|  |  |
| --- | --- |
| what is ML? | Machine learning (ML) is a modern software development technique and a type of artificial intelligence (AI) that enables computers to solve problems by using examples of real-world data. It allows computers to automatically learn and improve from experience without being explicitly programmed to do so.  Machine learning is part of the broader field of artificial intelligence. This field is concerned with the capability of machines to perform activities using human-like intelligence. Within machine learning there are several different kinds of tasks or techniques: |
| Supervised learning | In supervised learning, every training sample from the dataset has a corresponding label or output value associated with it. As a result, the algorithm learns to predict labels or output values. |
| Unsupervised learning  Reinforcement learning | In unsupervised learning, there are no labels for the training data. A machine learning algorithm tries to learn the underlying patterns or distributions that govern the data.  In reinforcement learning, the algorithm figures out which actions to take in a situation to maximize a reward (in the form of a number) on the way to reaching a specific goal. This is a completely different approach than supervised and unsupervised learning. |
| *Terminology:*  **Machine learning** | Machine learning, or ML, is a modern software development technique that enables computers to solve problems by using examples of real-world data. |
| **Supervised learning** | In supervised learning, every training sample from the dataset has a corresponding label or output value associated with it. As a result, the algorithm learns to predict labels or output values. |
| **Reinforcement learning** | In reinforcement learning, the algorithm figures out which actions to take in a situation to maximize a reward (in the form of a number) on the way to reaching a specific goal. |
| **Unsupervised learning** | In unsupervised learning, there are no labels for the training data. A machine learning algorithm tries to learn the underlying patterns or distributions that govern the data |
| Components of Machine Learning | Nearly all tasks solved with machine learning involve three primary components:  • A machine learning model  • A model training algorithm  • A model inference algorithm |
| What are machine learning models? | A machine learning model, like a piece of clay, can be molded into many different forms and serve many different purposes. A more technical definition would be that a machine learning model is a block of code or framework that can be modified to solve different but related problems based on the data provided.  **\*\*Important\*\***  *A model is an extremely generic program (or block of code), made specific by the data used to train it. It is used to solve different problems.* |
| Example 1 | Imagine you own a snow cone cart, and you have some data about the average number of snow cones sold per day based on the high temperature. You want to better understand this relationship to make sure you have enough inventory on hand for those high sales days.  In the graph above, you can see one example of a model, a linear regression model (indicated by the solid line). You can see that, based on the data provided, the model predicts that as the high temperate for the day increases so do the average number of snow cones sold. |
| Example 2 | Let us look at a different example that uses the same linear regression model, but with different data and to answer completely different questions.  Imagine that you work in higher education, and you want to better understand the relationship between the cost of enrollment and the number of students attending college.  In this example, our model predicts that as the cost of tuition increases the number of people attending college is likely to decrease.    Using the same linear regression model (indicated by the solid line), you can see that the number of people attending college does go down as the cost increases. |
| Conclusion: | Both examples showcase that a model is a generic program made specific by the data used to train it. |
| How are model training algorithms used to train a model? | In the preceding section, we talked about two key pieces of information: a model and data. In this section, we show you how those two pieces of information are used to create a trained model. This process is called *model training.* |
| Model training algorithms work through an interactive process | * Think about the changes that need to be made. The first thing you would do is inspect the raw clay and think about what changes can be made to make it look more like a teapot. Similarly, a model training algorithm uses the model to process data and then compares the results against some end goal, such as our clay teapot. * Make those changes. Now, you mold the clay to make it look more like a teapot. Similarly, a model training algorithm gently nudges specific parts of the model in a direction that brings the model closer to achieving the goal. * Repeat. By iterating over these steps over and over, you get closer and closer to what you want until you determine that you are close enough that you can stop. |
| Model Inference: Using Your Trained Model  Five Machine Learning Steps | Now you have our completed teapot. You inspected the clay, evaluated the changes that needed to be made, and made them, and now the teapot is ready for you to use. Enjoy your tea!  So, what does this mean from a machine learning perspective? We are ready to use the model inference algorithm to generate predictions using the trained model. This process is often referred to as **model inference.** |
| Major Steps in the Machine Learning Process | In the preceding diagram, you can see an outline of the major steps of the machine learning process. Regardless of the specific model or training algorithm used, machine learning practitioners practice a common workflow to accomplish machine learning tasks.  These steps are iterative. In practice, that means that at each step along the process, you review how the process is going. Are things operating as you expected? If not, go back and revisit your current step or previous steps to try and identify the breakdown. |
| **Step One** | |
| Define the Problem | * Define a very specific task. * Think back to the snow cone sales example. Now imagine that you own a frozen treats store, and you sell snow cones along with many other products. You wonder, "‘How do I increase sales?" It's a valid question, but it's the opposite of a very specific task. The following examples demonstrate how a machine learning practitioner might attempt to answer that question. * “Does adding a $1.00 charge for sprinkles on a hot fudge sundae increase the sales of hot fudge sundaes?” * “Does adding a $0.50 charge for organic flavors in your snow cone increase the sales of snow cones?” * Identify the machine learning task we might use to solve this problem. * This helps you better understand the data you need for a project. |
| What is a Machine Learning Task? | All model training algorithms, and the models themselves, take data as their input. Their outputs can be very different and are classified into a few different groups based on the task they are designed to solve. Often, we use the kind of data required to train a model as part of defining a machine learning task.  In this lesson, we will focus on two common machine learning tasks:   * Supervised learning * Unsupervised learning |
| Supervised and Unsupervised Learning  Supervised tasks | The presence or absence of labeling in your data is often used to identify a machine learning task.  Diagram  Description automatically generated  A task is supervised if you are using labeled data. We use the term labeled to refer to data that already contains the solutions, called labels.  *For example: Predicting the number of snow cones sold based on the temperatures is an example of supervised learning.*  Chart, scatter chart  Description automatically generated  In the preceding graph, the data contains both a temperature and the number of snow cones sold. Both components are used to generate the linear regression shown on the graph. Our goal was to predict the number of snow cones sold, and we feed that value into the model. We are providing the model with labeled data and therefore, we are performing a supervised machine learning task. |
| Unsupervised tasks | A task is considered to be unsupervised if you are using unlabeled data. This means you don't need to provide the model with any kind of label or solution while the model is being trained.  Let's take a look at unlabeled data.  A picture containing text, sky, outdoor  Description automatically generatedA picture containing outdoor, sky, plant, tree  Description automatically generated   * Take a look at the preceding picture. Did you notice the tree in the picture? What you just did, when you noticed the object in the picture and identified it as a tree, is called labeling the picture. Unlike you, a computer just sees that image as a matrix of pixels of varying intensity. * Since this image does not have the labeling in its original data, it is considered unlabeled. |
| How do we classify tasks when we don't have a label? | Unsupervised learning involves using data that doesn't have a label. One common task is called clustering. Clustering helps to determine if there are any naturally occurring groupings in the data.  Let's look at an example of how clustering in unlabeled data works.... |
| Identifying book micro-genres with unsupervised learning | Imagine that you work for a company that recommends books to readers.  The assumption: You are fairly confident that micro-genres exist, and that there is one called Teen Vampire Romance. Because you don’t know which micro-genres exist, you can't use supervised learning techniques.  This is where the unsupervised learning clustering technique might be able to detect some groupings in the data. The words and phrases used in the book description might provide some guidance on a book's micro-genre. |
| Further Classifying by using Label Types | Diagram  Description automatically generated  Initially, we divided tasks based on the presence or absence of labeled data while training our model. Often, tasks are further defined by the type of label which is present.  In supervised learning, there are two main identifiers you will see in machine learning:   * A categorical label has a discrete set of possible values. In a machine learning problem in which you want to identify the type of flower based on a picture, you would train your model using images that have been labeled with the categories of flower you would want to identify. Furthermore, when you work with categorical labels, you often carry out classification tasks\*, which are part of the supervised learning family. * A continuous (regression) label does not have a discrete set of possible values, which often means you are working with numerical data. In the snow cone sales example, we are trying to predict the number\* of snow cones sold. Here, our label is a number that could, in theory, be any value.   In unsupervised learning, clustering is just one example. There are many other options, such as deep learning |
| *Terminology:* **Clustering** | **Clustering**. Unsupervised learning task that helps to determine if there are any naturally occurring groupings in the data. |
| **Categorical label** | A ***categorical label*** has a discrete set of possible values, such as "is a cat" and "is not a cat." |
| **Continuous (regression) label** | A **continuous (regression) label** does not have a discrete set of possible values, which means possibly an unlimited number of possibilities. |
| **Discrete** | **Discrete**: A term taken from statistics referring to an outcome taking on only a finite number of values (such as days of the week). |
| **Label** | A **label** refers to data that already contains the solution. |
| **Unlabeled** | Using **unlabeled** data means you don't need to provide the model with any kind of label or solution while the model is being trained. |
| **Step Two** | |
| Build a Dataset | This step involves the machine learning process is to build a dataset that can be used to solve your machine learning-based problem. Understanding the data needed helps you select better models and algorithms so you can build more effective solutions.  **The most important step of the machine learning process**  Working with data is perhaps the most overlooked—yet most important—step of the machine learning process. In 2017, an O’Reilly study showed that machine learning practitioners spend 80% of their time working with their data. |
| The Four Aspects of Working with Data | Diagram  Description automatically generated  You can take an entire class just on working with, understanding, and processing data for machine learning applications. Good, high-quality data is essential for any kind of machine learning project. Let's explore some of the common aspects of working with data. |
| Data collection | Data collection can be as straightforward as running the appropriate SQL queries or as complicated as building custom web scraper applications to collect data for your project. You might even have to run a model over your data to generate needed labels. Here is the fundamental question:  *Does the data you've collected match the machine learning task and problem you have defined?* |
| Data inspection | The quality of your data will ultimately be the largest factor that affects how well you can expect your model to perform. As you inspect your data, look for:   * Outliers * Missing or incomplete values * Data that needs to be transformed or preprocessed so it's in the correct format to be used by your model |
| Summary statistics | Models can assume how your data is structured.  Now that you have some data in hand it is a good best practice to check that your data is in line with the underlying assumptions of your chosen machine learning model.  With many statistical tools, you can calculate things like the mean, inner-quartile range (IQR), and standard deviation. These tools can give you insight into the scope, scale, and shape of the dataset. |
| Data visualization | You can use data visualization to see outliers and trends in your data and to help stakeholders understand your data.  Look at the following two graphs. In the first graph, some data seems to have clustered into different groups. In the second graph, some data points might be outliers.  Chart, scatter chart, bubble chart  Description automatically generated  Some of the data points seem to be outliers    Some of the data seems to cluster in group Some of the data points seem to be outliers |
| Terminology:  ***Impute*** | Impute is a common term referring to different statistical tools which can be used to calculate missing values from your dataset. |
| **Outliers** | Outliers are data points that are significantly different from others in the same sample. |
| **Step Three**: Model Training | |
| Splitting your Dataset | The first step in model training is to randomly split the dataset. This allows you to keep some data hidden during training, so that data can be used to evaluate your model before you put it into production. Specifically, you do this to test against the bias-variance trade-off. If you are interested in learning more, see the Further learning and reading section.  Splitting your dataset gives you two sets of data:   * Training dataset: The data on which the model will be trained. Most of your data will be here. Many developers estimate about 80%. * Test dataset: The data withheld from the model during training, which is used to test how well your model will generalize to new data. |
| Model Training Terminology | The model training algorithm iteratively updates a model's parameters to minimize some loss function.  Let's define those two terms:   * Model parameters: Model parameters are settings or configurations the training algorithm can update to change how the model behaves. Depending on the context, you’ll also hear other more specific terms used to describe model parameters such as weights and biases. Weights, which are values that change as the model learns, are more specific to neural networks. * Loss function: A loss function is used to codify the model’s distance from this goal. For example, if you were trying to predict a number of snow cone sales based on the day’s weather, you would care about making predictions that are as accurate as possible. So you might define a loss function to be “the average distance between your model’s predicted number of snow cone sales and the correct number.” You can see in the snow cone example this is the difference between the two purple dots. |
| Putting it All Together | The end-to-end training process is   * Feed the training data into the model. * Compute the loss function on the results. * Update the model parameters in a direction that reduces loss.   You continue to cycle through these steps until you reach a predefined stop condition. This might be based on a training time, the number of training cycles, or an even more intelligent or application-aware mechanism. |
| Advice From the Experts | Remember the following advice when training your model.   1. Practitioners often use machine learning frameworks that already have working implementations of models and model training algorithms. You could implement these from scratch, but you probably won't need to do so unless you’re developing new models or algorithms. 2. Practitioners use a process called model selection to determine which model or models to use. The list of established models is constantly growing, and even seasoned machine learning practitioners may try many different types of models while solving a problem with machine learning. 3. Hyperparameters are settings on the model which are not changed during training but can affect how quickly or how reliably the model trains, such as the number of clusters the model should identify. 4. Be prepared to iterate.   Pragmatic problem solving with machine learning is rarely an exact science, and you might have assumptions about your data or problem which turn out to be false. Don’t get discouraged. Instead, foster a habit of trying new things, measuring success, and comparing results across iterations. |
| Extended Learning:  **Linear models** | One of the most common models covered in introductory coursework, linear models simply describe the relationship between a set of input numbers and a set of output numbers through a linear function (think of y = mx + b or a line on a x vs y chart).  Classification tasks often use a strongly related logistic model, which adds an additional transformation mapping the output of the linear function to the range [0, 1], interpreted as “probability of being in the target class.” Linear models are fast to train and give you a great baseline against which to compare more complex models. A lot of media buzz is given to more complex models, but for most new problems, consider starting with a simple model. |
| **Tree-based models** | Tree-based models are probably the second most common model type covered in introductory coursework. They learn to categorize or regress by building an extremely large structure of nested if/else blocks, splitting the world into different regions at each if/else block. Training determines exactly where these splits happen and what value is assigned at each leaf region.  For example, if you’re trying to determine if a light sensor is in sunlight or shadow, you might train tree of depth 1 with the final learned configuration being something like if (sensor\_value > 0.698), then return 1; else return 0;. The tree-based model XGBoost is commonly used as an off-the-shelf implementation for this kind of model and includes enhancements beyond what is discussed here. Try tree-based models to quickly get a baseline before moving on to more complex models. |
| **Deep learning models** | Extremely popular and powerful, deep learning is a modern approach based around a conceptual model of how the human brain functions. The model (also called a *neural network*) is composed of collections of *neurons* (very simple computational units) connected together by *weights* (mathematical representations of how much information to allow to flow from one neuron to the next). The process of training involves finding values for each weight.  Various neural network structures have been determined for modeling different kinds of problems or processing different kinds of data.  A short (but not complete!) list of noteworthy examples includes:   * **FFNN**: The most straightforward way of structuring a neural network, the Feed Forward Neural Network (FFNN) structures neurons in a series of layers, with each neuron in a layer containing weights to all neurons in the previous layer. * **CNN**: Convolutional Neural Networks (CNN) represent nested filters over grid-organized data. They are by far the most commonly used type of model when processing images. * **RNN**/**LSTM**: Recurrent Neural Networks (RNN) and the related Long Short-Term Memory (LSTM) model types are structured to effectively represent *for loops* in traditional computing, collecting state while iterating over some object. They can be used for processing sequences of data. * **Transformer**: A more modern replacement for RNN/LSTMs, the transformer architecture enables training over larger datasets involving sequences of data. |
| Machine Learning Using Python Libraries | * For more classical models (linear, tree-based) as well as a set of common ML-related tools, take a look at scikit-learn. The web documentation for this library is also organized for those getting familiar with space and can be a great place to get familiar with some extremely useful tools and techniques. * For deep learning, mxnet, tensorflow, andpytorch are the three most common libraries. For the purposes of the majority of machine learning needs, each of these is feature-paired and equivalent. |
| Terminology:  **Hyperparameters** | Hyperparameters are settings on the model which are not changed during training but can affect how quickly or how reliably the model trains, such as the number of clusters the model should identify. |
| **Loss function** | A loss function is used to codify the model’s distance from this goal. |
| **Training dataset** | Training dataset: The data on which the model will be trained. Most of your data will be here. |
| **Test dataset** | Test dataset: The data withheld from the model during training, which is used to test how well your model will generalize to new data. |
| **Model parameters** | Model parameters are settings or configurations the training algorithm can update to change how the model behaves. |
| **Step Four** | |
| Model Evaluation | After you have collected your data and trained a model, you can start to evaluate how well your model is performing. The metrics used for evaluation are likely to be very specific to the problem you have defined. As you grow in your understanding of machine learning, you will be able to explore a wide variety of metrics that can enable you to evaluate effectively. |
| Using Model Accuracy  Extended Learning:  **Using Log Loss** | Model accuracy is a fairly common evaluation metric. Accuracy is the fraction of predictions a model gets right.  Here's an example:  A picture containing logo  Description automatically generated  Petal length to determine species  Imagine that you built a model to identify a flower as one of two common species based on measurable details like petal length. You want to know how often your model predicts the correct species. This would require you to look at your model's accuracy.  Log loss seeks to calculate how uncertain your model is about the predictions it is generating. In this context, uncertainty refers to how likely a model thinks the predictions being generated are to be correct.  A picture containing icon  Description automatically generated  For example, let's say you're trying to predict how likely a customer is to buy either a jacket or t-shirt.  Log loss could be used to understand your model's uncertainty about a given prediction. In a single instance, your model could predict with 5% certainty that a customer is going to buy a t-shirt. In another instance, your model could predict with 80% certainty that a customer is going to buy a t-shirt. Log loss enables you to measure how strongly the model believes that its prediction is accurate.  In both cases, the model predicts that a customer will buy a t-shirt, but the model's certainty about that prediction can change. |
| Remember: This Process is Iterative | Diagram  Description automatically generated  Iterative steps of machine learning  Every step we have gone through is highly iterative and can be changed or re-scoped during the course of a project. At each step, you might find that you need to go back and reevaluate some assumptions you had in previous steps. Don't worry! This ambiguity is normal. |
| Terminology:  **Log loss** | Log loss seeks to calculate how uncertain your model is about the predictions it is generating. |
| **Model Accuracy**  **Some method to help evaluating our models** | Model Accuracy is the fraction of predictions a model gets right.  Diagram  Description automatically generated   1. Mean absolute error (MAE): This is measured by taking the average of the absolute difference between the actual values and the predictions. Ideally, this difference is minimal. 2. Root mean square error (RMSE): This is similar MAE, but takes a slightly modified approach so values with large error receive a higher penalty. RMSE takes the square root of the average squared difference between the prediction and the actual value. 3. Coefficient of determination or R-squared (R^2): This measures how well-observed outcomes are actually predicted by the model, based on the proportion of total variation of outcomes. |

|  |  |
| --- | --- |
| **Supervised Learning Example** | |
| **House Price Prediction** | House price prediction is one of the most common examples used to introduce machine learning.  Traditionally, real estate appraisers use many quantifiable details about a home (such as number of rooms, lot size, and year of construction) to help them estimate the value of a house.  You detect this relationship and believe that you could use machine learning to predict home prices.  Diagram  Description automatically generated  Machine language models to determine house values |
| Step One: Define the Problem | *Can we estimate the price of a house based on lot size or the number of bedrooms?*  Diagram  Description automatically generatedYou access the sale prices for recently sold homes or have them appraised. Since you have this data, this is a supervised learning task. You want to predict a continuous numeric value, so this task is also a regression task.  Regression task |
| Step Two: Building a Dataset  Step Three: Model Training | * Data collection: You collect numerous examples of homes sold in your neighborhood within the past year and pay a real estate appraiser to appraise the homes whose selling price is not known. * Data exploration: You confirm that all of your data is numerical because most machine learning models operate on sequences of numbers. If there is textual data, you need to transform it into numbers. You'll see this in the next example. * Data cleaning: Look for things such as missing information or outliers, such as the 10-room mansion. Several techniques can be used to handle outliers, but you can also just remove those from your dataset.   Table  Description automatically generated  Data cleaning: removing outlier values   * Chart, scatter chart    Description automatically generatedData visualization: You can plot home values against each of your input variables to look for trends in your data. In the following chart, you see that when lot size increases, the house value increases.   Regression line of a model  Prior to actually training your model, you need to split your data. The standard practice is to put 80% of your dataset into a training dataset and 20% into a test dataset.  **Linear model selection** As you see in the preceding chart, when lot size increases, home values increase too. This relationship is simple enough that a linear model can be used to represent this relationship.  A linear model across a single input variable can be represented as a line. It becomes a plane for two variables, and then a hyperplane for more than two variables. The intuition, as a line with a constant slope, does not change.  **Using a Python library** The Python scikit-learn library has tools that can handle the implementation of the model training algorithm for you. |
| Step Four: Evaluation | A picture containing text, clock  Description automatically generatedOne of the most common evaluation metrics in a regression scenario is called root mean square or RMS. The math is beyond the scope of this lesson, but RMS can be thought of roughly as the "average error” across your test dataset, so you want this value to be low.  The math behind RMS  In the following chart, you can see where the data points are in relation to the blue line. You want the data points to be as close to the "average" line as possible, which would mean less net error.  You compute the root mean square between your model’s prediction for a data point in your test dataset and the true value from your data. This actual calculation is beyond the scope of this lesson, but it's good to understand the process at a high level.  Chart, scatter chart  Description automatically generated  **Interpreting Results**  In general, as your model improves, you see a better RMS result. You may still not be confident about whether the specific value you’ve computed is good or bad.  Many machine learning engineers manually count how many predictions were off by a threshold (for example, $50,000 in this house pricing problem) to help determine and verify the model's accuracy. |
| Step Five: Inference: Try out your model | Now you are ready to put your model into action. As you can see in the following image, this means seeing how well it predicts with new data not seen during model training.  Diagram  Description automatically generated |
| Terminology:  **Continuous** | Continuous: Floating-point values with an infinite range of possible values. The opposite of categorical or discrete values, which take on a limited number of possible values. |
| **Hyperplane** | Hyperplane: A mathematical term for a surface that contains more than two planes. |
| **Plane** | Plane: A mathematical term for a flat surface (like a piece of paper) on which two points can be joined by a straight line. |
| **Regression** | Regression: A common task in supervised machine learning. |
| **Unsupervised Learning example** | |
| Book Genre Exploration | Machine learning process can be applied to an unsupervised machine learning task that uses book description text to identify different micro-genres. |
| Step One: Define the Problem | Diagram  Description automatically generated  Model used to predict micro-genres  *Find clusters of similar books based on the presence of common words in the book descriptions.*  You do editorial work for a book recommendation company, and you want to write an article on the largest book trends of the year. You believe that a trend called "micro-genres" exists, and you have confidence that you can use the book description text to identify these micro-genres.  By using an unsupervised machine learning technique called clustering, you can test your hypothesis that the book description text can be used to identify these "hidden" micro-genres.  Earlier in this lesson, you were introduced to the idea of unsupervised learning. This machine learning task is especially useful when your data is not labeled.  Diagram  Description automatically generated  Unsupervised learning using clustering |
| Step Two: Build your Dataset  Step Three: Train the Model | To test the hypothesis, you gather book description text for 800 romance books published in the current year.  Data exploration, cleaning and preprocessing  For this project, you believe capitalization and verb tense will not matter, and therefore you remove capitals and convert all verbs to the same tense using a Python library built for processing human language. You also remove punctuation and words you don’t think have useful meaning, like 'a' and 'the'. The machine learning community refers to these words as stop words.  Before you can train the model, you need to do some data preprocessing, called data vectorization, to convert text into numbers.  You transform this book description text into what is called a bag of words representation shown in the following image so that it is understandable by machine learning models.  Diagram  Description automatically generatedHow the bag of words representation works is beyond the scope of this course. If you are interested in learning more, see the Additional Reading section at the bottom of the page.  Now you are ready to train your model.  You pick a common cluster-finding model called k-means. In this model, you can change a model parameter, k, to be equal to how many clusters the model will try to find in your dataset.  Your data is unlabeled: you don't how many microgenres might exist. So you train your model multiple times using different values for k each time.  What does this even mean? In the following graphs, you can see examples of when k=2 and when k=3.  Shape, circle  Description automatically generatedDiagram, schematic  Description automatically generated    K = 2 K = 3  During the model evaluation phase, you plan on using a metric to find which value for k is most appropriate. |
| Step Four: Model Evaluation | Chart  Description automatically generatedIn machine learning, numerous statistical metrics or methods are available to evaluate a model. In this use case, the silhouette coefficient is a good choice. This metric describes how well your data was clustered by the model. To find the optimal number of clusters, you plot the silhouette coefficient as shown in the following image below. You find the optimal value is when k=19.  Optimum number (k=19) of clusters  Often, machine learning practitioners do a manual evaluation of the model's findings.  You find one cluster that contains a large collection of books you can categorize as “paranormal teen romance.” This trend is known in your industry, and therefore you feel somewhat confident in your machine learning approach. You don’t know if every cluster is going to be as cohesive as this, but you decide to use this model to see if you can find anything interesting about which to write an article. |
| Step Five: Inference (Use the Model) | As you inspect the different clusters found when k=19, you find a surprisingly large cluster of books. Here's an example from fictionalized cluster #7.  Table  Description automatically generated  Clustered data |
| Terminology:  **Bag of words** | Bag of words: A technique used to extract features from the text. It counts how many times a word appears in a document (corpus), and then transforms that information into a dataset. |
| **Data vectorization** | Data vectorization: A process that converts non-numeric data into a numerical format so that it can be used by a machine learning model. |
| **Silhouette coefficient** | Silhouette coefficient: A score from -1 to 1 describing the clusters found during modeling. A score near zero indicates overlapping clusters, and scores less than zero indicate data points assigned to incorrect clusters. A score approaching 1 indicates successful identification of discrete non-overlapping clusters. |
| **Stop words** | **Stop words**: A list of words removed by natural language processing tools when building your dataset. There is no single universal list of stop words used by all-natural language processing tools. |
| **Reinforcement Learning Example** | |
| Spill Detection from Video | In the previous two examples, we used classical methods like linear models and k-means to solve machine learning tasks. In this example, we’ll use a more modern model type.  Note: This example uses a neural network. The algorithm for how a neural network works is beyond the scope of this lesson. However, there is still value in seeing how machine learning applies in this case. |
| Step One: Defining the Problem | Imagine you run a company that offers specialized on-site janitorial services. A client, an industrial chemical plant, requires a fast response for spills and other health hazards. You realize if you could automatically detect spills using the plant's surveillance system, you could mobilize your janitorial team faster.  Machine learning could be a valuable tool to solve this problem.  Diagram  Description automatically generated  Detecting spills with machine learning |
| Step Two: Model Training (and selection) | This task is a supervised classification task, as shown in the following image. As shown in the image above, your goal will be to predict if each image belongs to one of the following classes:   * Contains spill * Does not contain spill   Diagram  Description automatically generated  Image classification |
| Step Two: Building a Dataset | * Collecting   + Using historical data, as well as safely staged spills, you quickly build a collection of images that contain both spills and non-spills in multiple lighting conditions and environments. * Exploring and cleaning   + You go through all the photos to ensure the spill is clearly in the shot. There are Python tools and other techniques available to improve image quality, which you can use later if you determine a need to iterate. * Data vectorization (converting to numbers)   + Many models require numerical data, so all your image data needs to be transformed into a numerical format. Python tools can help you do this automatically.   + In the following image, you can see how each pixel in the image on the left can be represented in the image on the right by a number between 0 and 1, with 0 being completely black and 1 being completely white.   Background pattern  Description automatically generated  Icon  Description automatically generated      Chemical spill image Numeric representation of chemical spill image  **Split the data**   * You split your image data into a training dataset and a test dataset. |
| Step Three: Model Training | Traditionally, solving this problem would require hand-engineering features on top of the underlying pixels (for example, locations of prominent edges and corners in the image), and then training a model on these features.  Today, deep neural networks are the most common tool used for solving this kind of problem. Many deep neural network models are structured to learn the features on top of the underlying pixels so you don’t have to learn them. You’ll have a chance to take a deeper look at this in the next lesson, so we’ll keep things high-level for now. |
| CNN (convolutional neural network) | Neural networks are beyond the scope of this lesson, but you can think of them as a collection of very simple models connected together. These simple models are called neurons, and the connections between these models are trainable model parameters called weights.  Convolutional neural networks are a special type of neural network particularly good at processing images. |
| Step Four: Model Evaluation | As you saw in the last example, there are many different statistical metrics you can use to evaluate your model. As you gain more experience in machine learning, you will learn how to research which metrics can help you evaluate your model most effectively. Here's a list of common metrics:   |  |  |  | | --- | --- | --- | | **Accuracy** | **False positive rate** | **Precision** | | Confusion matrix | False negative rate | Recall | | F1 Score | Log Loss | ROC curve | |  | Negative predictive value | Specificity |   In cases such as this, accuracy might not be the best evaluation mechanism.  Why not? You realize the model will see the 'Does not contain spill' class almost all the time, so any model that just predicts “no spill” most of the time will seem pretty accurate.  *What you really care about is an evaluation tool that rarely misses a real spill.*  After doing some internet sleuthing, you realize this is a common problem and that Precision and Recall will be effective. You can think of precision as answering the question, "Of all predictions of a spill, how many were right?" and recall as answering the question, "Of all actual spills, how many did we detect?"  Manual evaluation plays an important role. You are unsure if your staged spills are sufficiently realistic compared to actual spills. To get a better sense how well your model performs with actual spills, you find additional examples from historical records. This allows you to confirm that your model is performing satisfactorily. |
| Step Five: Model Inference | The model can be deployed on a system that enables you to run machine learning workloads such as AWS Panorama.  Thankfully, most of the time, the results will be from the class 'Does not contain spill.'  Text, whiteboard  Description automatically generated  No spill detected  Diagram  Description automatically generatedBut, when the class 'Contains spill' is detected, a simple paging system could alert the team to respond.  Spill detected |
| Terminology:  **Convolutional neural networks(CNN)** | Convolutional neural networks(CNN) are a special type of neural network particularly good at processing images. |
| **Neural networks** | Neural networks: a collection of very simple models connected together.   * These simple models are called neurons * The connections between these models are trainable model parameters called weights. |