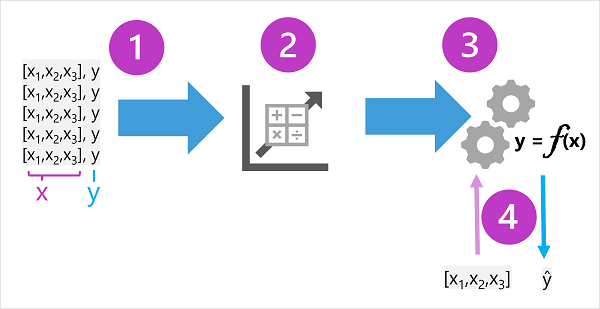
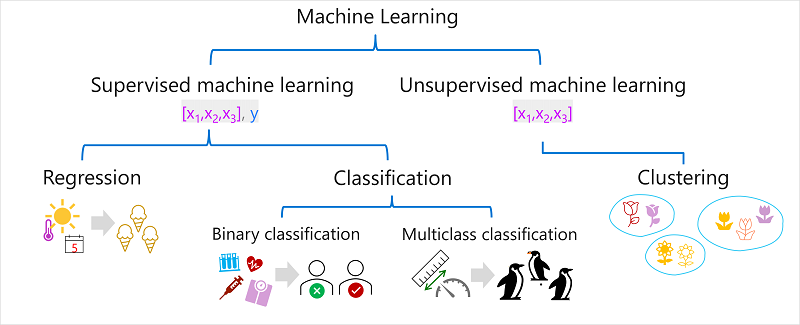
Module 2: Machine Learning (ML)

# What is machine learning?

* The fundamental idea of machine learning is to use past data to predict unknown outcomes or values.
  + Training = defining the function (that will calculate an output)
  + Inferencing = using the training model (function) to predict new values.
* 
  + 1) Observations (features or attributes) --> features (x), labels (y)
    - **Ex.** In the ice cream sales scenario,
      * Train a model to predict the number of ice cream sales based on the weather.
      * The weather measurements for a day (temperature, rainfall, windspeed, and so on) = features (x)
      * # of ice creams sold on each day = label (y).
  + 2) Algorithm (determines relationships between features and label)
  + 3) Results of algo = MODEL (encapsulates the calculation derived by algo as a function
  + 4) Model can now be used for inferencing

# Types of ML

* 

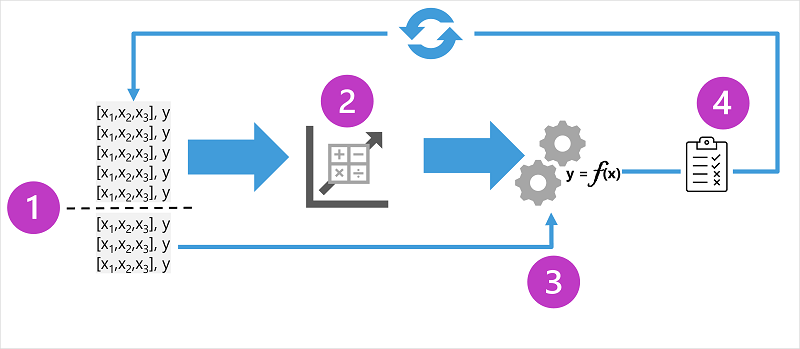
# Supervised ML

* A term for ML algos where training data includes both *feature* values and known *label* values
  + Supervised ML trains models by determining relationships between features and labels to predict unknown labels
* **Regression**
  + Type of Supervised ML where the label predicted by model is **numeric**
    - **Ex**. The *number of ice creams* sold on a given day, based on the temperature, rainfall, and windspeed.
* **Classification**
  + Form of supervised machine learning in which the label represents a categorization, or class
  + Two Scenarios:
    - 1. Binary Classification (the label determines if observed item **IS/ISN’T**)
      * Binary classification models predict one of two mutually exclusive outcomes.
      * Model predicts a **binary true/false** or **positive/negative** prediction for a 1 class.
        + **Ex**. Whether a bank customer will default on a loan based on income, credit history, age.
    - 2. Multi-class
      * Model predicts a label that represents one of multiple possible classes
        + Ex. Determine genre of a movie (comedy, horror, romance, adventure, or science fiction) based on its cast, director, and budget.

# Unsupervised ML

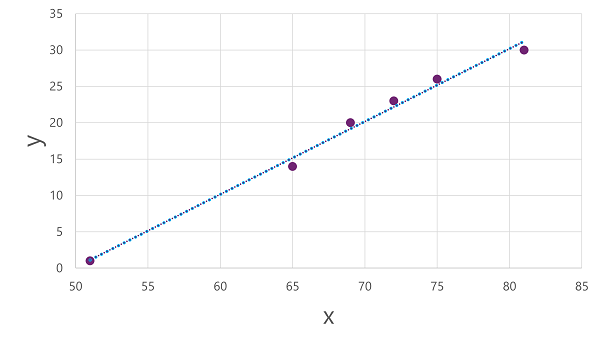
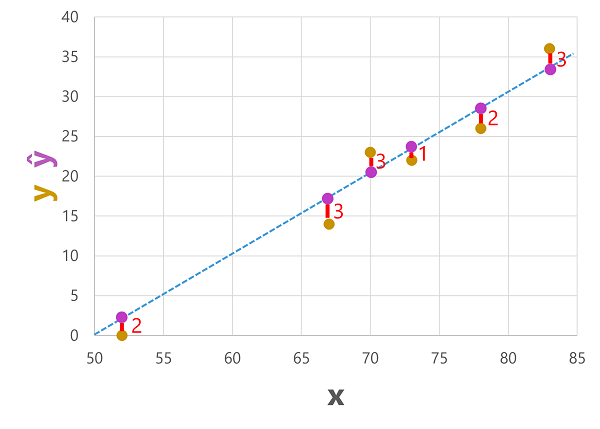
* Training models using data that consists only of **feature** values without any known labels. This model seeks to determine relationships between features only
  + **Ex**. Group similar flowers based on their size, number of leaves, and number of petals.
* In clustering, there's no previously known cluster label and the algorithm groups the data observations based purely on similarity of features.

## Regression Models

* 
  1. Split the training data (randomly) to create a dataset with which to train the model while holding back a subset of the data that you'll use to validate the trained model.
  2. Use an algorithm (such as linear regression) to fit the training data to a model.
  3. Use the validation data you held back to test the model by predicting labels for the features.
  4. Compare the known actual labels in the validation dataset to the labels that the model predicted.
     1. Then aggregate differences between the predicted and actual label values (to calculate a metric)
  5. Repeat

**Example**:

Ice Cream Sales and Daily Temperature

* Features = input; Labels = output
* After plotting a dataset onto a graph, you then use Linear Regression algorithm to create the FUNCTION (f(x) = x-50.
  + You can use this to predict ice creams based on temperature
* Then compare this to the ‘validation’ data set to see the differences
  + 

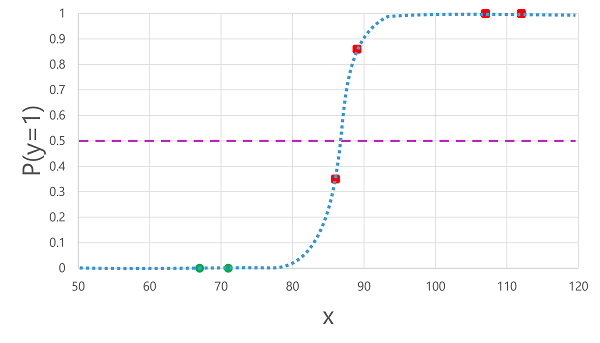
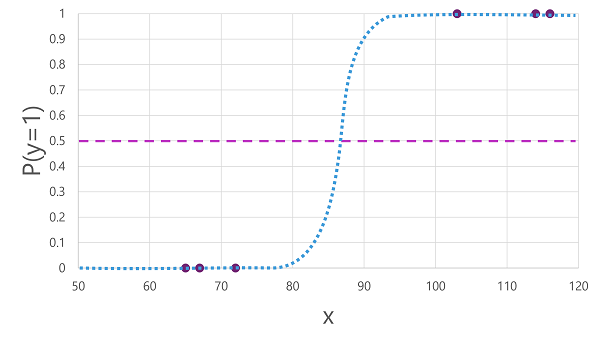
## Regression Evaluation Metrics

1. **Mean Absolute Error (MAE)**
   * **Average** Difference between predicted labels and actual values
2. **Mean Squared Error (MSE)**
   * Can be desirable to have a model that is consistently wrong by a small amount VS. one that makes fewer, but larger errors.
   * Do this by squaring the individual errors and calculating the mean of the squared values to amplify larger errors.
3. **Root MSE**
   * MSE shows magnitude of errors.. but not # of ice creams mispredicted
   * RMSE does this
4. **Coefficient of determination (R2 or R-Squared)**
   1. Metric that measures the *proportion of variance* in the validation results based on the model INSTEAD of some random aspect of the validation data
      1. Ex. a day with a highly unusual number of ice creams sales because of a local festival).

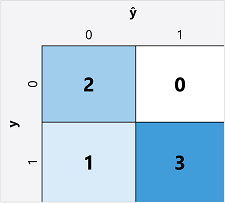
# Binary classification

* Binary classification algorithms are used to train a model that predicts one of two possible labels for a single class. Essentially, predicting true or false.

**Example**: *Does blood glucose level of a patient to predict whether or not the patient has diabetes?*

* 
* Using *logistic regression* algorithm to determine sigmoid (s-func) checking for 0 or 1
* Then compare to validation dataset to determine True vs. False

### Binary Classification Evaluation Metrics

* Create a Confusion Matrix
  + 
    - ŷ=0 and y=0: True negatives (TN)
    - ŷ=1 and y=0: False positives (FP)
    - ŷ=0 and y=1: False negatives (FN)
    - ŷ=1 and y=1: True positives (TP)
* **Accuracy** = calculate from the confusion matrix using the proportion of predictions that model got right
  + (TN+TP) ÷ (TN+FN+FP+TP)
* **Recall** = measures the proportion of positive cases that the model identified correctly
  + TP ÷ (TP+FN)
* **Precision** = measures the proportion of predicted positive cases where the true label is actually positive.
  + How many predicted to have diabetes ACTUALLY have diabetes?

# Multi-class Classification

* Used to calculate probability values for multiple class labels (model can predict the most probable class for a given observation).
  + **Ex**. Based on flipper length (x – features), you can observed which bird species it is (y – label).
* 2 Types of Algorithms for Multi-class Classification Model
  + **One v. Rest (OvR) Algorithm**
    - Train a binary classification function for each class/possiblity
      * **Ex**. 3 Sigmoid Functions per class
        + Bird Type A --> f0(x) = P(y=0 | x)
        + Bird Type B --> f1(x) = P(y=1 | x)
        + Bird Type C --> f2(x) = P(y=2 | x)
  + **Multinomial Algorithm**
    - Use single function returning a multi-valued output (a vector (an array of values) containing the probability distribution for all possible classes)
    - **Ex**. f(x) =[P(y=0|x), P(y=1|x), P(y=2|x)]
      * *Soft max function*, which could produce an output like the following example:
        + [0.2, 0.3, 0.5]

# Clustering

* A form of unsupervised machine learning in which observations are grouped into clusters based on similarities in their data values, or features

**Example:** Botanist observes a sample of flowers and records the number of leaves and petals on each flower

* No labels only 2 features (goal is to group flowers based on # of petals and leaves)

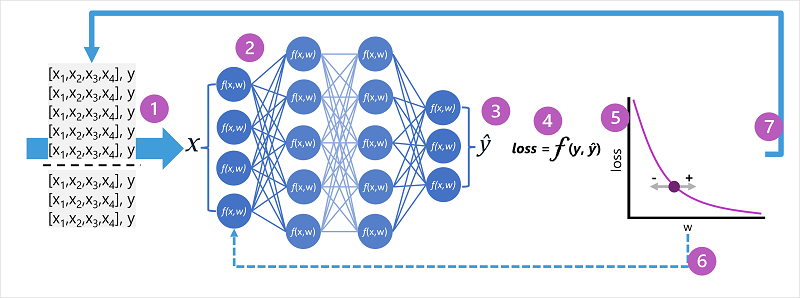
## Training a Clustering Model

* ***K-Means* clustering**, which consists of the following steps:
  + Vectorize the features (x) to define n-dimensional coordinates (where n is the number of features).
    - **Ex**. 2 features: number of leaves (x1) and number of petals (x2).
      * Feature vector = ([x1,x2])
  + You decide how many clusters you want to use to group the flowers – call this value k.
    - Ex. k value of 3 == three clusters
      * Then k points are plotted at random coordinates called *centroids*.
  + Each data point (in this case a flower) is assigned to its nearest centroid.
  + Each centroid is moved to the centre of the data points assigned to it
  + After the centroid is moved, the data points may now be closer to a different centroid, so the data points are reassigned to clusters based on the new closest centroid.
  + Repeat moving centroids and cluster reallocations until the clusters become stable or a predetermined maximum number of iterations is reached.

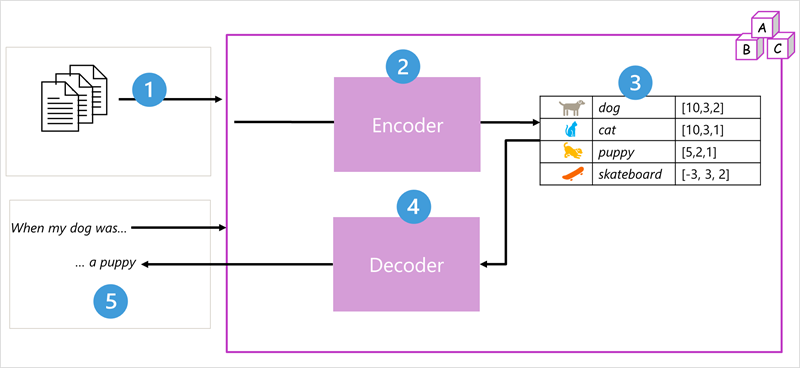
## Evaluation of Clustering

* Evaluation is based on how well the resulting clusters are separated from one another.
* Metrics:
  + Average distance to cluster centre: How close, on average, each point in the cluster is to the centroid of the cluster.
  + Average distance to other centre: How close, on average, each point in the cluster is to the centroid of all other clusters.
  + Maximum distance to cluster centre: The furthest distance between a point in the cluster and its centroid.
  + Silhouette: A value between -1 and 1 that summarizes the ratio of distance between points in the same cluster and points in different clusters (The closer to 1, the better the cluster separation).

# Deep Learning

* Form of ML that tries to emulate the way the human brain learns (artificial neural network that simulates electrochemical activity in biological neurons by using mathematical functions)
* Penguin Example: <https://learn.microsoft.com/en-us/training/modules/fundamentals-machine-learning/8-deep-learning>
* 

# Transformer models

* Transformer models with enough vocabulary are capable of generating human-like language responses.
* Transformer model architecture consists of two components, or blocks:
  + An **encoder** block that creates semantic representations of the training vocabulary.
  + A **decoder** block that generates new language sequences.
* 
  + Train model with a large volume of natural language text
  + The sequences of text are broken down into tokens (**ex**. individual words) and the encoder block processes these token sequences to determine relationships between tokens (**attention technique**)
    - Ex: which tokens influence the presence of other tokens in a sequence, different tokens that are commonly used in the same context, and so on.)
  + Output from the encoder is a collection of vectors (multi-valued numeric arrays) in which each element of the vector represents a semantic attribute of the tokens (**embeddings**).
  + The decoder block works on a new sequence of text tokens and uses the embeddings to generate an appropriate natural language output.

## Tokenization

* Break text into unique values (ex. Letters)

## Embeddings

* Create semantic relationships between the tokens using contextual vectors, known as embeddings