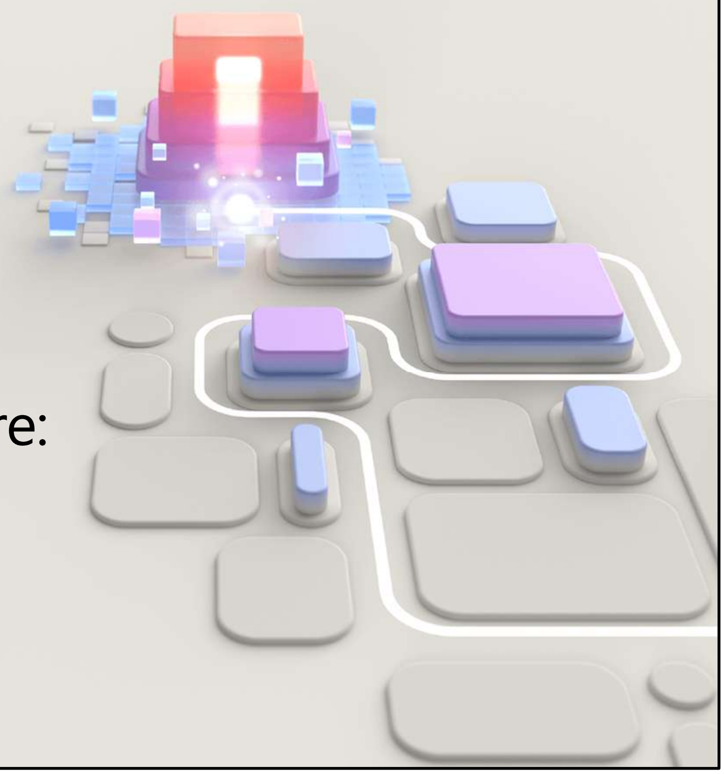
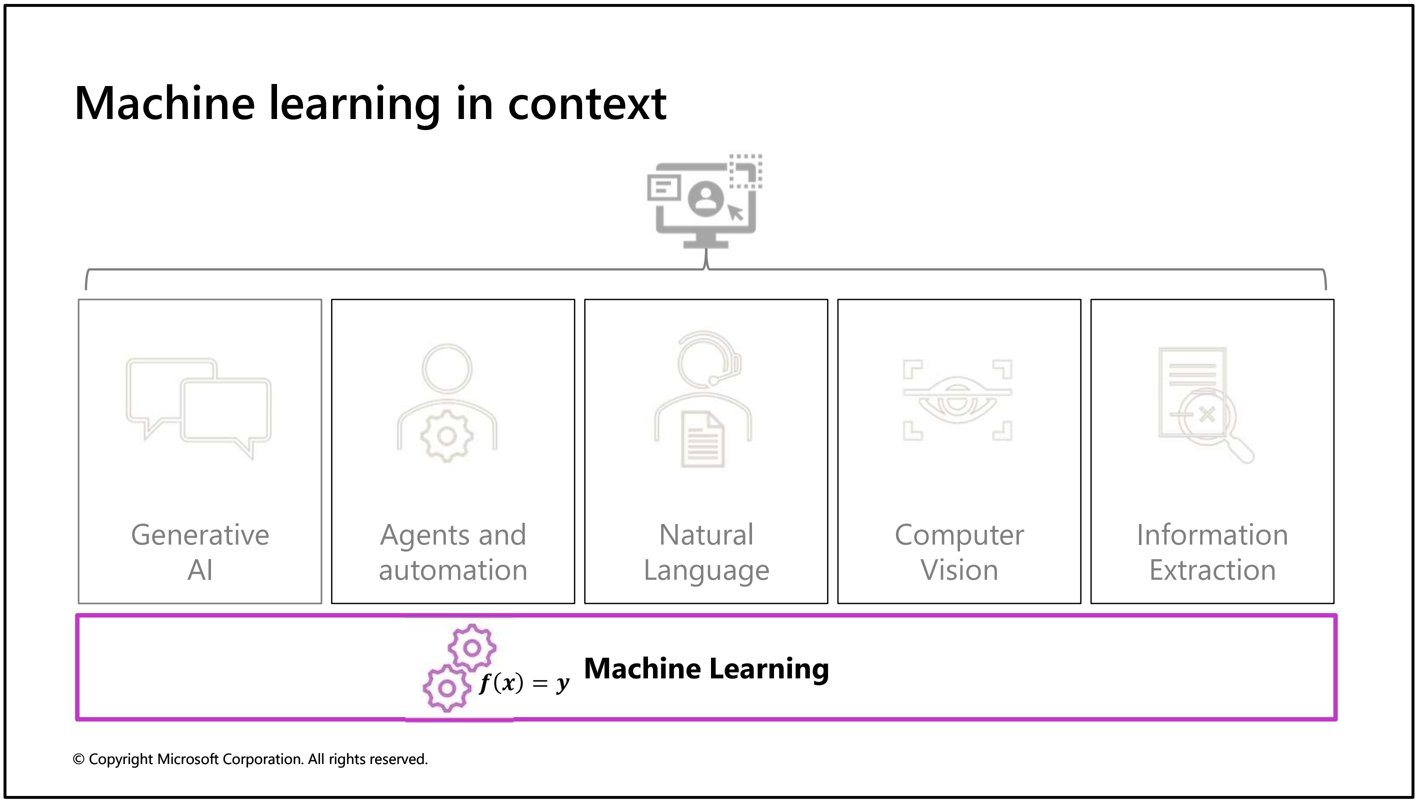
Microsoft

Introduction to AI in Azure:
Machine Learning





In this section we will focus on machine learning. It's the foundation of modern AI.

Agenda

- Introduction to machine learning concepts
- Get started with machine learning in Azure

Time estimates:

- Introduction to machine learning concepts – 35 (including lab exercise)
- Get started with machine learning in Azure – 25 (including demo)

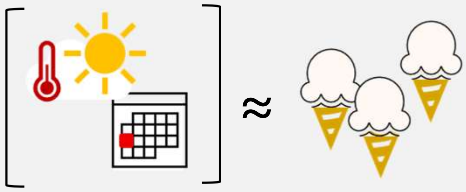
Use the link on the slide to see the Microsoft Learn learning module from which this section is derived.

What is machine learning?

Predictive models that encapsulate relationships between:

- **Features** (known characteristics of something)
- **Label** (the thing we want to predict)

Example: Predicting ice cream sales for a given day



Features
- Day of week
- Month
- Temperature
- …

Label
- Number of ice creams sold

Generalized function

$$f(\underbrace{x}_{[x_1, x_2, x_3]}) = \hat{y}$$

Features — Label

(typically, an array of multiple features)

Vad är maskininlärning?

Maskininlärning handlar om att skapa prediktiva modeller som kan förutsäga framtida värden genom att hitta mönster i data. Processen delas upp i två huvudfaser: träning och inferens.

1. Träningsdata (Training Data)

För att börja behövs historisk data – det vill säga tidigare observationer.
Varje rad består av:

Features (x) – det är de variabler eller egenskaper vi använder för att förutsäga något (t.ex. temperatur, veckodag, kampanj).

Label (y) – det vi vill förutsäga (t.ex. glassförsäljning den dagen).

Exempel:
$[x_1, x_2, x_3]$, y
Där $[x_1, x_2, x_3]$ är t.ex. väder, datum och temperatur, och y är antal sålda glassar.

2. Algoritm

Nu används en algoritm – det är en matematisk metod som analyserar sambandet mellan x (features) och y (label). Algoritmen skapar en funktion som försöker generalisera detta samband, alltså hitta ett mönster eller en regel som gäller för datan.

3. Modell

Efter att algoritmen har lärt sig sambandet skapas en modell, som i praktiken är funktionen y = f(x).
Modellen kan nu ta emot nya x-värden och ge ett förutsagt y. Denna modell är resultatet av träningsfasen – den "förstår" hur tidigare data hänger ihop.

Inferens (Användning av modellen)

4. Inferensdata (Inferencing Data)

När modellen är färdigtränad används den på ny data – alltså nya features $[x_1, x_2, x_3]$ utan känt resultat. Denna data saknar label (y), eftersom det är den vi vill förutsäga.

Prediktion

Modellen använder då sin inlärda funktion för att generera ett förutsagt värde:
$\hat{y} = f(x)$
Här är ŷ den förutsagda labeln – ett uppskattat resultat baserat på tidigare lärdom.

Sammanfattning

Vi tränar modellen med data vi känner till (x och y).

Algoritmen hittar samband och bygger en modell.

Vi matar in ny data (x) utan label.

Modellen förutspår det okända värdet (ŷ).

Detta är grunden i maskininlärning: att låta datorn *lära sig* från tidigare data för att kunna fatta smarta beslut i framtiden.

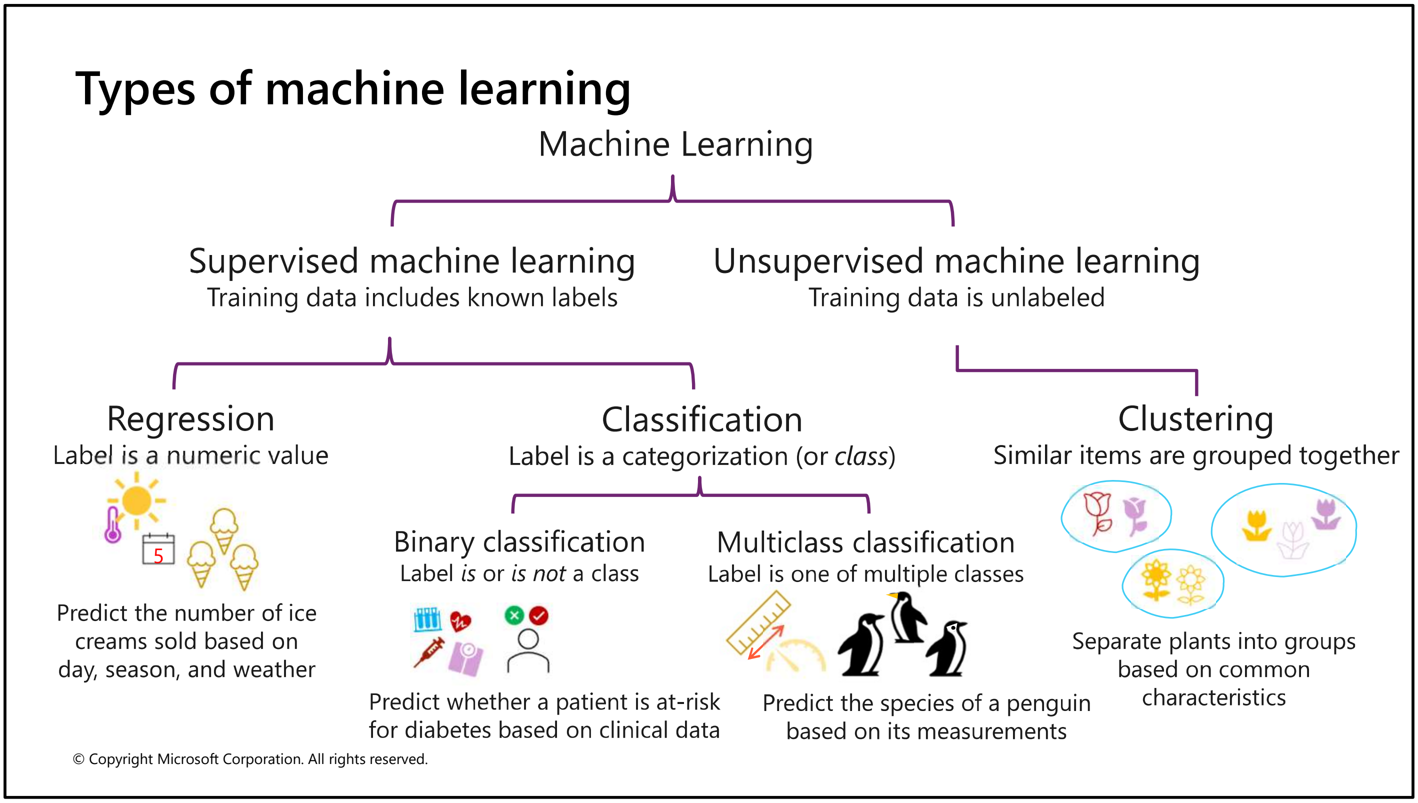
Fundamentally, a machine learning model is a software application that encapsulates a *function* to calculate an unknown output value based on one or more known input values. The process of defining that function is known as *training*. After the function has been defined, you can use it to predict new values in a process called *inferencing*.

The training data for a machine learning model consists of past observations. In most cases, the observations include the observed attributes or *features* of the thing being observed, and the known value of the thing you want to train a model to predict (known as the *label*). In mathematical terms, you'll often see the features referred to using the shorthand variable name ***x***, and the label referred to as ***y***. Usually, an observation consists of multiple feature values, so ***x*** is actually a *vector* (an array with multiple values), like this: $[x_1,x_2,x_3,...]$.

An *algorithm* is applied to the data to try to determine a relationship between the features and the label, and generalize that relationship as a calculation that can be performed on ***x*** to calculate ***y***. The specific algorithm used depends on the kind of predictive problem you're trying to solve (more about this later), but the basic principle is to try to *fit* the data to a function in which the values of the features can be used to calculate the label.

The result of the algorithm is a *model* that encapsulates the calculation derived by the algorithm as a *function* - let's call it ***f***. In mathematical notation: $f(x) = y$

After the *training* phase is complete, the trained model can be used for *inferencing*. The model is essentially a software program that encapsulates the function produced by the training process. You can input a set of feature values, and receive as an output a prediction of the corresponding label. Because the output from the model is a prediction that was calculated by the function, and not an observed value, you'll often see the output from the function shown as $\hat{y}$ (which is rather delightfully verbalized as "y-hat").



Olika typer av maskininlärning

Maskininlärning (Machine Learning) delas in i två huvudgrupper: Supervised och Unsupervised lärande.
Skillnaden ligger i om träningen sker med kända svar (labels) eller inte.

1. Supervised Machine Learning

Här tränas modellen med data där både input (features) och rätt svar (labels) är kända. Detta används när du vill förutsäga något.

a) Regression

Vad det är: Förutsägelse av numeriska värden.

Exempel: Förutsäga antal sålda glassar beroende på dag, väder och säsong.

Label: Ett tal, t.ex. 120 glassar.

b) Classification

Vad det är: Förutsäga kategorier eller klasser.

Två undertyper:

- Binary classification
  - Endast två möjliga utfall (ja/nej, 1/0).
  - Exempel: Avgöra om en patient riskerar att få diabetes.
- Multiclass classification
  - Flera möjliga kategorier.
  - Exempel: Identifiera vilken art en pingvin tillhör baserat på kroppsmått.

2. Unsupervised Machine Learning

Här har datan inga labels. Modellen ska själv hitta mönster eller strukturer i datan.

a) Clustering

Vad det är: Gruppera liknande dataobjekt automatiskt.

Exempel: Dela in växter i grupper baserat på deras egenskaper.
Modellen vet inte vad grupperna är – den upptäcker dem själv.

Sammanfattning

Supervised = tränas med svar (förutsäga något känt).
Unsupervised = inga svar ges (upptäck mönster).
Regression = tal, Classification = kategorier, Clustering = grupper utan kända etiketter.
Detta utgör grunden för att förstå och välja rätt typ av maskininlärning beroende på uppgiften.

Supervised machine learning

Supervised machine learning is a general term for machine learning algorithms in which the training data includes both *feature* values and known *label* values. Supervised machine learning is used to train models by determining a relationship between the features and labels in past observations, so that unknown labels can be predicted for features in future cases.

Regression

Regression is a form of supervised machine learning in which the label predicted by the model is a numeric value. For example:

•The number of ice creams sold on a given day, based on the temperature, rainfall, and windspeed.

•The selling price of a property based on its size in square feet, the number of bedrooms it contains, and socio-economic metrics for its location.

•The fuel efficiency (in miles-per-gallon) of a car based on its engine size, weight, width, height, and length.

Classification

Classification is a form of supervised machine learning in which the label represents a categorization, or *class*. There are two common classification scenarios.

Binary classification

In *binary classification*, the label determines whether the observed item *is* (or *isn't*) an instance of a specific class. Or put another way, binary classification models predict one of two mutually exclusive outcomes. For example:

•Whether a patient is at risk for diabetes based on clinical metrics like weight, age, blood glucose level, and so on.

•Whether a bank customer will default on a loan based on income, credit history, age, and other factors.

•Whether a mailing list customer will respond positively to a marketing offer based on demographic attributes and past purchases.

In all of these examples, the model predicts a binary *true/false* or *positive/negative* prediction for a single possible class.

Multiclass classification

Multiclass classification extends binary classification to predict a label that represents one of multiple possible classes. For example,

•The species of a penguin (*Adelie*, *Gentoo*, or *Chinstrap*) based on its physical measurements.

•The genre of a movie (*comedy*, *horror*, *romance*, *adventure*, or *science fiction*) based on its cast, director, and budget.

In most scenarios that involve a known set of multiple classes, multiclass classification is used to predict mutually exclusive labels. For example, a penguin can't be both a *Gentoo* and an *Adelie*. However, there are also some algorithms that you can use to train *multilabel* classification models, in which there may be more than one valid label for a single observation. For example, a movie could potentially be categorized as both *science fiction* and *comedy*.

Unsupervised machine learning

Unsupervised machine learning involves training models using data that consists only of *feature* values without any known labels. Unsupervised machine learning algorithms determine relationships between the features of the observations in the training data.

Clustering

The most common form of unsupervised machine learning is *clustering*. A clustering algorithm identifies similarities between observations based on their features, and groups them into discrete clusters. For example:

•Group similar flowers based on their size, number of leaves, and number of petals.

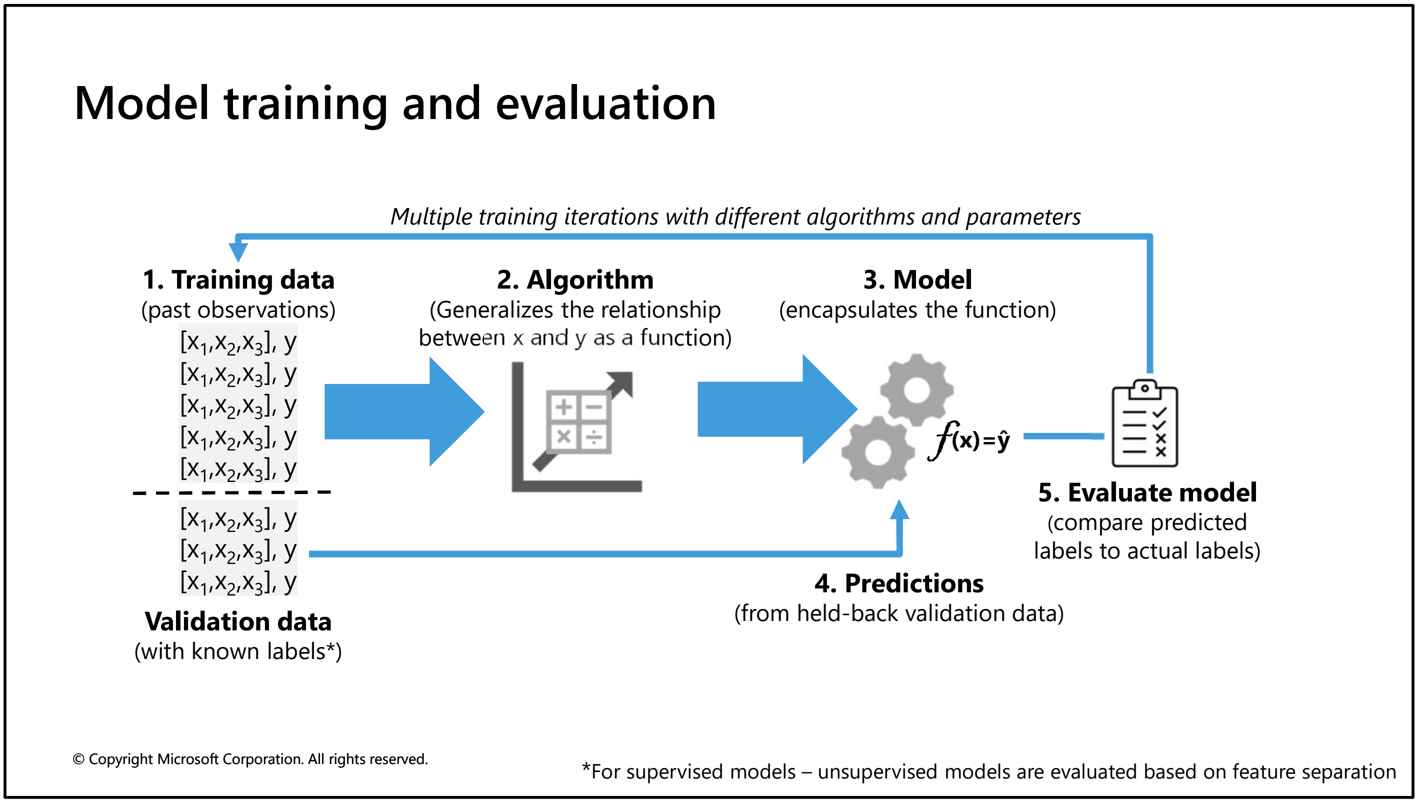
•Identify groups of similar customers based on demographic attributes and purchasing behavior.

In some ways, clustering is similar to multiclass classification; in that it categorizes observations into discrete groups. The difference is that when using classification, you already know the classes to which the observations in the training data belong; so the algorithm works by determining the relationship between the features and the known classification label. In clustering, there's no previously known cluster label and the algorithm groups the data observations based purely on similarity of features.

In some cases, clustering is used to determine the set of classes that exist before training a classification model. For example, you might use clustering to segment your customers into groups, and then analyze those groups to identify and categorize different classes of customer (*high value - low volume*, *frequent small purchaser*, and so on). You could then use your categorizations to label the observations in your clustering results and use the labeled data to train a classification model that predicts to which customer category a new customer might belong.

How this relates to x and y from the previous slide:

1. In the ice cream sales scenario, our goal is to train a model that can predict the number of ice cream sales based on the weather. The weather measurements for the day (temperature, rainfall, windspeed, and so on) would be the *features* (***x***), and the number of ice creams sold on each day would be the *label* (***y***).
2. In the medical scenario, the goal is to predict whether or not a patient is at risk of diabetes based on their clinical measurements. The patient's measurements (weight, blood glucose level, and so on) are the *features* (***x***), and the likelihood of diabetes (for example, *1* for at risk, *0* for not at risk) is the *label* (***y***).
3. In the Antarctic research scenario, we want to predict the species of a penguin based on its physical attributes. The key measurements of the penguin (length of its flippers, width of its bill, and so on) are the *features* (***x***), and the species (for example, *0* for Adelie, *1* for Gentoo, or *2* for Chinstrap) is the *label* (***y***).



Modellträning och utvärdering i maskininlärning

Att träna en maskininlärningsmodell innebär inte bara att hitta mönster i data – det kräver också att modellen testas och utvärderas. Denna process kan delas in i fem steg:

1. Träningsdata (Training data)

Det hela börjar med historisk data, där både features (x) och labels (y) är kända.
Modellen ska lära sig sambandet mellan dem.

2. Algoritm

En algoritm används för att analysera relationen mellan x och y, och försöker skapa en funktion som generaliserar sambandet. Olika algoritmer kan testas med olika inställningar (parametrar).

3. Modell

När algoritmen har "lärt sig" sambandet skapas en modell, som är en matematisk funktion y = f(x). Den kan användas för att göra förutsägelser på ny data.

4. Validering och prediktion (Validation and Prediction)

En del av datan – kallad valideringsdata – har också kända labels men används inte i träningen.
Modellen får denna data och gör prediktioner, alltså förutsägelser av ŷ utan att "veta" de riktiga svaren.

5. Utvärdering (Evaluate model)

Nu jämförs modellens förutsägelser (ŷ) med de verkliga etiketterna (y).
Skillnaden mäts med olika prestandamått (t.ex. noggrannhet, felkvadrat, precision), vilket visar hur bra modellen fungerar.

Upprepad träning

Om modellen inte presterar tillräckligt bra kan man göra flera träningsomgångar med olika algoritmer, parametrar eller datafördelningar – en process kallad modelloptimering.

Sammanfattning

Träna modellen med data med kända svar.

Bygg modellen med en lämplig algoritm.

Använd modellen på ny data (validering).

Jämför resultatet med de riktiga svaren.

Förbättra vid behov.

Detta är centralt för att bygga en modell som både lär sig effektivt och presterar pålitligt i verkligheten.

1. Split the data into a *training* set and a *validation* set
2. Apply an algorithm to *fit* the training data to a model
3. The trained model encapsulates the relationships in the data
4. Use the model to generate predictions from the validation data
5. Use evaluation metrics to compare predicted vs actual labels (supervised) or measure cluster separation (unsupervised)
6. Repeat…

The learning component of machine learning occurs during training. We try to capture the relationships between the features and label in a model. Training is the action of iterating on an algorithm to best fit, or encapsulate those relationships.

After training, we have a model that we can test. We can use some of the data set aside, validation data, to test how closely our model's predicted labels are to actual labels. In the case of unsupervised learning where we don't know the actual labels, we measure how well our model's predicted labels were at separating clusters. There are many types of evaluation metrics. The important thing to remember is that the goal of machine learning is to find a model that gets as close as possible to predicting the actual label. The best model can still have some margin of error.

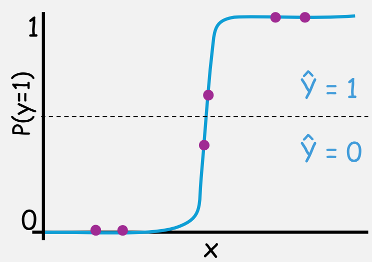
Model algorithms and evaluation metrics

Regression

Example, *Linear Regression*

Evaluation metrics:
- *Mean absolute error (MAE)*
- *Mean squared error (MSE)*
- *Root mean squared error (RMSE)*
- *Coefficient of determination (R^2)*

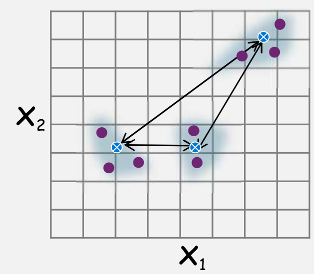
Classification



Example, *Logistic Regression*

Evaluation metrics (per class and overall):
- *Accuracy*
- *Recall*
- *Precision*
- *F1 Score*

Clustering



Example, *K-Means*

Evaluation metrics:
- *Cluster centroid distance*
- *Mean distance to centroid within cluster*
- *Sillhouette score*

Bilden sammanfattar tre vanliga typer av maskininlärningsproblem och hur deras modeller utvärderas: regression, klassificering och klustring.

Till vänster visas regression, där målet är att förutsäga ett kontinuerligt numeriskt värde, till exempel pris eller temperatur. Exemplet är linjär regression, där en linje anpassas till datapunkter. Vanliga utvärderingsmått är MAE, MSE och RMSE, som mäter hur stora felen är mellan verkliga och predikterade värden, samt R^2 som anger hur stor del av variationen modellen förklarar.

I mitten visas klassificering, där modellen förutspår diskreta klasser, till exempel ja/nej. Exemplet är logistisk regression, som ger en sannolikhet som sedan omvandlas till en klass. Utvärdering sker med mått som accuracy, precision, recall och F1-score, vilka beskriver hur väl modellen skiljer mellan klasser.

Till höger visas klustring, en osuperviserad metod där data grupperas utan facit. K-means delar in datapunkter i kluster baserat på avstånd till centroids. Kvaliteten mäts med till exempel genomsnittligt avstånd inom kluster och silhouette score, som visar hur tydliga klustren är.

Regression

Linear regression is probably the simplest algorithm to understand. It works by plotting the features (x) and label (y) and then calculating a straight line through the points (in multidimensional space depending on the number of features) with the minimum mean distance between the line and the plotted points. The slope of the line determines the function definition, and is used to calculate the predicted label (ŷ) as the point on the line that intersects with the feature (x) values.

To evaluate a regression model, use it to calculate the predicted label (ŷ) for each held-back observation based on their feature (x) values. You can then compare the known label (y) to the predicted label (ŷ) and measure how far off the prediction is. Specific metrics include:

- Mean absolute error (MAE): The average (mean) difference bewteen the predicted ŷ values and the actual y values.

- Mean squared error (MSE): The average (mean) difference between ŷ and y, squared. Squaring the differences magnifies the effect of large inaccuracies.

- Root mean squared error (RMSE): The root of the MSE metric (so that that the error is expressed in the same unit as the label – for example, ice creams sold)

- Coefficient of determination (as R-squared): The proportion of the difference between predicted and actual values that can be explained by expected statistical variance in the model (as opposed to being the result of additional, factors that are not represented in the model)

Classification

Despite the name, logistic regression is an algorithm commonly used for classification, not regression. It works by calculating a value between 0 and 1 for each class based on the features (x). Each class result indicates the probability that the observation belongs to that class. In binary classification (where there is only one class with a possible value of true or false – for example, a patient has diabetes or does not); the result of the logistic regression calculation represents the probability that y is true, and a threshold value (usually 0.5) is used to determine if the model predicts 0 (false) or 1 (true). For multiclass classification, the class with the highest predicted probability is the one that the model predicts.

Evaluation metrics for classification include:

- Accuracy: The number of proportion of correct predictions out of the total number of predictions.
- Recall: The proportion of positive cases correctly identified by the model.
- Precision: The proportion of positive predictions that are correct
- F1 Score: A combined score for recall and precision

Clustering

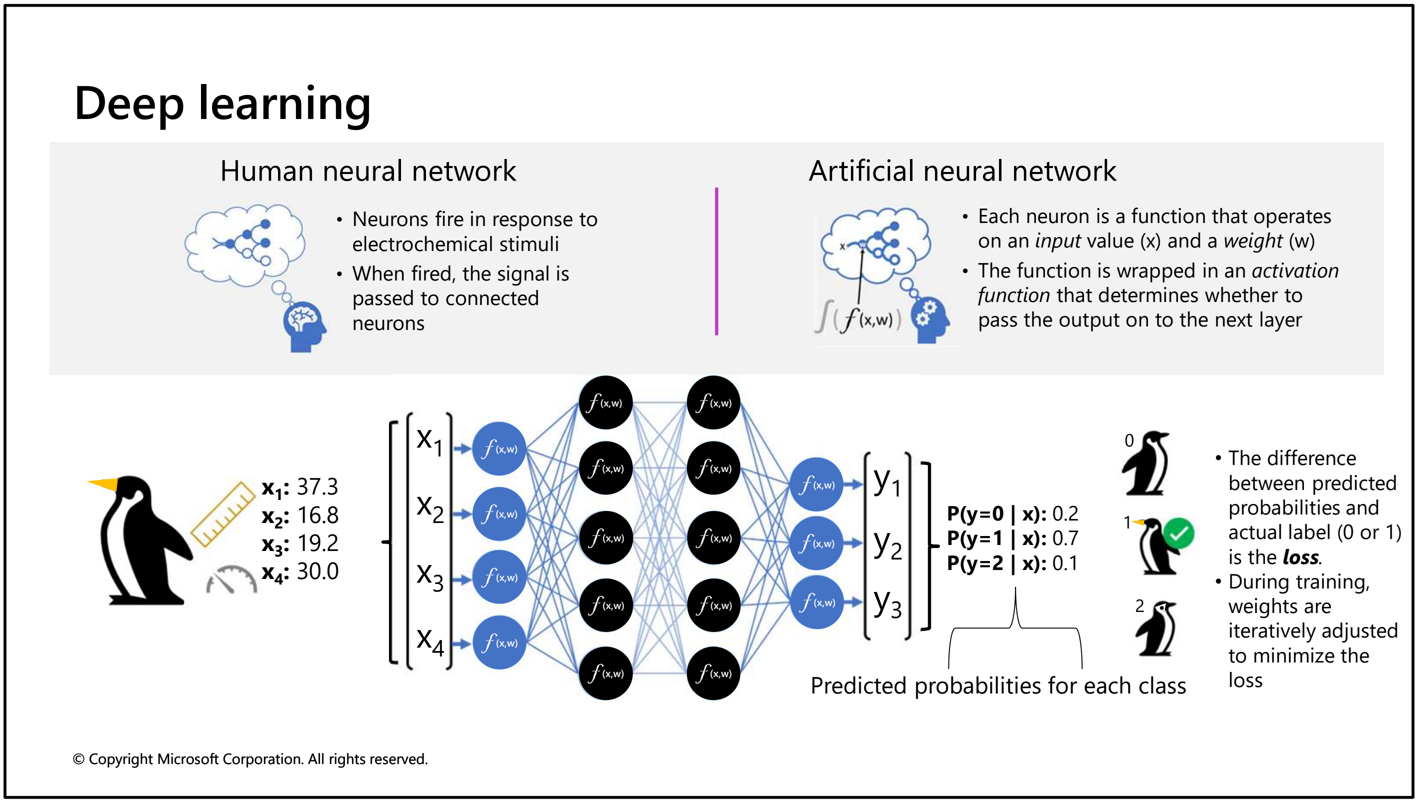
K-means clustering works by plotting the data in multidimensional space based on their feature (x) values. You then select the number if clusters you want the algorithm to find (a value we call *k*) and randomly assign center points (called *centroids*) in the same multidimensional space. You then iteratively:

- Assign observations to their nearest centroid
- Move the centroids to the center of their observations (based on the mean distance)
- Reassign observations to the nearest centroid (which may have changed)

You repeat this process for multiple iterations until the clusters are stable or a preset limit of iterations is reached.

To evaluate a clustering model, there are no known labels, so you must base the model performance on how well it separates the data into clusters. Common metrics include:

- The distance between the cluster centroids.
- The mean distance between observation data points and their cluster's centroid
- A calculated *silhouette* score that measures separation on a scale of 0 to 1.



🧠 Vad visar bilden?

Det här är ett exempel på deep learning där ett neuralt nätverk används för att gissa vilken sorts pingvin det är – alltså multiklassklassificering.

🐧 Så här går det till, steg för steg:

Mätvärden (x_1 till x_4):
Man mäter t.ex. näbblängd, vinglängd och andra egenskaper hos pingvinen.

Inmatning till nätverket:
De fyra mätvärdena (x_1–x_4) skickas in i det neurala nätverket.

Dolda lager (svarta cirklar):
Här bearbetas värdena i flera steg. Varje cirkel är som en liten beräkning.

Utdatavärden (y_1–y_3):
Nätverket räknar ut hur troligt det är att pingvinen tillhör art 0, 1 eller 2.

Slutgissning:
Den högsta sannolikheten är för art 1 (70 %), så modellen gissar att pingvinen är av typ 1 – vilket är rätt!

✅ Sammanfattning:

Du matar in siffror ➜ nätverket bearbetar dem ➜ det gissar vilken klass (pingvinart) som är mest trolig.

Det är så ett neuralt nätverk "tänker".

Deep learning is an advanced form of machine learning that tries to emulate the way the human brain learns. The key to deep learning is the creation of an artificial *neural network* that simulates electrochemical activity in biological neurons by using mathematical functions.

Artificial neural networks are made up of multiple *layers* of neurons - essentially defining a deeply nested function. This architecture is the reason the technique is referred to as *deep learning* and the models produced by it are often referred to as *deep neural networks* (DNNs). You can use deep neural networks for many kinds of machine learning problem, including regression and classification, as well as more specialized models for natural language processing and computer vision.

To better understand how a deep neural network model works, let's explore an example in which a neural network is used to define a classification model for penguin species.

The feature data (***x***) consists of some measurements of a penguin. Specifically, the measurements are:

•The length of the penguin's bill.

•The depth of the penguin's bill.

•The length of the penguin's flippers.

•The penguin's weight.

The label we're trying to predict (***y***) is the species of the penguin, and that there are three possible species it could be:

•Adelie

•Gentoo

•Chinstrap

The process for inferencing a predicted penguin class using this network is:

1.The feature vector for a penguin observation is fed into the input layer of the neural network, which consists of a neuron for each ***x*** value. In this example, the following ***x*** vector is used as the input: ***[37.3, 16.8, 19.2, 30.0]***

2.The functions for the first layer of neurons each calculate a weighted sum by combining the ***x*** value and ***w*** weight. ***

3.Each neuron in a layer is connected to all of the neurons in the next layer. The results of each layer are fed forward through the network until they reach the output layer.

4.The output layer produces a vector of values; in this case, using a *softmax* or similar function to calculate the probability distribution for the three possible classes of penguin. In this example, the output vector is: ***[0.2, 0.7, 0.1]***

5.The elements of the vector represent the probabilities for classes 0, 1, and 2. The second value is the highest, so the model predicts that the species of the penguin is **1** (Gentoo).

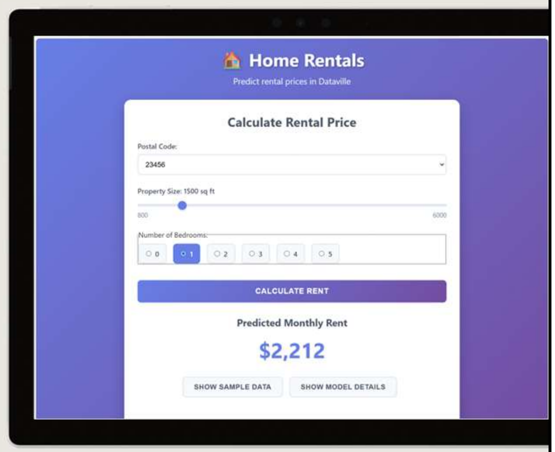
The model learns the appropriate weights by iteratively passing the training data through and evaluating the predicted probabilities for each class against the known class label. The weights are adjusted in each iteration to reduce the amount of error (*loss*) between the predicted probabilities (for example [0.2, 0.7, 0.1] and the values that represent the actual class label (for example [0.0, 1.0, 0.0] for class 1 (Gentoo)

Exercise

Explore machine learning scenarios

In this exercise, you'll explore applications that predict unknown values.

Start the exercise at:
https://go.microsoft.com/fwlink/?linkid=2334118



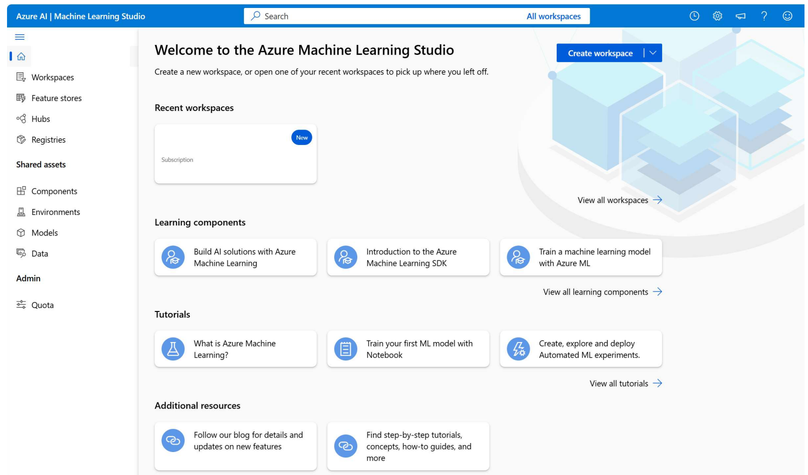
Note: The applications used in this exercise are based on simple machine learning models with simplified features. However, they're based on real models that were trained and tested using Azure Machine Learning - a platform for machine learning model development, deployment, and management.

After completing the exercise, ask the students questions to check their understanding. For example:, what kind of machine learning is demonstrated in the home rental estimator (regression), seed identifier (multiclass classification), and customer segmentation (clustering) scenarios.

Use the link on the slide to see the Microsoft Learn learning module from which this section is derived.

What is Azure Machine Learning?

- Azure Machine Learning is a cloud-based platform for machine learning
- Azure Machine Learning Studio is a user interface for accessing Azure Machine Learning capabilities
- Machine learning models trained with Azure Machine Learning can be published as services and consumed by applications



After talking to the slide for a minute or so, show students your Azure ML workspace in the Azure Portal, and then show them Azure ML Studio (ml.azure.com)

Microsoft Azure Machine Learning is a cloud service for training, deploying, and managing machine learning models. It's designed to be used by data scientists, software engineers, devops professionals, and others to manage the end-to-end lifecycle of machine learning projects, including:

•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.

After you've provisioned an Azure Machine Learning workspace, you can use it in *Azure Machine Learning studio*; a browser-based portal for managing your machine learning resources and jobs.

In Azure Machine Learning studio, you can (among other things):

•Import and explore data.

•Create and use compute resources.

•Run code in notebooks.

•Use automated machine learning to train models.

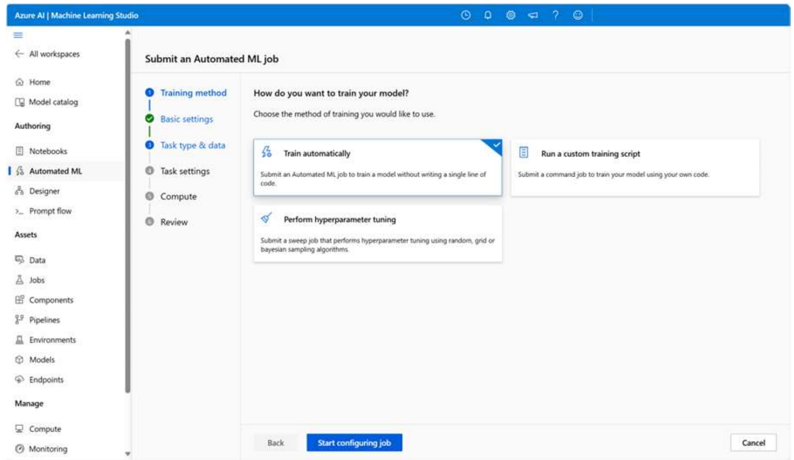
•View details of trained models, including evaluation metrics, responsible AI information, and training parameters.

•Deploy trained models for on-request and batch inferencing.

•Import and manage models from a comprehensive model catalog.

What is Automated Machine Learning?

- A step-by-step wizard that helps you run machine learning training jobs
- Supports multiple machine learning types, including regression, time-series forecasting, classification, computer vision, and natural language processing tasks
- Connect to your data, define the training job and target metrics, and deploy the best resulting model

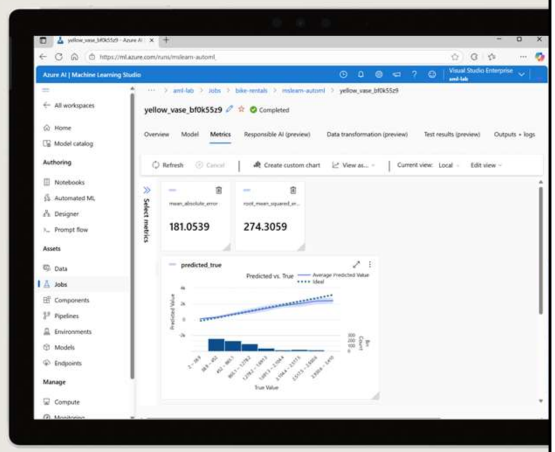


Exercise – If time permits

Explore Automated Machine Learning

In this exercise, you'll use the Automated Machine Learning capability of Azure Machine Learning to train a model.

Start the exercise at:
https://go.microsoft.com/fwlink/?linkid=2250144



You require an Azure subscription to perform this exercise – which may be provided by an Authorized Lab Hoster.

The exercise can take a substantial amount of time. Our recommendation in one-day deliveries is for the instructor to demonstrate the core tasks in this exercise; completing the exercise head of time (https://go.microsoft.com/fwlink/?linkid=2250144) so you have a trained and deployed model to demonstrate. During class, show the initial steps to create the Auto ML job; then cancel and view the results of the job you ran earlier. Finally, view and test the deployed endpoint.

Knowledge check

1 **You want to create a model to predict the cost of heating an office building based on its size in square feet and the number of employees working there. What kind of machine learning problem is this?**

- ☑ Regression
- ☐ Classification
- ☐ Clustering

2 **You need to evaluate a classification model. Which metric can you use?**

- ☐ Mean squared error (MSE)
- ☑ Precision
- ☐ Silhouette

3 **In deep learning, what is loss?**

- ☐ Data deleted from the training set
- ☑ The difference between predicted and actual label values
- ☐ The financial cost of training a model

Här är korta svar med max 10 ord per motivering:

1. Prediktera uppvärmningskostnad

✅ Rätt (Regression): Förutsäger ett kontinuerligt numeriskt värde.
❌ Fel (Classification): Inga diskreta klasser ska väljas.
❌ Fel (Clustering): Ingen gruppering utan facit efterfrågas.

2. Utvärdera en klassificeringsmodell

❌ Fel (MSE): Används för regression, inte klassificering.
✅ Rätt (Precision): Mäter andel korrekta positiva prediktioner.
❌ Fel (Silhouette): Används för klustring, inte klassificering.

3. Vad är loss i deep learning?

❌ Fel: Loss handlar inte om borttagen träningsdata.
✅ Rätt: Mäter felet mellan prediktion och verkligt värde.
❌ Fel: Loss är inte ekonomisk kostnad.

Allow students a few minutes to think about the questions and then reveal the correct answers.

Summary

Introduction to Machine Learning concepts

- What is machine learning?
- Types of machine learning
- Model training and validation
- What is Deep Learning?

Get started with machine learning on Azure

- What is Azure Machine Learning?
- What is Automated Machine Learning?