Lesson 1

[**Artificial Intelligence**](https://en.wikipedia.org/wiki/Artificial_intelligence)**:**In computer science, **artificial intelligence** (**AI**), sometimes called **machine intelligence**, is intelligence demonstrated by machines, in contrast to the **natural intelligence** displayed by humans. Leading AI textbooks define the field as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving".

Computer science defines AI research as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. A more elaborate definition characterizes AI as “a system’s ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation.”

[**Machine Learning**](https://en.wikipedia.org/wiki/Machine_learning)**:** is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning is a subfield of soft computing within computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. In 1959, Arthur Samuel defined machine learning as a "field of study that gives computers the ability to learn without being explicitly programmed". Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Such algorithms operate by building a model from an example training set of input observations in order to make data-driven predictions or decisions expressed as outputs, rather than following strictly static program instructions.

[**Deep Learning**](https://en.wikipedia.org/wiki/Deep_learning)**:** (also known as **deep structured learning** or **hierarchical learning**) is part of a broader family of machine learning methods based on artificial neural networks. Learning can be supervised, semi-supervised or unsupervised.

Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.

[**Algorithms**](https://en.wikipedia.org/wiki/Algorithm)**:** In mathematics and computer science, an **algorithm** is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation. Algorithms are unambiguous specifications for performing calculation, data processing, automated reasoning, and other tasks.

**Ground Truth:**Information or data provided by direct observation (i.e., empirical evidence) as opposed to information or data provided by inference.

[**Supervised Learning**](https://en.wikipedia.org/wiki/Supervised_learning)**:**is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from *labeled training data* consisting of a set of *training examples*. In supervised learning, each example is a *pair* consisting of an input object (typically a vector) and a desired output value (also called the *supervisory signal*). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way (see inductive bias).

[**Unsupervised Learning**](https://en.wikipedia.org/wiki/Unsupervised_learning)**:**is a type of self-organized Hebbian learning that helps find previously unknown patterns in data set without pre-existing labels. It is also known as self-organization and allows modeling probability densities of given inputs. It is one of the main three categories of machine learning, along with supervised and reinforcement learning. Semi-supervised learning has also been described, and is a hybridization of supervised and unsupervised techniques.

Lesson 2

[**Data Preprocessing**](https://en.wikipedia.org/wiki/Data_pre-processing)**:**is an important step in the data mining process. The phrase "garbage in, garbage out" is particularly applicable to data mining and machine learning projects. Data-gathering methods are often loosely controlled, resulting in out-of-range values (e.g., Income: −100), impossible data combinations (e.g., Sex: Male, Pregnant: Yes), missing values, etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running an analysis. Often, data preprocessing is the most important phase of a machine learning project, especially in computational biology.

[**CRISP-DM Model**](https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining): Cross-industry standard process for data mining, known as CRISP-DM, is an open standard process model that describes common approaches used by data mining experts. It is the most widely-used analytics model.

[**Data Set Splitting**](https://guide.freecodecamp.org/machine-learning/dataset-splitting/)[1]**:** Dataset Splitting emerges as a necessity to eliminate bias to training data in ML algorithms. Modifying parameters of a ML algorithm to best fit the training data commonly results in an overfit algorithm that performs poorly on actual test data. For this reason, we split the dataset into multiple, discrete subsets on which we train different parameters.

[**Data Scaling**](https://en.wikipedia.org/wiki/Feature_scaling): Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step.

[**Data Standardizing**](https://machinelearningmastery.com/normalize-standardize-machine-learning-data-weka/)[2]: Data standardization is the process of rescaling one or more attributes so that they have a mean value of 0 and a standard deviation of 1.

[**Data Normalization**](https://machinelearningmastery.com/normalize-standardize-machine-learning-data-weka/)[3]: Data normalization is the process of rescaling one or more attributes to the range of 0 to 1. This means that the largest value for each attribute is 1 and the smallest value is 0.

[**Categorical Variables**](https://en.wikipedia.org/wiki/Categorical_variable): In statistics, a **categorical variable** is a variable that can take on one of a limited, and usually fixed, number of possible values, assigning each individual or other unit of observation to a particular group or nominal category on the basis of some qualitative property. In computer science and some branches of mathematics, categorical variables are referred to as enumerations or enumerated types. Commonly (though not in this article), each of the possible values of a categorical variable is referred to as a **level**. The probability distribution associated with a random categorical variable is called a categorical distribution.

[**Nominal Categories**](https://en.wikipedia.org/wiki/Nominal_category): A **nominal category** or a **nominal group** is a group of objects or ideas that can be collectively grouped on the basis of a particular characteristic—a qualitative property. A variable that codes whether each one in a set of observations is in a particular nominal category is called a categorical variable.

[**Ordinal Categories**](https://en.wikipedia.org/wiki/Ordinal_data): **Ordinal data** is a categorical, statistical data type where the variables have natural, ordered categories and the distances between the categories is not known. These data exist on an **ordinal scale**, one of four levels of measurement described by S. S. Stevens in 1946. The ordinal scale is distinguished from the nominal scale by having a ranking. It also differs from interval and ratio scales by not having category widths that represent equal increments of the underlying attribute.

[**Label Encoding**](https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/)[4]: refers to an important pre-processing step for the structured dataset in supervised learning that converts the labels into numeric form so as to convert it into the machine-readable form and machine learning algorithms can then decide in a better way on how those labels must be operated.

[**One Hot Encoding**](https://en.wikipedia.org/wiki/One-hot): In digital circuits and machine learning, **one-hot** is a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0). A similar implementation in which all bits are '1' except one '0' is sometimes called **one-cold**. In statistics, dummy variables represent a similar technique for representing categorical data.

One-hot encoding is often used for indicating the state of a state machine. When using binary or Gray code, a decoder is needed to determine the state. A one-hot state machine, however, does not need a decoder as the state machine is in the *n*th state if and only if the *n*th bit is high.

[**Ordinal Encoding**](https://contrib.scikit-learn.org/categorical-encoding/ordinal.html)[5]: Ordinal encoding uses a single column of integers to represent the classes under which an optional mapping dict can be passed in, in which case, we use the knowledge that there is some true order to the classes themselves, however in other cases, the classes are assumed to have no true order and integers are selected at random.

Guiding Questions

Develop your answers to the following guiding questions while watching lectures and working on assignments throughout the module.

1. What is the relationship between artificial intelligence and machine learning?
2. What is the difference between supervised and unsupervised learning?
3. Why data pre-processing is necessary?
4. What data pre-processing technics are for categorical features and what are for continuous features?

Lesson 3

[**Classification**](https://en.wikipedia.org/wiki/Statistical_classification): In the terminology of machine learning, classification is considered an instance of supervised learning, i.e., learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance.

[**Regression**](https://en.wikipedia.org/wiki/Regression_analysis): In statistical modeling, **regression analysis** is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates', or 'features').

[**Text Analysis**](https://en.wikipedia.org/wiki/Content_analysis):**Content analysis** is a research method for studying documents and communication artifacts, which might be texts of various formats, pictures, audio or video. Social scientists use content analysis to examine patterns in communication in a replicable and systematic manner. One of the key advantages of using content analysis to analyze social phenomena is its non-invasive nature, in contrast to simulating social experiences or collecting survey answers.

By systematically labeling the content of a set of texts, researchers can analyze patterns of content quantitatively using statistical methods, or use qualitative methods to analyze meanings of content within texts.

[**Clustering**](https://en.wikipedia.org/wiki/Cluster_analysis): **Cluster analysis** or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved.

[**Time Series**](https://en.wikipedia.org/wiki/Time_series): A **time series** is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.

**Time series *analysis*** comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data.

Module 2

Lesson 1

[**Linear Regression**](https://en.wikipedia.org/wiki/Linear_regression): In statistics, linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

[**Simple Linear Regression**](https://en.wikipedia.org/wiki/Simple_linear_regression): In statistics, **simple linear regression** is a linear regression model with a single explanatory variable.That is, it concerns two-dimensional sample points with one independent variable and one dependent variable (conventionally, the *x* and *y* coordinates in a Cartesian coordinate system) and finds a linear function (a non-vertical straight line) that, as accurately as possible, predicts the dependent variable values as a function of the independent variables. The adjective *simple* refers to the fact that the outcome variable is related to a single predictor.

[**Dummy Variables**](https://en.wikipedia.org/wiki/Dummy_variable_(statistics)): In statistics and econometrics, particularly in regression analysis, a dummy variable is one that takes only the value 0 or 1 to indicate the absence or presence of some categorical effect that may be expected to shift the outcome. They can be thought of as numeric stand-ins for qualitative facts in a regression model, sorting data into mutually exclusive categories (such as smoker and non-smoker).

Lesson 2

[**Logistic Regression**](https://en.wikipedia.org/wiki/Logistic_regression): In statistics, the logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1 and the sum adding to one.

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is**estimating the parameters of a logistic model (a form of binary regression).**

[**Confusion Matrix**](https://en.wikipedia.org/wiki/Confusion_matrix): In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

It is a special kind of contingency table, with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions (each combination of dimension and class is a variable in the contingency table).

Lesson 3

[**Decision Tree**](https://en.wikipedia.org/wiki/Decision_tree): A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

Module 3

Lesson 1

[**K-nearest neighbors**](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm): In pattern recognition, the *k*-nearest neighbors algorithm (*k*-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

Lesson 2

[**Support Vector Machine**](https://en.wikipedia.org/wiki/Support-vector_machine): In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

Lesson 3

[**Bagging**](https://en.wikipedia.org/wiki/Bagging): In statistics, data mining and machine learning, bootstrap aggregating

[**Random Forest**](https://en.wikipedia.org/wiki/Random_forest): Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Module 4

Lesson 1

[**Mean Absolute Error (MAE)**](https://en.wikipedia.org/wiki/Mean_absolute_error): In statistics, mean absolute error (MAE) is a measure of difference between two continuous variables. Assume *X* and *Y* are variables of paired observations that express the same phenomenon. Examples of *Y* versus *X* include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement. Consider a scatter plot of *n* points, where point *i* has coordinates (*xi*, *yi*)... Mean Absolute Error (MAE) is the average vertical distance between each point and the identity line. MAE is also the average horizontal distance between each point and the identity line.

[**Mean Squared Error (MSE)**](https://en.wikipedia.org/wiki/Mean_squared_error): In statistics, the mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss. The fact that MSE is almost always strictly positive (and not zero) is because of randomness or because the estimator does not account for information that could produce a more accurate estimate.

The MSE is a measure of the quality of an estimator—it is always non-negative, and values closer to zero are better.

[**Root Mean Squared Error (RMSE)**](https://en.wikipedia.org/wiki/Root-mean-square_deviation): The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. The RMSD represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences. These deviations are called *residuals* when the calculations are performed over the data sample that was used for estimation and are called *errors* (or prediction errors) when computed out-of-sample. The RMSD serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. RMSD is a measure of accuracy, to compare forecasting errors of different models for a particular dataset and not between datasets, as it is scale-dependent.

RMSD is always non-negative, and a value of 0 (almost never achieved in practice) would indicate a perfect fit to the data. In general, a lower RMSD is better than a higher one. However, comparisons across different types of data would be invalid because the measure is dependent on the scale of the numbers used.

RMSD is the square root of the average of squared errors. The effect of each error on RMSD is proportional to the size of the squared error; thus larger errors have a disproportionately large effect on RMSD. Consequently, RMSD is sensitive to outliers.

[**R-squared (*R*2)**](https://en.wikipedia.org/wiki/Coefficient_of_determination): In statistics, the coefficient of determination, denoted *R*2 or *r*2 and pronounced "R squared", is the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

It is a statistic used in the context of statistical models whose main purpose is either the prediction of future outcomes or the testing of hypotheses, on the basis of other related information. It provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model.

There are several definitions of *R*2 that are only sometimes equivalent. One class of such cases includes that of simple linear regression where *r*2 is used instead of *R*2. When an intercept is included, then *r*2 is simply the square of the sample correlation coefficient (i.e., *r*) between the observed outcomes and the observed predictor values. If additional regressors are included, *R*2 is the square of the coefficient of multiple correlation. In both such cases, the coefficient of determination normally ranges from 0 to 1.

Lesson 2

[**Accuracy Score**](https://developers.google.com/machine-learning/crash-course/classification/accuracy)[6]: Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right.

**Precision:**The proportion of the prediction that is correct.

**Recall:**Proportion of actual class that is predicted correctly.

[**Type 1 Error (False Positive)**](https://en.wikipedia.org/wiki/False_positives_and_false_negatives): A false positive error, or in short a false positive, commonly called a "false alarm", is a result that indicates a given condition exists, when it does not. For example, in the case of "The Boy Who Cried Wolf", the condition tested for was "is there a wolf near the herd?"; the shepherd at first wrongly indicated there was one, by calling "Wolf, wolf!"

A false positive error is a type I error where the test is checking a single condition, and wrongly gives an affirmative (positive) decision. However it is important to distinguish between the type 1 error rate and the probability of a positive result being false. The latter is known as the false positive risk (see Ambiguity in the definition of false positive rate, below).

[**Type 2 Error (False Negative)**](https://en.wikipedia.org/wiki/False_positives_and_false_negatives): A false negative error, or in short a false negative, is a test result that indicates that a condition does not hold, while in fact it does. In other words, erroneously, no effect has been inferred. An example for a false negative is a test indicating that a woman is not pregnant whereas she is actually pregnant. Another example is a truly guilty prisoner who is acquitted of a crime. The condition "the prisoner is guilty" holds (the prisoner is indeed guilty). But the test (a trial in a court of law) failed to realize this condition, and wrongly decided that the prisoner was not guilty, falsely concluding a negative about the condition.

A false negative error is a type II error occurring in a test where a single condition is checked for and the result of the test is erroneously that the condition is absent.

Lesson 3

[**ROC (Receiver operating characteristic)**](https://en.wikipedia.org/wiki/Receiver_operating_characteristic): A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or *probability of detection* in machine learning. The false-positive rate is also known as *probability of false alarm* and can be calculated as (1 − specificity). It can also be thought of as a plot of the power as a function of the Type I Error of the decision rule (when the performance is calculated from just a sample of the population, it can be thought of as estimators of these quantities). The ROC curve is thus the sensitivity as a function of fall-out. In general, if the probability distributions for both detection and false alarm are known, the ROC curve can be generated by plotting the cumulative distribution function (area under the probability distribution from {\displaystyle -\infty } to the discrimination threshold) of the detection probability in the y-axis versus the cumulative distribution function of the false-alarm probability on the x-axis.

[**AUC (Area under curve)**](https://en.wikipedia.org/wiki/Receiver_operating_characteristic#Area_under_the_curve): When using normalized units, the area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative').

Module 5

Lesson 1

[**Feature selection**](https://en.wikipedia.org/wiki/Feature_selection): In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for several reasons:

* simplification of models to make them easier to interpret by researchers/users,
* shorter training times,
* to avoid the curse of dimensionality,
* enhanced generalization by reducing overfitting[2] (formally, reduction of variance)

The central premise when using a feature selection technique is that the data contains some features that are either *redundant* or *irrelevant*, and can thus be removed without incurring much loss of information. *Redundant* and *irrelevant* are two distinct notions, since one relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated.

[**Filter methods**](https://en.wikipedia.org/wiki/Feature_selection#Filter_method): Filter type methods select variables regardless of the model. They are based only on general features like the correlation with the variable to predict. Filter methods suppress the least interesting variables. The other variables will be part of a classification or a regression model used to classify or to predict data. These methods are particularly effective in computation time and robust to overfitting.

Filter methods tend to select redundant variables when they do not consider the relationships between variables. However, more elaborate features try and minimize this problem by removing variables highly correlated to each other, such as the FCBF algorithm.

[**Wrapper methods**](https://en.wikipedia.org/wiki/Feature_selection#Wrapper_method): evaluate subsets of variables which allows, unlike filter approaches, to detect the possible interactions between variables. The two main disadvantages of these methods are:

* The increasing overfitting risk when the number of observations is insufficient.
* The significant computation time when the number of variables is large.

[**Embedded methods**](https://en.wikipedia.org/wiki/Feature_selection#Embedded_method): Embedded methods have been recently proposed that try to combine the advantages of both previous methods. A learning algorithm takes advantage of its own variable selection process and performs feature selection and classification simultaneously, such as the FRMT algorithm.

[**Variance Threshold**](https://php-ml.readthedocs.io/en/latest/machine-learning/feature-selection/variance-threshold/)[7]: A simple baseline approach to feature selection. It removes all features whose variance doesn’t meet some threshold. By default, it removes all zero-variance features, i.e. features that have the same value in all samples.

[**Univariate Techniques**](https://www.statisticshowto.datasciencecentral.com/univariate/)[8]: Univariate analysis, the simplest form of analyzing data, is when your data has only one variable. Unlike regression, it doesn’t deal with causes or relationships and it’s major purpose is to describe. Instead, it takes data, summarizes that data and finds patterns in the data.

Lesson 2

[**K-Fold cross-validation**](https://en.wikipedia.org/wiki/Cross-validation_(statistics)#k-fold_cross-validation): In *k*-fold cross-validation, the original sample is randomly partitioned into *k* equal sized subsamples. Of the *k* subsamples, a single subsample is retained as the validation data for testing the model, and the remaining *k* − 1 subsamples are used as training data. The cross-validation process is then repeated *k* times, with each of the *k* subsamples used exactly once as the validation data. The *k* results can then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling (see below) is that all observations are used for both training and validation, and each observation is used for validation exactly once. 10-fold cross-validation is commonly used, but in general *k* remains an unfixed parameter.

[**Stratified K-Fold cross validation**](https://en.wikipedia.org/wiki/Cross-validation_(statistics)#k-fold_cross-validation): In *stratified* *k*-fold cross-validation, the partitions are selected so that the mean response value is approximately equal in all the partitions. In the case of binary classification, this means that each partition contains roughly the same proportions of the two types of class labels.

Lesson 3

[**Grid Search**](https://en.wikipedia.org/wiki/Hyperparameter_optimization#Grid_search):The traditional way of performing hyperparameter optimization has been *grid search*, or a *parameter sweep*, which is simply an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm. A grid search algorithm must be guided by some performance metric, typically measured by cross-validation on the training set or evaluation on a held-out validation set.

Since the parameter space of a machine learner may include real-valued or unbounded value spaces for certain parameters, manually set bounds and discretization may be necessary before applying grid search.

Module 6

Lesson 1

[**Text Tokenization**](https://en.wikipedia.org/wiki/Lexical_analysis#Tokenization): In computer science, lexical analysis, lexing or tokenization is the process of converting a sequence of characters (such as in a computer program or web page) into a sequence of tokens (strings with an assigned and thus identified meaning). A program that performs lexical analysis may be termed a *lexer*, *tokenizer*, or *scanner*, though *scanner* is also a term for the first stage of a lexer. A lexer is generally combined with a parser, which together analyze the syntax of programming languages, web pages, and so forth.

*Tokenization* is the process of demarcating and possibly classifying sections of a string of input characters. The resulting tokens are then passed on to some other form of processing. The process can be considered a sub-task of parsing input.

[**Stop Words**](https://en.wikipedia.org/wiki/Stop_words): In computing, stop words are words which are filtered out before processing of natural language data (text). Stop words are generally the most common words in a language; there is no single universal list of stop words used by all natural language processing tools, and indeed not all tools even use such a list. Some tools avoid removing stop words to support phrase search.

[**Stemming**](https://en.wikipedia.org/wiki/Stemming): In linguistic morphology and information retrieval, stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Algorithms for stemming have been studied in computer science since the 1960s. Many search engines treat words with the same stem as synonyms as a kind of query expansion, a process called conflation.

A computer program or subroutine that stems word may be called a *stemming program*, *stemming algorithm*, or *stemmer*.

[**Bag of Words**](https://en.wikipedia.org/wiki/Bag-of-words_model): The bag-of-words model is a simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity. The bag-of-words model has also been used for computer vision.

The bag-of-words model is commonly used in methods of document classification where the (frequency of) occurrence of each word is used as a feature for training a classifier.

[**Document Term Matrix (DTM)**](https://en.wikipedia.org/wiki/Document-term_matrix): A document-term matrix or term-document matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms. There are various schemes for determining the value that each entry in the matrix should take. One such scheme is tf-idf. They are useful in the field of natural language processing.

[**TF-DIF (Term Frequency-Inverse Document Frequency)**](https://en.wikipedia.org/wiki/Tf%E2%80%93idf): In information retrieval, tf–idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. tf–idf is one of the most popular term-weighting schemes today; 83% of text-based recommender systems in digital libraries use tf–idf.

Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. tf–idf can be successfully used for stop-words filtering in various subject fields, including text summarization and classification.

One of the simplest ranking functions is computed by summing the tf–idf for each query term; many more sophisticated ranking functions are variants of this simple model.

[**CountVectorizer**](https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/)[9]: The CountVectorizer provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary.

Lesson 2

[**The 20 Newsgroups data set**](http://qwone.com/~jason/20Newsgroups/)[10]: The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned nearly evenly across 20 different newsgroups. It is a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering.”

[**Bayes' Theorem**](https://en.wikipedia.org/wiki/Bayes%27_theorem): In probability theory and statistics, **Bayes’s theorem** (alternatively **Bayes’s law** or **Bayes’s rule**) describes the probability of an event, based on prior knowledge of conditions that might be related to the event. For example, if the probability that somoene has cancer is related to their age, using Bayes’ theorem the age can be used to more accurately assess the probability of cancer than can be done without knowledge of the age.

[**Naïve Bayes Classifier**](https://en.wikipedia.org/wiki/Naive_Bayes_classifier): In machine learning, **naïve Bayes classifiers** are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models.

[**Linear SVC**](https://pythonprogramming.net/linear-svc-example-scikit-learn-svm-python/)[11]: The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data.

[**Logistic Regression**](https://en.wikipedia.org/wiki/Logistic_regression): In statistics, the **logistic model** (or **logit model**) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1 and the sum adding to one.

Lesson 3

[**N-grams**](https://en.wikipedia.org/wiki/N-gram): In the fields of computational linguistics and probability, an *n*-gram is a contiguous sequence of *n* items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The *n*-grams typically are collected from a text or speech corpus. When the items are words, *n*-grams may also be called *shingles*.

An ***n*-gram model** models sequences, notably natural languages, using the statistical properties of *n*-grams.

[**Sentiment Analysis**](https://en.wikipedia.org/wiki/Sentiment_analysis): Sentiment analysis (also known as opinion mining or emotion AI) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

[**Stemming**](https://en.wikipedia.org/wiki/Stemming): In linguistic morphology and information retrieval, stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Algorithms for stemming have been studied in computer science since the 1960s. Many search engines treat words with the same stem as synonyms as a kind of query expansion, a process called conflation.

A computer program or subroutine that stems word may be called a *stemming program*, *stemming algorithm*, or *stemmer*.

Module 7

Lesson 1

[***k*-means clustering**](https://en.wikipedia.org/wiki/K-means_clustering): is a method of [vector quantization](https://en.wikipedia.org/wiki/Vector_quantization), originally from signal processing, that is popular for cluster analysis in data mining. *k*-means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. *k*-Means minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. Better Euclidean solutions can for example be found using k-medians and k-medoids.

The algorithm has a loose relationship to the *k*-nearest neighbor classifier, a popular machine learning technique for classification that is often confused with *k*-means due to the name. Applying the 1-nearest neighbor classifier to the cluster centers obtained by *k*-means classifies new data into the existing clusters. This is known as nearest centroid classifier or Rocchio algorithm.

[**Principal Component Analysis (PCA**)](https://en.wikipedia.org/wiki/Principal_component_analysis): is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing *n* observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

[**Elbow method**](https://en.wikipedia.org/wiki/Elbow_method_(clustering)): is a heuristic method of interpretation and validation of consistency within cluster analysis designed to help find the appropriate number of clusters in a dataset. It is often ambiguous and not very reliable, and hence other approaches for determining the number of clusters such as the silhouette method are preferable.

This method looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion". This "elbow" cannot always be unambiguously identified. Percentage of variance explained is the ratio of the between-group variance to the total variance, also known as an F-test. A slight variation of this method plots the curvature of the within group variance.

[**Adjusted Rand Index**](https://en.wikipedia.org/wiki/Rand_index#Adjusted_Rand_index): is the corrected-for-chance version of the Rand index. Such a correction for chance establishes a baseline by using the expected similarity of all pair-wise comparisons between clusterings specified by a random model. Traditionally, the Rand Index was corrected using the Permutation Model for clusterings (the number and size of clusters within a clustering are fixed, and all random clusterings are generated by shuffling the elements between the fixed clusters). However, the premises of the permutation model are frequently violated; in many clustering scenarios, either the number of clusters or the size distribution of those clusters vary drastically. For example, consider that in K-means the number of clusters is fixed by the practitioner, but the sizes of those clusters are inferred from the data. Variations of the adjusted Rand Index account for different models of random clusterings.

Though the Rand Index may only yield a value between 0 and +1, the adjusted Rand index can yield negative values if the index is less than the expected index.

[**Silhouette**](https://en.wikipedia.org/wiki/Silhouette_(clustering)): refers to a method of interpretation and validation of consistency within clusters of data. The technique provides a succinct graphical representation of how well each object has been classified.

The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

The silhouette can be calculated with any distance metric, such as the Euclidean distance or the Manhattan distance.

Lesson 3

[**Density-based clustering**](https://en.wikipedia.org/wiki/Cluster_analysis#Density-based_clustering): In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points.

[**DBSCAN**](https://en.wikipedia.org/wiki/DBSCAN): Density-based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996. It is a density-based clustering non-parametric algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature.

[**Hyperparameter**](https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)): In machine learning, a hyperparameter is a parameter whose value is set before the learning process begins. By contrast, the values of other parameters are derived via training.

Hyperparameters can be classified as model hyperparameters, that cannot be inferred while fitting the machine to the training set because they refer to the model selection task, or algorithm hyperparameters, that in principle have no influence on the performance of the model but affect the speed and quality of the learning process. An example of the first type is the topology and size of a neural network. An example of the second type is learning rate or mini-batch size.

[**Core point**](https://en.wikipedia.org/wiki/DBSCAN#Preliminary): Consider a set of points in some space to be clustered. Let *ε* be a parameter specifying the radius of a neighborhood with respect to some point. For the purpose of DBSCAN clustering, the points are classified as *core points*, (*density*-)*reachable points* and *outliers*, as follows:

At least min\_samples of points within distance eps.

[**Border points**](https://stackoverflow.com/questions/26711352/in-dbscan-how-to-determine-border-points)[12]: Border points are points that are (in DBSCAN) **part of a cluster, but not dense themselves** (i.e. every cluster member that is *not* a core point).

A core point, but within eps to at least one core point.

[**Noise points**](https://en.wikipedia.org/wiki/DBSCAN#Algorithm): DBSCAN requires two parameters: ε (eps) and the minimum number of points required to form a dense region (minPts). It starts with an arbitrary starting point that has not been visited. This point's ε-neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized ε-environment of a different point and hence be made part of a cluster.

Neither core point nor border point.

**Clusters:** All core points that are reachable to each other form a cluster.  All border points of these core points also belong to the cluster.

Module 8

Lesson 2

[**Time Series**](https://en.wikipedia.org/wiki/Time_series#Methods_for_analysis): Time-series: A single variable is captured over a period of time, such as the unemployment rate over a 10-year period. A line chart may be used to demonstrate the trend.

**Time series *analysis*** comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. **Time series *forecasting*** is the use of a model to predict future values based on previously observed values. While regression analysis is often employed in such a way as to test theories that the current values of one or more independent time series affect the current value of another time series, this type of analysis of time series is not called "time series analysis", which focuses on comparing values of a single time series or multiple dependent time series at different points in time.Interrupted time series analysis is the analysis of interventions on a single time series.

[**Rolling Statistics**](https://en.wikipedia.org/wiki/Moving_average): In statistics, a moving average (rolling average or running average) is a calculation to analyze data points by creating a series of averages of different subsets of the full data set. It is also called a moving mean (MM) or rolling mean and is a type of finite impulse response filter. Variations include: simple, and cumulative, or weighted forms (described below).

A moving average is commonly used with time series data to smooth out short-term fluctuations and highlight longer-term trends or cycles. The threshold between short-term and long-term depends on the application, and the parameters of the moving average will be set accordingly.

[**Death Cross**](https://en.wikipedia.org/wiki/Moving_average_crossover): In the statistics of time series, and in particular the analysis of financial time series for stock trading purposes, a moving-average crossover occurs when, on plotting two moving averages each based on different degrees of smoothing, the traces of these moving averages cross. It does not predict future direction but shows trends. This indicator uses two (or more) moving averages, a slower moving average and a faster moving average. The faster moving average is a short term moving average. For end-of-day stock markets, for example, it may be 5-, 10- or 25-day period while the slower moving average is medium or long term moving average (e.g. 50-, 100- or 200-day period). A short term moving average is faster because it only considers prices over short period of time and is thus more reactive to daily price changes. On the other hand, a long term moving average is deemed slower as it encapsulates prices over a longer period and is more lethargic. However, it tends to smooth out price noises which are often reflected in short term moving averages.

[**Resampling**](https://en.wikipedia.org/wiki/Resampling_(statistics)): In statistics, resampling is any of a variety of methods for doing one of the following: 1) Estimating the precision of sample statistics (medians, variances, percentiles) by using subsets of available data (jackknifing) or drawing randomly with replacement from a set of data points (bootstrapping), 2) Exchanging labels on data points when performing significance tests (permutation tests, also called exact tests, randomization tests, or re-randomization tests), 3) Validating models by using random subsets (bootstrapping, cross validation)

[**Decomposition**](https://en.wikipedia.org/wiki/Decomposition_of_time_series): This is an important technique for all types of time series analysis, especially for seasonal adjustment. It seeks to construct, from an observed time series, a number of component series (that could be used to reconstruct the original by additions or multiplications) where each of these has a certain characteristic or type of behavior. For example, time series are usually decomposed into:

* The trend component at time *t*, which reflects the long-term progression of the series (secular variation). A trend exists when there is a persistent increasing or decreasing direction in the data. The trend component does not have to be linear.
* The cyclical component at time *t*, which reflects repeated but non-periodic fluctuations. The duration of these fluctuations is usually of at least two years.
* The seasonal component at time *t*, reflecting seasonality (seasonal variation). A seasonal pattern exists when a time series is influenced by seasonal factors. Seasonality occurs over a fixed and known period (e.g., the quarter of the year, the month, or day of the week).
* The irregular component (or "noise") at time *t*, which describes random, irregular influences. It represents the residuals or remainder of the time series after the other components have been removed.

References

[1] freeCodeCamp: <https://guide.freecodecamp.org/machine-learning/dataset-splitting/>

[2] [3] Jason Brownlee / Machine Learning Mastery / <https://machinelearningmastery.com/normalize-standardize-machine-learning-data-weka/>

[4] aakarsha\_chugh / GeeksforGeeks / <https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/>

[5] Will McGinnis / Category Encoders/ <https://contrib.scikit-learn.org/categorical-encoding/ordinal.html>

[6] © Google Developers / CC BY 4.0 / http://tinyurl.com/yyzrckay

[7] © Arkadiusz Kondas / MIT License / http://tinyurl.com/s8rt44a

[8] Statistics How To / <https://www.statisticshowto.datasciencecentral.com/univariate/>

[9] Jason Brownlee / Machine Learning Mastery / <https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/>

[10] Jason Rennie / <http://qwone.com/~jason/20Newsgroups/>

[11] PythonProgramming.net / <https://pythonprogramming.net/linear-svc-example-scikit-learn-svm-python>

[12] Anony-Mousse -Reinstate Monica / <https://stackoverflow.com/questions/26711352/in-dbscan-how-to-determine-border-points>

In Lesson 1 of the module,

we will discuss the relationship

between artificial intelligence,machine learning, and deep learning.

We'll also introduce the machine learning algorithms

you're going to learn in this course.

I bet you have heard the terms

like artificial intelligence

or AI and deep learning along with machine learning.

Many people use these three terms interchangeably,

but there is a reason why

our course is not called AI for Accounting.

Artificial intelligence was first brought up in 1950s.

Roughly speaking, AI is

a system that mimics the cognitive functions

of humans and carries out

tasks in an intelligent manner like a human.

When we talk about AI, C-3PO

or Terminator always comes up in mind.

They talk, walk, and behave like a human,

and sometimes even express opinions or show emotions.

They are example of so-called strong AI or general AI.

We're not able to establish strong AI yet.

All AI we currently have are

so-called narrow AI or weak AI,

which can handle particular tasks

like a self-driving car.

So AI is the whole system.

But to enable AI to carry out certain tasks,

we rely on machine learning

to make data-driven decisions.

Machine learning is an algorithm

or a technique of parsing data,

learn from that data and then apply

what they have learned to make an informed decision.

For example, a self-driving car will

collect surrounding data with camera, radar,

and GPS, then process this data with

machine learning algorithms and make

decisions on how to maneuver the vehicle.

Deep learning is actually

a special kind of machine learning that is

completely based on artificial neural networks

which mimics the human brain.

Deep learning requires tremendous processing power and as

the processing power increases

exponentially in the last decade,

deep learning came into the picture.

The self-driving car we mentioned above uses

deep learning to process data and make decisions.

While deep learning already has some impacts on

accounting like transcripts of conference calls,

it still has relatively low adoptions in accounting world

partly due to lack of

interpretability of artificial neural networks.

So in this course,

we'll focus on other machine learning algorithms

and their applications in accounting field.

There are two main types of machine learning algorithms,

supervised learning and unsupervised learning.

In supervised learning, the model or the algorithm

is provided with ground truth

or true output for given input.

For example, fraudulent transaction detection

can be implemented with a supervised learning model.

The inputs are transaction details such as amount,

time, location, merchant, etc.

The output is whether a transaction is valid or not,

which we already know for past transactions.

We then feed the inputs and

the outputs into a machine learning model

to train the model.

After the model is trained with the past data,

it can be used to evaluate new transactions at

real-time and predict whether they're throttling or not.

We're going to introduce two types of

supervised learning in this course,

classification and regression.

Classification is used when the output is discrete.

Like in a fraud detection,

the output is either fraudulent or not fraudulent.

Regression is used when the output is continuous,

like predicting sales or

profits based on market conditions.

In unsupervised learning,

corresponding output is not available.

The model tries to identify patterns

and features from the data directly.

For example, unsupervised learning can be used to divide

customers into different groups by

their profiles and purchasing habits.

This customer segmentation can help merchants to better

serve their customers or

promote their products more effectively.

We're going to introduce clustering

for unsupervised learning in this course.

In this course, we're going to learn

these five types of machine learning algorithms.

Classification and regression

are supervised learning algorithms.

Clustering is an unsupervised learning algorithm.

Text analysis is

a special kind of classification problem.

We introduced it separately

because unlike normal classifications,

the training data of texts analysis is a collection of

texts like customer review or newspaper articles.

Machine learning models can only deal with numeric data.

So we have to first manipulate the text dataset and

convert them to numeric data, then apply classification.

Time series analysis is

another special machine learning algorithm.

Is called time series analysis because the data

is indexed by time like daily stock price,

hourly weather conditions, etc.

Time series analysis tries to find the trend over

time and predict future values

based on previous information.

**This lesson is about data preprocessing**.

To understand data preprocessing, let's review the CRISP-DM framework first.

In the CRISP-DM framework, the third step is data preparation.

We clean up the data, handle missing values,

merge various data sources together in this step.

Play video starting at 31 seconds and follow transcript0:31

After this step, we'll have a clean data set.

But before we can feed the data set to a machine learning model,

we still need to preprocess the data.

Data preprocessing is often misunderstood as part of data preparation,

but the fact is data preprocessing is

part of modeling because it depends on the machine learning models.

Different machine learning models requires different data preprocessing.

Play video starting at 1 minute 6 seconds and follow transcript1:06

In this lesson, we will introduce three data preprocessing techniques,

categorical variable encoding, dataset splitting, and data scaling.

Among them, dataset splitting is for supervised learning only.

Play video starting at 1 minute 23 seconds and follow transcript1:23

Other two are for both supervised and unsupervised learning.

Play video starting at 1 minute 28 seconds and follow transcript1:28

The first data preprocessing technique we introduced is

categorical variable encoding.

There are two types of categorical variables, nominal and ordinal.

A nominal variable has no intrinsic order in its categories.

For example, gender is a nominal categorical variable because

the two categories, male and female, have no intrinsic order.

Some other nominal variables examples are day of the week,

country of origin, color of shirt, etc.

An ordinal variable, on the other hand, has some sort of order, for

example, ranking in a race first place, second place, and third place.

There's an intrinsic order in the categories.

Some other ordinal variable examples are passenger class, shirt or shoe size, etc.

Categorical variables often have text values but

all machine learning algorithms can only work with numeric data.

So we will have to encode categorical variables into numeric values.

For different type of categorical variables and

machine learning models, there are many different encoding techniques.

We introduced three most common encoding techniques in this lesson.

The simplest approach is label encoding which converts

each category into a unique numeric value.

One hot encoding creates extra dummy variables for categorical variable.

Ordinal encoding maps a category value

to a particular numeric value based on its characteristic.

In this slide, we compare label encoding with one hot encoding.

The original dataset has a categorical variable food,

which has three categories, apple, beef, and cabbage.

With label encoding, the categories

are mapped to integer 0, 1, and 2.

The map between category value and numerical value is somewhat random.

Normally, alphabetical order of categorical values is used

to determine which numerical value of category is mapped to.

In this case, apple is mapped to 0.

Beef is mapped to 1.

And cabbage is mapped to 2.

Play video starting at 4 minutes 4 seconds and follow transcript4:04

Label encoding is straightforward, but it has an apparent problem.

Play video starting at 4 minutes 9 seconds and follow transcript4:09

It introduces an order into the categorical value.

Some machine learning algorithms are sensitive

to the values like linear regression.

For this kind of algorithms,

one hot encoding normally provides a better learning result.

With one hot encoding, extra dummy

variables are created, one dummy variable for each category value.

In this example, three dummy variables are created, food-apple,

food-beef, and food-cabbage.

A dummy variable has value 0 or 1.

So for apple, only food-apple has value 1.

With one hot encoding, we don't introduce extra information into the dataset.

The disadvantage of one hot encoding is that it increases

the dataset size, especially

when there are a lot of categorical variables with many different categories.

Some machine learning algorithms don't rely on values of categorical variables

like Decision Tree or Random Forest.

For those kind of models, label encoding is good enough.

Play video starting at 5 minutes 27 seconds and follow transcript5:27

For some other algorithms like linear regression, which is sensitive to

the categorical values, one hot encoding normally provides better results.

Ordinal encoding is used to encode ordinal categorical variables.

With ordinal encoding, a categorical value is mapped

to a particular numerical value based on its characteristic.

For example, in this slide, we map size

large to 2, medium to 1, and small to 0.

Ordinal encoding has the same problem that label encoding has.

In this example, even though large is bigger than medium,

it's not necessarily two times of medium as the numerical value indicates.

Play video starting at 6 minutes 17 seconds and follow transcript6:17

When working with algorithms that are sensitive to categorical values,

Play video starting at 6 minutes 23 seconds and follow transcript6:23

we still prefer one hot encoding over ordinal encoding for

ordinal categorical variables.

Play video starting at 6 minutes 30 seconds and follow transcript6:30

The second data preprocessing technique is dataset splitting which is for

supervised learning only.

Dataset splitting divides the data set into two subsets, training and test.

Training dataset is used to train the model.

Play video starting at 6 minutes 49 seconds and follow transcript6:49

Test set is used to evaluate the model.

To split a dataset, we first import train\_test\_split

from the sklearn.model\_selection module.

In this example, we pass the data and the label to the function.

Play video starting at 7 minutes 11 seconds and follow transcript7:11

The third argument is test\_size which defines

the proportion of the dataset to be included in the test set.

test\_size would be between 0 and 1.

In this case, the test\_size is 0.4,

which means 40% of the data set will be included in the test set and

60% will be included in the training set.

train\_test\_split function will split the dataset randomly.

So every time you run the code, you will get different training and test set.

To ensure that we get the same result every time,

we can set random\_state to a particular value.

The value of random\_state can be any integer.

Now, we will run the code multiple times

every time you will get the same training and test set.

train\_test\_split will return a pair of data object for

each data object passed to the function.

In this case, we passed two data objects, data and label.

So the function returns four data objects, train and test for

data and train and test for label.

After the data and the label are splitted, we can use d\_train and

l\_train to train a model then use d\_test and l\_test to evaluate the model.

The third data preprocessing technique is data scaling.

Many machine learning algorithms are sensitive to the spread of features.

For example, an algorithm might get more weight and features with a larger spread.

Even if this procedure produces a suboptimal result,

to prevent this kind of problems, we scale the features.

We introduce two types of data scaling in this lesson,

standardizing and normalization.

Play video starting at 9 minutes 9 seconds and follow transcript9:09

Standardizing scales data to mean zero and standard deviation one.

Use standardizing when the data is normally distributed.

Play video starting at 9 minutes 21 seconds and follow transcript9:21

Normalization scales data to a particular range, normally from 0 to 1.

Scaling is not required for all machine learning algorithms.

For example, for Decision Tree or Random Forest, we don't need to scale the data.

Play video starting at 9 minutes 38 seconds and follow transcript9:38

We will discuss scaling in more details in the future lessons

when we introduce the specific machine learning algorithms.

We will use standardizing to demonstrate the data scaling process with Python.

To standardize data,

we use the StandardScaler in the sklearn.preprocessing module.

We first create a StandardScaler object then

train the object with the training data.

This is to learn the distribution of the data.

Then we call transform to transform both

training dataset and test dataset.

The reason that we only fit the scalar object with the training data

is that the test data is used for evaluation only.

The distribution of the test dataset should not be used to transform

the dataset.

In this lesson, we will demonstrate how to apply machine learning models with

Python Scikit-Learn module.

We will use the iris dataset to demonstrate three types of machine

learning algorithms.

Classification, regression and clustering.

The iris dataset is a seaborne built-in data set.

The data set consists of 50 samples from each of three species of Iris.

Play video starting at 35 seconds and follow transcript0:35

Four features were measured from each sample.

The length and width of the sepals and petals in centimeters.

Play video starting at 44 seconds and follow transcript0:44

The species column has three different values, setosa, versicolor and virginica.

Play video starting at 52 seconds and follow transcript0:52

It is the output in the classification example.

We will encode the species column to map the species to numeric values.

Since this column serves as the classification output,

we will use label encoder to encode it.

Play video starting at 1 minute 9 seconds and follow transcript1:09

The four iris feature columns will be scaled before feeding

into a machine learning model.

We will also split the data set to training and

test set for the classification and regression examples.

Play video starting at 1 minute 26 seconds and follow transcript1:26

For the clustering example, we will only use the four feature columns.

Since there's no output involved in the clustering.

We don't need to split the data set for clustering since is unsupervised learning.

But we still need to scale the four feature columns.

Play video starting at 1 minute 47 seconds and follow transcript1:47

Python Scikit-Learn library has various machine learning algorithms.

And all of them have similar interface.

Which means we can follow the same steps when we are applying different algorithms.

The steps are, first, pre-process the data,

then create a model then train the model, then evaluate the model.

Finally, predict with the model.

For classification, we demonstrate k-nearest neighbor's classifier.

And for regression we demonstrate a decision tree regressor.

And for clustering we demonstrate a k-means algorithm.

Play video starting at 2 minutes 27 seconds and follow transcript2:27

Don't worry if you don't know anything about these algorithms.

Because we will learn the algorithms in the following lessons.

In this lesson just understand the basic steps

of applying Scikit-Learn machine learning models.

Play video starting at 2 minutes 42 seconds and follow transcript2:42

It's very simple to perform machine learning with Python.

Assume we already processed the data.

You normally just need several lines to apply a machine learning model

on the data as shown in this slide.

This is a sample code to apply

a classification with K nearest neighbor classifier.

Play video starting at 3 minutes 6 seconds and follow transcript3:06

You'll first construct the model with a model specific parameters.

Then you'll train the model on training data set by calling the fit function.

Then predict with a trained model by calling the predict function.

Play video starting at 3 minutes 23 seconds and follow transcript3:23

You can apply all machine learning models defined in the Scikit-Learn module

with similar approach.

Again, don't worry if you don't know anything

about the algorithms demonstrated in this lesson.

We are going to discuss these machine learning algorithms

in detail in the following modules.