you first need to understand the business problem and the data available to you.

Understand the data science process

To train a machine learning model, the process commonly involves the following steps:



1. **Define the problem:** Together with business users and analysts, decide on what the model should predict and when it's successful.
2. **Get the data:** Find data sources and get access by storing your data in a Lakehouse.
3. **Prepare the data:** Explore the data by reading it from a Lakehouse into a notebook. Clean and transform the data based on the model's requirements.
4. **Train the model:** Choose an algorithm and hyperparameter values based on trial and error by tracking your experiments with MLflow.
5. **Generate insights:** Use model batch scoring to generate the requested predictions.

As a data scientist, most of your time is spent on preparing the data and training the model. How you prepare the data and which algorithm you choose to train a model can influence your model's success.

You can prepare and train a model by using open-source libraries available for the language of your choice. For example, if you work with Python, you can prepare the data with Pandas and Numpy, and train a model with libraries like [Scikit-Learn](#), [PyTorch](#), or [SynapseML](#).

When experimenting, you want to keep an overview of all the different models you've trained. You want to understand how your choices influence the model's success. By tracking your experiments with MLflow in Microsoft Fabric, you're able to easily manage and deploy the models you've trained.

3. Explore and process data with Microsoft Fabric

<https://learn.microsoft.com/en-us/training/modules/get-started-data-science-fabric/3-explore-prepare-data>

Explore and process data with Microsoft Fabric

Completed

Data is the cornerstone of data science, particularly when aiming to train a machine learning model for achieving artificial intelligence. Typically, models exhibit enhanced performance as the training dataset size increases. In addition to the quantity of data, the quality of the data is equally crucial.

To guarantee both the quality and quantity of your data, using Microsoft Fabric's robust data ingestion and processing engines is worthwhile. You have the flexibility to opt for either a low-code or code-first approach when establishing the essential data ingestion, exploration, and transformation pipelines.

Ingest your data into Microsoft Fabric

To work with data in Microsoft Fabric, you first need to ingest data. You can ingest data from multiple sources, both local and cloud data sources. For example, you can ingest data from a CSV file stored on your local machine or in an Azure Data Lake Storage (Gen2).

Tip

Learn more about how to [ingest and orchestrate data from various sources with Microsoft Fabric](#).

After connecting to a data source, you can save the data into a Microsoft Fabric **lakehouse**. You can use the lakehouse as a central location to store any structured, semi-structured, and unstructured files. You can then easily connect to the lakehouse whenever you want to access your data for exploration or transformation.

Explore and transform your data

As a data scientist, you may be most familiar with writing and executing code in **notebooks**. Microsoft Fabric offers a familiar notebook experience, powered by Spark compute.

Apache Spark is an open source parallel processing framework for large-scale data processing and analytics.

Notebooks are automatically attached to Spark compute. When you run a cell in a notebook for the first time, a new Spark session starts. The session persists when you run subsequent cells. The Spark session will automatically stop after some time of inactivity to save costs. You can also manually stop the session.

When you're working in a notebook, you can choose the language you want to use. For data science workloads, you're likely to work with PySpark (Python) or SparkR (R).

Train a machine learning model and track with MLflow

Use the code in this notebook to train and track models.

+ Code + Markdown

```
1 import pandas as pd
2 # Load data into pandas DataFrame from "/lakehouse/default/" + "Files/churn.csv"
3 df = pd.read_csv("/lakehouse/default/" + "Files/churn.csv")
4 display(df)
```

[2] ✓ 2 sec - Command executed in 2 sec 43 ms by Mary-Jo Diepeveen on 3:53:40 PM, 5/08/23 PySpark (Python) ▾

> Spark jobs (1 of 1 succeeded) Log

Table Chart |→ Export results ▾

Index	CustomerID	years_with_company	total_day_calls	total_eve_calls	total_calls
1	1000038	0	117	88	32
2	1000183	1	164	102	22
3	1000326	3	116	43	45
4	1000340	0	92	24	11
5	1000471	1	53	59	37

Ready 1 of 2 cells

Within the notebook, you can explore your data using your preferred library, or with any of the built-in visualization options. If necessary, you can transform your data and save the processed data by writing it back to the lakehouse.

Prepare your data with the Data Wrangler

To help you explore and transform your data more quickly, Microsoft Fabric offers the easy-to-use **Data Wrangler**.

After launching the Data Wrangler, you'll get a descriptive overview of the data you're working with. You can view the summary statistics of your data to find any issues like missing values.

To clean your data, you can choose any of the built-in data-cleaning operations. When you select an operation, a preview of the result and the associated code is automatically generated for you. When you have selected all necessary operations, you can export the transformations to code and execute it on your data.

4. Train and score models with Microsoft Fabric

<https://learn.microsoft.com/en-us/training/modules/get-started-data-science-fabric/4-train-deploy-model>

Train and score models with Microsoft Fabric

Completed

When you've ingested, explored, and preprocessed your data, you can use the data to train a model. Training a model is an iterative process, and you want to be able to track your work.

Microsoft Fabric integrates with MLflow to easily track and log your work, enabling you to review your work at any time to decide what the best approach is to train the final model. When you track your work, your results are easily reproducible.

Any work you want to track, can be tracked as **experiments**.

Understand experiments

Whenever you train a model in a notebook that you want to track, you create an experiment in Microsoft Fabric.

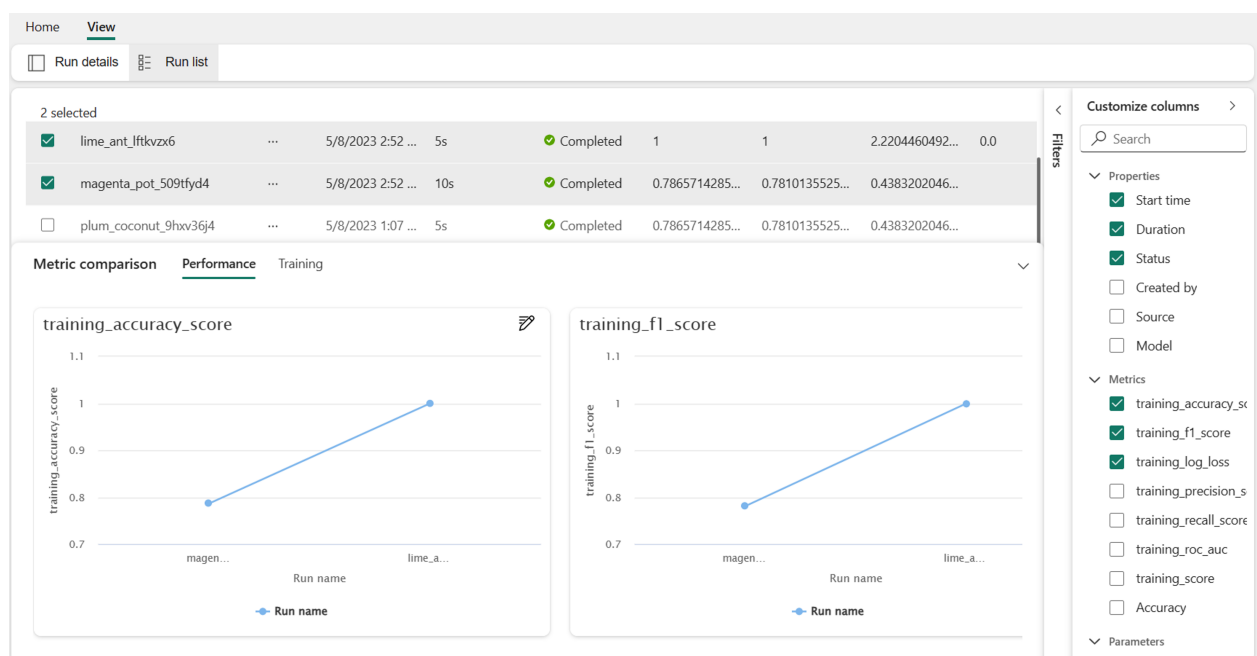
An experiment can consist of multiple runs. Each **run** represents a task you executed in a notebook, like training a machine learning model.

For example, to train a machine learning model for sales forecasting, you can try different training datasets with the same algorithm. Each time you train a model with a different dataset, you create a new experiment run. Then, you can compare the experiment runs to determine the best performing model.

Start tracking metrics

To compare experiment runs, you can track parameters, metrics, and artifacts for each run.

All parameters, metrics, and artifacts you track in an experiment run are shown in the experiments overview. You can view experiment runs individually in the **Run details** tab, or compare across runs with the **Run list**:



By tracking your work with MLflow, you can compare model training iterations and decide which configuration resulted in the best model for your use case.

Understand models

After you train a model, you want to use it for scoring. With scoring, you use the model on new data to generate predictions or insights. When you train and track a model with MLflow, artifacts are stored within the experiment run to represent your model and its metadata. You can save these artifacts in Microsoft Fabric as a **model**.

By saving your model artifacts as a registered model in Microsoft Fabric, you can easily manage your models. Anytime you train a new model and save it under the same name, you add a new version to the model.

The screenshot displays the Microsoft Fabric MLflow interface for a registered model named 'model-churn-sklearn'. The interface is divided into several sections:

- Header:** Includes 'Home' and 'View' tabs, and a toolbar with actions like 'Apply this version', 'Download model version', 'Delete version', and 'Sensitivity'.
- Left Sidebar:** Shows the model name 'model-churn-sklearn' and a list of versions: Version 3 (4:09:42 PM), Version 2 (1:07:50 PM), and Version 1 (12:44:27 PM).
- Properties Panel:** Displays metadata for the selected version (Version 3), including:
 - Version name: Version 3
 - Created time: 5/8/2023 4:09 PM
 - Last modified: 5/8/2023 4:09 PM
 - Created by: [User Icon]
 - Experiment name: experiment-churn
 - Run name: magenta_pot_509tyd4
- Model version metrics (7):** A table showing various performance metrics for the model version.
- Right Panel:** Features a green cube icon and a button labeled 'Apply this version' with a sub-label 'Apply model'.

Name	Value
Model version metrics (7)	
training_accuracy_score	0.7865714285714286
training_f1_score	0.781013552568592
training_log_loss	0.4383202046459716
training_precision_score	0.7809789511181253
training_recall_score	0.7865714285714286
training_roc_auc	0.8589021514883585
training_score	0.7865714285714286

Use a model to generate insights

To use a model for generating predictions, you can use the PREDICT function in Microsoft Fabric. The PREDICT function is built to easily integrate with MLflow models and allows you to use the model for generating batch predictions.

For example, every week you receive sales data from several stores. Based on the historical data, you've trained a model that can predict the sales for the next week, based on the sales of the last few weeks. You tracked the model with MLflow and saved it in Microsoft Fabric. Whenever the new weekly sales data comes in, you use the PREDICT function to let the model generate the forecast for the next week. The forecasted sales data is stored as a table in a lakehouse, which is visualized in a Power BI report for business users to consume.

5. Exercise - Explore data science in Microsoft Fabric

<https://learn.microsoft.com/en-us/training/modules/get-started-data-science-fabric/5-exercise-use-notebook>

Exercise - Explore data science in Microsoft Fabric

Completed

Now it's your chance to explore the data science features in Microsoft Fabric.

Note

You need a Microsoft Fabric trial license with the Fabric preview enabled in your tenant. See [Getting started with Fabric](#) to enable your Fabric trial license.

Launch the exercise and follow the instructions.

[Launch Exercise](#)

6. Module assessment

<https://learn.microsoft.com/en-us/training/modules/get-started-data-science-fabric/6-knowledge-check>

Module assessment

Completed

7. Summary

Summary

Completed

Microsoft Fabric offers one central workspace to perform data science from beginning to end.

To perform data science, you need to first define the problem. Then, you can identify the data you need and ingest it into Microsoft Fabric. When you've ingested your data, you can explore and prepare your data, using notebooks or the Data Wrangler.

To train machine learning models as part of your data science project, you can track your work with experiments. To use a model to generate insights, you can use the built-in PREDICT function.