What is AI?

Simply put, AI is software that imitates human behaviors and capabilities. Key workloads include:

Machine learning - This is often the foundation for an AI system, and is the way we "teach" a computer model to make predictions and draw conclusions from data.

Computer vision - Capabilities within AI to interpret the world visually through cameras, video, and images.

Natural language processing - Capabilities within AI for a computer to interpret written or spoken language, and respond in kind.

Document intelligence - Capabilities within AI that deal with managing, processing, and using high volumes of data found in forms and documents.

Knowledge mining - Capabilities within AI to extract information from large volumes of often unstructured data to create a searchable knowledge store.

Generative AI - Capabilities within AI that create original content in a variety of formats including natural language, image, code, and more.

 Machine Learning (ML) is the foundation for most AI solutions.

 Researchers, known as data scientists, have been working on AI since the 1950s.

 Modern AI applications originate from ML, which combines computer science and mathematics.

 Example: The Yield, an Australian agricultural tech company, uses sensors, data, and ML to help farmers make decisions about weather, soil, and plant conditions.

 ML works by learning from data, which is generated in massive amounts daily from various sources like social media, sensors, and devices.

 Data scientists train ML models using this data to make predictions and inferences.

 Example scenario: An environmental conservation organization uses ML to help volunteers identify wildflower species via a phone app.

* Botanists collect and label wildflower data.
* An algorithm processes the data, finding relationships between features and species.
* A model is created to identify species in new samples found by volunteers.

 AI approaches have advanced, enabling the completion of complex tasks.

 Microsoft Azure provides the Azure Machine Learning service, a cloud-based platform for ML models.

* Automated machine learning allows non-experts to create ML models quickly.
* Azure Machine Learning designer offers a no-code development graphical interface.
* Data metric visualization helps analyze and optimize experiments.
* Notebooks allow writing and running code in integrated Jupyter Notebook servers.
* **Computer Vision Overview**
  + Computer Vision is a branch of AI focused on visual processing.
  + It involves using machine learning models to analyze visual inputs from cameras, videos, or images.
* **Example: Seeing AI App**
  + Designed for the blind and low vision community.
  + Uses AI to describe nearby people, text, and objects, making the visual world accessible.
* **Common Computer Vision Tasks**
  + **Image Classification**
    - Training models to classify images based on their contents.
    - Example: Identifying types of vehicles (e.g., taxis, buses) in traffic monitoring.
  + **Object Detection**
    - Identifying and classifying individual objects within an image, highlighting them with bounding boxes.
    - Example: Locating different classes of vehicles in traffic images.
  + **Semantic Segmentation**
    - Classifying individual pixels in an image according to the object they belong to.
    - Example: Highlighting different vehicles in traffic images using specific colors.
  + **Image Analysis**
    - Combining models to extract information from images, including tags and descriptive captions.
    - Example: Captioning a scene with a person and a dog on a street.
  + **Face Detection, Analysis, and Recognition**
    - Detecting and recognizing human faces in images.
    - Example: Identifying individuals based on facial features in a crowd.
  + **Optical Character Recognition (OCR)**
    - Detecting and reading text in images.
    - Example: Reading road signs or extracting information from scanned documents.
* **Microsoft Azure AI Vision**
  + Offers tools to develop computer vision solutions.
  + Features available in Azure Vision Studio and other programming languages.
  + Key capabilities:
    - **Image Analysis**: Analyzing images and videos to extract descriptions, tags, objects, and text.
    - **Face**: Building solutions for face detection and recognition.
    - **OCR**: Extracting printed or handwritten text from images, digitizing scanned text

**Natural Language Processing (NLP) Overview**

* NLP is a field of AI focused on creating software that understands written and spoken language.
* Applications include text analysis, speech interpretation, translation, and command interpretation.

**Capabilities of NLP Software**

* **Text Analysis and Interpretation**
  + Analyzing and interpreting text in documents, emails, and other sources.
* **Speech Interpretation and Synthesis**
  + Interpreting spoken language and synthesizing speech responses.
* **Automatic Translation**
  + Translating spoken or written phrases between languages.
* **Command Interpretation**
  + Interpreting commands and determining appropriate actions.

 **Example: Starship Commander**

* A VR game by Human Interact set in a science fiction world.
* Utilizes NLP to allow players to control the narrative and interact with in-game characters and systems.

 **NLP in Microsoft Azure**

* **Azure AI Language**
  + Features for understanding and analyzing text.
  + Training conversational language models to understand spoken or text-based commands.
  + Building intelligent applications.
* **Azure AI Speech**
  + Features include speech recognition and synthesis.
  + Real-time translations and conversation transcriptions.
* **Tools for Exploration and Development**
  + Azure Language Studio: Explore and test Azure AI Language features.
  + Azure Speech Studio: Explore and test Azure AI Speech features.
  + Services are available for use and testing in various programming languages.

**Document Intelligence**

* **Overview**
  + Focuses on managing, processing, and utilizing large volumes of data found in various forms and documents.
  + Automates processing for contracts, health documents, financial forms, and more.
* **Document Intelligence in Microsoft Azure**
  + **Azure AI Document Intelligence**
    - Builds solutions to manage and accelerate data collection from scanned documents.
    - Automates document processing in applications and workflows.
    - Enhances data-driven strategies and enriches document search capabilities.
    - Uses prebuilt models for intelligent document processing (e.g., invoices, receipts, health insurance cards, tax forms).
    - Allows creation of custom models with user-labeled datasets.
    - Features available in Document Intelligence Studio and various programming languages.

**Knowledge Mining**

* **Overview**
  + Extracts information from large volumes of often unstructured data to create a searchable knowledge store.
* **Knowledge Mining in Microsoft Azure**
  + **Azure AI Search**
    - An enterprise search solution for building indexes used for internal or public-facing content.
    - Utilizes Azure AI services (image processing, document intelligence, NLP) to extract data.
    - Indexes previously unsearchable documents and surfaces insights from large data volumes quickly.

**Examples**

* **Document Intelligence Example**
  + Information extracted from a tax form using Azure AI Document Intelligence.
* **Knowledge Mining Example**
  + A travel website using Azure AI Search to enable searches for destination information extracted from images or text using AI services.
* **Generative AI Overview**
  + Creates original content.
  + Interacts via chat applications.
  + Outputs in formats like natural language, image, code, and audio.
* **Generative AI in Microsoft Azure**
  + **Azure OpenAI Service**
    - Deploys, customizes, and hosts generative AI models.
    - Combines OpenAI's models with Azure's security and scalability.
  + **Azure AI Studio**
    - Creates generative AI solutions.
    - Develops custom copilot chat-based assistants.
* **Example**
  + Azure OpenAI Service model generating content, e.g., writing a cover letter.

**AI Challenges and Risks**

* **Bias in Results**
  + Example: A loan-approval model discriminates by gender due to biased training data.
* **Errors and Harm**
  + Example: An autonomous vehicle experiences a system failure and causes a collision.
* **Data Exposure**
  + Example: A medical diagnostic bot is trained using sensitive patient data stored insecurely.
* **Inclusivity Issues**
  + Example: A home automation assistant provides no audio output for visually impaired users.
* **Trust in Complex Systems**
  + Example: Users must understand the basis of recommendations from an AI-based financial tool.
* **Liability for AI Decisions**
  + Example: An innocent person is wrongly convicted due to facial recognition evidence – responsibility is unclear.

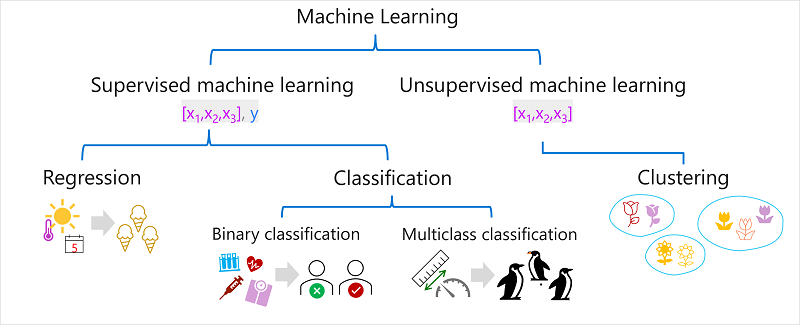
**Responsible AI Principles**

* **Fairness**
  + AI systems must treat all individuals fairly without bias.
  + Example: Loan approval models should not discriminate based on gender, ethnicity, etc.
  + Tools: Azure Machine Learning interprets models to identify and mitigate bias.
* **Reliability and Safety**
  + AI systems must perform reliably and safely.
  + Example: Autonomous vehicles and medical diagnostic tools must be rigorously tested to prevent harm.
* **Privacy and Security**
  + AI systems must secure data and respect privacy.
  + Large data volumes used for training AI models must be protected to ensure privacy and security.
* **Inclusiveness**
  + AI systems should empower all individuals and be inclusive.
  + AI should benefit everyone, regardless of physical ability, gender, sexual orientation, or ethnicity.
* **Transparency**
  + AI systems should be understandable to users.
  + Users should be informed about the system's purpose, functionality, and limitations.
* **Accountability**
  + People must be accountable for AI systems.
  + Designers and developers should follow governance frameworks to ensure ethical and legal standards are met.

Machine learning utilizes past data to predict unknown outcomes or values. It involves training a model using historical observations, where features (x) are related to known labels (y). The model learns a function (f) that maps features to labels. During inferencing, the trained model predicts labels (ŷ) for new input features. The process involves:

* **Training**: Using historical data to define the relationship between features and labels.
* **Inferencing**: Using the trained model to predict labels for new data.

**Types of Machine Learning**



* **Supervised Machine Learning**
  + Uses labeled data (features and known labels).
  + **Regression**
    - Predicts numeric values.
    - Examples: Predicting ice cream sales, property prices, or car fuel efficiency.
  + **Classification**
    - Predicts categories or classes.
    - **Binary Classification**
      * Predicts one of two outcomes (e.g., risk for diabetes, loan default).
    - **Multiclass Classification**
      * Predicts among multiple possible classes (e.g., penguin species, movie genre).
* **Unsupervised Machine Learning**
  + Uses unlabeled data (only features).
  + **Clustering**
    - Identifies similarities between observations and groups them into clusters.
    - Examples: Grouping similar flowers, identifying customer segments based on behavior.
  + Clustering vs. Classification
    - Clustering categorizes observations into groups based on feature similarity.
    - Classification predicts classes based on known labels; clustering can help identify classes before classification.

**Regression Modeling**

Regression models predict numeric label values based on features and known labels. The training process involves multiple iterations:

1. **Data Splitting**: Divide data into training and validation sets.
2. **Model Training**: Fit training data to a model using an algorithm (e.g., linear regression).
3. **Validation**: Use validation data to test the model's predictions.
4. **Evaluation**: Compare predicted labels to actual labels to assess accuracy.

**Regression evaluation metrics**

Based on the differences between the predicted and actual values, you can calculate some common metrics that are used to evaluate a regression model.

**Mean Absolute Error (MAE)**

The variance in this example indicates by how many ice creams each prediction was wrong. It doesn't matter if the prediction was over or under the actual value (so for example, -3 and +3 both indicate a variance of 3). This metric is known as the absolute error for each prediction, and can be summarized for the whole validation set as the mean absolute error (MAE).

**Mean Squared Error (MSE)**

The mean absolute error metric takes all discrepancies between predicted and actual labels into account equally. However, it may be more desirable to have a model that is consistently wrong by a small amount than one that makes fewer, but larger errors. One way to produce a metric that "amplifies" larger errors by squaring the individual errors and calculating the mean of the squared values. This metric is known as the mean squared error (MSE).

**Root Mean Squared Error (RMSE)**

The mean squared error helps take the magnitude of errors into account, but because it squares the error values, the resulting metric no longer represents the quantity measured by the label. In other words, we can say that the MSE of our model is 6, but that doesn't measure its accuracy in terms of the number of ice creams that were mispredicted; 6 is just a numeric score that indicates the level of error in the validation predictions.

If we want to measure the error in terms of the number of ice creams, we need to calculate the square root of the MSE;

**Coefficient of determination (R2)**

All of the metrics so far compare the discrepancy between the predicted and actual values in order to evaluate the model. However, in reality, there's some natural random variance in the daily sales of ice cream that the model takes into account. In a linear regression model, the training algorithm fits a straight line that minimizes the mean variance between the function and the known label values

**Binary Classification**

Binary classification predicts one of two possible labels for a single class (e.g., true/false). The iterative process of training, validating, and evaluating models is similar to regression. Here's an example using blood glucose levels to predict diabetes:

**Binary Classification Evaluation Metrics**

In binary classification, several metrics evaluate the performance of a model in predicting positive and negative cases. Here are the key metrics:

1. **Accuracy**: Measures the proportion of correct predictions out of the total predictions.

Accuracy=TP+TNTP+TN+FP+FN\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}Accuracy=TP+TN+FP+FNTP+TN​

* + **TP (True Positives)**: Correctly predicted positive cases.
  + **TN (True Negatives)**: Correctly predicted negative cases.
  + **FP (False Positives)**: Incorrectly predicted positive cases.
  + **FN (False Negatives)**: Incorrectly predicted negative cases.

1. **Recall (True Positive Rate, Sensitivity)**: Measures the proportion of actual positive cases that were correctly identified by the model.

Recall=TPTP+FN\text{Recall} = \frac{TP}{TP + FN}Recall=TP+FNTP​

1. **Precision**: Measures the proportion of predicted positive cases that were actually positive.

Precision=TPTP+FP\text{Precision} = \frac{TP}{TP + FP}Precision=TP+FPTP​

1. **F1-score**: The harmonic mean of precision and recall, providing a balance between the two metrics.

F1-score=2×Precision×RecallPrecision+Recall\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}F1-score=Precision+Recall2×Precision×Recall​

1. **Area Under the Curve (AUC)**: Represents the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate against the false positive rate at various threshold settings.

These metrics provide insights into the model's performance in binary classification tasks, helping assess its effectiveness in distinguishing between positive and negative cases.

*Multiclass classification* is used to predict to which of multiple possible classes an observation belongs. As a supervised machine learning technique, it follows the same iterative *train, validate, and evaluate* process as regression and binary classification in which a subset of the training data is held back to validate the trained model.

To train a multiclass classification model, we need to use an algorithm to fit the training data to a function that calculates a probability value for each possible class. There are two kinds of algorithm you can use to do this:

* One-vs-Rest (OvR) algorithms
* Multinomial algorithms

**Clustering with K-Means Algorithm**

Clustering is an unsupervised machine learning technique used to group similar observations together based on their features. One of the popular clustering algorithms is K-Means clustering, which follows these steps:

1. **Vectorization of Features**: Each observation's features are vectorized to define n-dimensional coordinates, where n is the number of features. For example, if we have two features - number of leaves and number of petals, the feature vector would have two coordinates: [number of leaves, number of petals].
2. **Choosing the Number of Clusters (k)**: Determine the number of clusters you want to create. This value, denoted as k, is predefined by the user. For instance, if you want to create three clusters, you set k = 3.
3. **Initializing Centroids**: Randomly select k data points from the dataset as initial centroids. These centroids represent the center points of the clusters.
4. **Assigning Points to Nearest Centroids**: Each data point is assigned to the nearest centroid based on a distance metric, typically Euclidean distance.
5. **Moving Centroids**: Calculate the mean of the data points assigned to each centroid. Move the centroid to the calculated mean position.
6. **Reassigning Points to Clusters**: After moving centroids, reassign each data point to the nearest centroid.
7. **Iterative Optimization**: Steps 5 and 6 are repeated iteratively until either the centroids no longer move significantly or a maximum number of iterations is reached.

K-Means clustering aims to minimize the within-cluster sum of squares, effectively creating clusters where the data points within each cluster are as close to each other as possible and as far as possible from data points in other clusters.

This process results in distinct clusters of data points, each characterized by the centroid and containing observations that are similar to each other based on the features used for clustering. K-Means clustering is widely used in various domains, including customer segmentation, image compression, and anomaly detection.

**Deep Learning: Understanding and Training Neural Networks**

Deep learning is an advanced form of machine learning that mimics the learning process of the human brain. At the core of deep learning is the creation of artificial neural networks, which simulate the electrochemical activity in biological neurons using mathematical functions.

**Anatomy of an Artificial Neural Network:**

* **Neurons**: Neurons in an artificial neural network operate on input values (x) and weights (w), applying mathematical functions to produce an output. Each neuron typically includes an activation function that determines whether to propagate the output to the next layer.
* **Layers**: Artificial neural networks consist of multiple layers of neurons, forming a deeply nested function. These layers include an input layer, one or more hidden layers, and an output layer. The input layer receives the feature values, while the output layer produces the predicted values or probabilities for each class.
* **Activation Function**: An activation function introduces non-linearity into the neural network, allowing it to learn complex patterns in the data. Common activation functions include sigmoid, tanh, and ReLU (Rectified Linear Unit).

**Training a Deep Neural Network:**

1. **Forward Propagation**: During training, the feature values (x) are fed forward through the layers of the neural network, producing predicted values (ŷ).
2. **Loss Calculation**: A loss function compares the predicted values (ŷ) to the actual labels (y), quantifying the difference between them. Common loss functions include mean squared error for regression tasks and cross-entropy loss for classification tasks.
3. **Backpropagation**: Backpropagation is the process of updating the weights of the neural network to minimize the loss. It involves computing the gradient of the loss function with respect to each weight and adjusting the weights using optimization techniques such as gradient descent.
4. **Optimization**: Optimization algorithms, such as stochastic gradient descent (SGD) or Adam, iteratively adjust the weights of the neural network to minimize the loss function. These algorithms update the weights based on the computed gradients and learning rate.
5. **Epochs**: The training process iterates over multiple epochs, where each epoch involves one pass through the entire training dataset. Training continues until the model achieves satisfactory performance on the validation dataset or reaches a predefined number of epochs.

**Example: Using Deep Learning for Classification:**

Consider a scenario where a deep neural network is trained to classify penguin species based on features such as bill length, bill depth, flipper length, and weight.

1. **Input Layer**: Feature values (x) representing penguin measurements are fed into the input layer of the neural network.
2. **Hidden Layers**: The data passes through one or more hidden layers, where neurons apply weighted functions and activation functions to produce intermediate outputs.
3. **Output Layer**: The output layer produces a vector of probabilities for each class, indicating the likelihood of the penguin belonging to each species.
4. **Inference**: During inference, the penguin's features are input into the trained neural network, and the output layer predicts the most probable species based on the highest probability value.

Deep learning enables complex pattern recognition and can be applied to various machine learning tasks, including regression, classification, natural language processing, and computer vision. Through iterative training and optimization, deep neural networks can learn to make accurate predictions from raw data, making them powerful tools for data-driven decision-making.

 **Azure Machine Learning**: Cloud service for managing machine learning projects.

 **Key Features**:

* **Data Exploration and Preparation**: Centralized storage and management of datasets.
* **Model Training and Evaluation**: On-demand compute resources for training and evaluation.
* **Model Management**: Register and manage trained models, track versions and performance metrics.
* **Model Deployment**: Deploy trained models for integration with applications and services.
* **Responsible AI Practices**: Built-in support for model explainability and fairness assessment.
* **AutoML**: Automated machine learning for model selection and hyperparameter tuning.
* **Orchestrated Pipelines**: Define pipelines using visual tools for streamlined processes.
* **Integration with ML Frameworks**: Seamless integration with common ML frameworks like MLflow.

 **Provisioning Azure Machine Learning**:

* Create an Azure Machine Learning workspace in an Azure subscription.
* Supporting resources like storage accounts are created automatically as needed.

 **Azure Machine Learning Studio**:

* Browser-based portal for managing ML resources and jobs.
* Features include data import, compute resource management, notebook execution, automated ML, model deployment, and more.
* Visualize evaluation metrics, explore responsible AI information, deploy models, and manage model catalog.

 **Azure AI Services Overview**:

* Azure offers a range of AI capabilities for integration into web or mobile applications.
* Services include image recognition, natural language processing, speech-to-text, text-to-speech, AI-powered search, and more.

 **Examples of Azure AI Services**:

* Azure AI Content Safety: Detects harmful content in text or images.
* Azure AI Language: Summarizes text, classifies information, extracts key phrases.
* Azure AI Speech: Converts speech to text and text to speech accurately.

 **Principles of Azure AI Services**:

* Prebuilt and ready to use: Utilizes pre-trained machine learning models for accessibility.
* Accessed through APIs: Can be integrated into applications with minimal coding.
* Available on Azure: Managed as Azure resources, consistent with other Azure services.

 **Customization and Use Cases**:

* Customization: Some services can be tailored to specific requirements.
* Use Cases: Examples include education with Immersive Reader and sports performance analysis with Azure AI Vision.

 **Accessing Azure AI Services**:

* APIs: Accessed through REST APIs, client libraries, or integration with tools like Logic Apps and Power Automate.
* Azure Integration: Managed as Azure resources, consistent with other Azure services.

**Azure AI Service Resources**:

* **Multi-service Resource**:
  + Created in the Azure portal.
  + Provides access to multiple Azure AI services with a single key and endpoint.
  + Suitable for scenarios requiring several AI services or exploring AI capabilities.
  + All AI services are billed together when using this resource.
* **Single-service Resources**:
  + Created in the Azure portal.
  + Provides access to a single Azure AI service (e.g., Speech, Vision, Language, etc.).
  + Each AI service has a unique key and endpoint.
  + Useful when only one AI service is required or for separating cost information.

**Using Azure AI Services**:

* **Building Applications**:
  + Utilize REST API, software development kits (SDKs), or visual studio interfaces.
* **Studio Interfaces**:
  + Offer a user-friendly interface for exploring Azure AI services.
  + Different studios available for various AI services like Vision, Language, Speech, etc.
  + Test out services using provided samples or experiment with custom content.
* **Associating AI Service Resource**:
  + Before usage, associate the AI service resource with the desired studio on the Settings page.
  + Select the resource and choose "Use Resource" to associate it.
* **Example: Azure AI Content Safety Service**:
  + Identifies harmful text or images.
  + To explore, create a multi-service Azure AI services resource or a single-service Content Safety resource.
  + On the Content Safety Studio Settings page, associate the resource by selecting it and choosing "Use Resource".

**Authentication for Azure AI Services**:

* **Verification Process**:
  + Authenticate users or services to ensure authorized access to the AI service.
* **API Access**:
  + Most Azure AI services are accessed via a RESTful API, defining the information exchange between the service and its users.
* **Authentication Mechanism**:
  + Requests made to AI service resources must be authenticated, typically using an endpoint and a resource key.
* **Endpoint**:
  + Specifies how to reach the AI service resource instance.
  + Found in the Azure portal under Resource Management > Keys and Endpoint.
* **Resource Key**:
  + Protects the privacy of the resource and can be changed periodically for security.
* **Authentication Header**:
  + When accessing the service through code, include the endpoint and keys in the authentication header.
* **Studio Interface Authentication**:
  + Credentials are authenticated during sign-in, ensuring secure access to Azure AI services.