



Applying Supervised Learning

```
%% Generalized Linear Model - Logistic Regression  
glm = GeneralizedLinearModel.fit(Xtrain,double(Ytrain),  
    'linear','Distribution','binomial','link','logit');
```

```
%% Discriminant Analysis  
da = ClassificationDiscriminant.fit(Xtrain,Ytrain,  
    'discrimType','quadratic');
```

```
%% Classification Using Nearest Neighbors  
knn = ClassificationKNN.fit(Xtrain,Ytrain,...  
    'Distance','seuclidean');
```

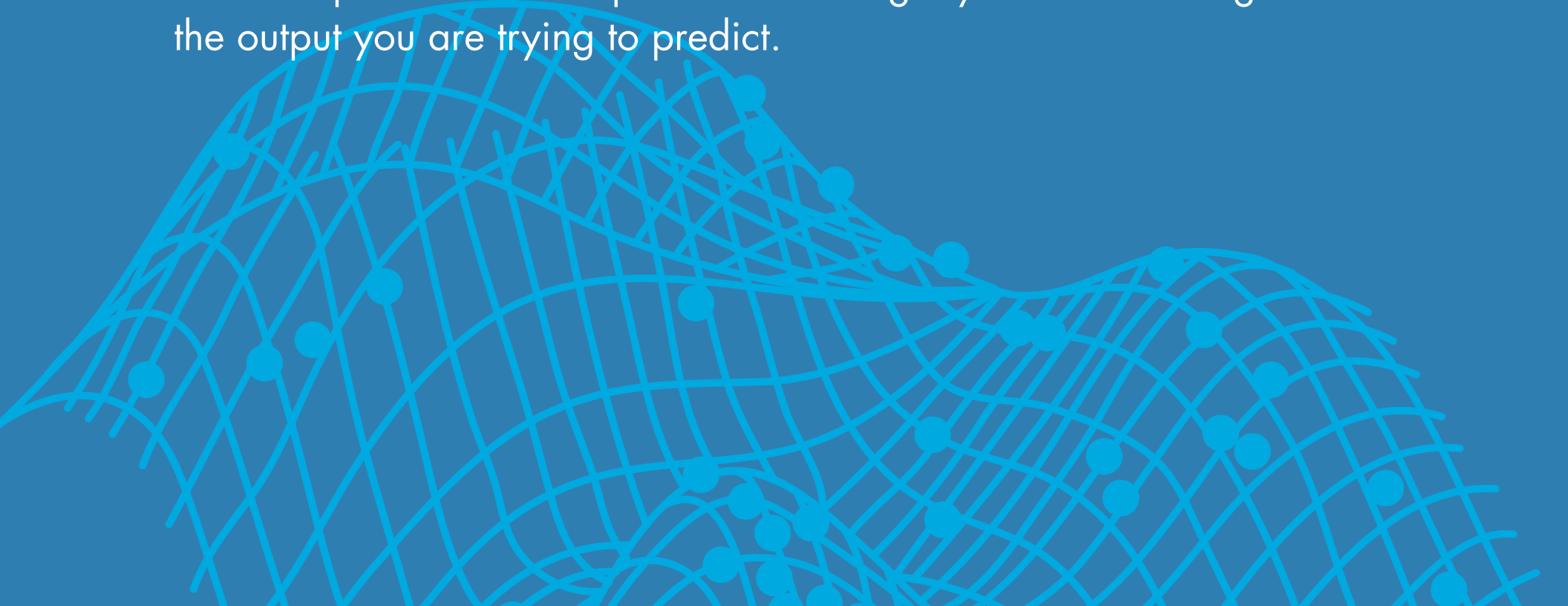
```
%% Ensemble Learning: TreeBagger  
opts = statset('UseParallel',true);
```

```
tb = TreeBagger(150,Xtrain,Ytrain,'method','classification',...  
    'Options',opts,'OOBVarImp','on','cost',[0 1; 5 0]);
```



When to Consider Supervised Learning

A supervised learning algorithm takes a known set of input data (the training set) and known responses to the data (output), and trains a model to generate reasonable predictions for the response to new input data. Use supervised learning if you have existing data for the output you are trying to predict.

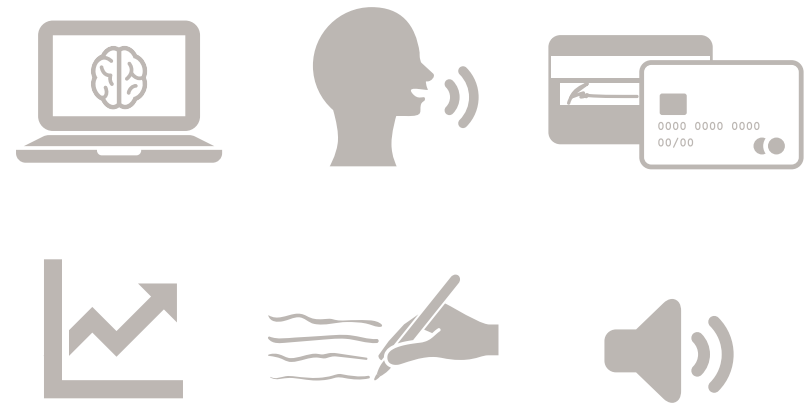
A decorative graphic at the bottom of the slide depicts a neural network. It consists of a grid of blue lines forming a mesh, with several blue circular nodes placed at various points within the grid. The lines and nodes are arranged in a way that suggests a complex, interconnected structure, typical of a neural network architecture.

Supervised Learning Techniques

All supervised learning techniques are a form of classification or regression.

Classification techniques predict discrete responses—for example, whether an email is genuine or spam, or whether a tumor is small, medium, or large. Classification models are trained to classify data into categories. Applications include medical imaging, speech recognition, and credit scoring.

Regression techniques predict continuous responses—for example, changes in temperature or fluctuations in electricity demand. Applications include forecasting stock prices, handwriting recognition, and acoustic signal processing.



- Can your data be tagged or categorized? If your data can be separated into specific groups or classes, use classification algorithms.
- Working with a data range? If the nature of your response is a real number—such as temperature, or the time until failure for a piece of equipment—use regression techniques.

Selecting the Right Algorithm

As we saw in section 1, selecting a machine learning algorithm is a process of trial and error. It's also a trade-off between specific characteristics of the algorithms, such as:

- Speed of training
- Memory usage
- Predictive accuracy on new data
- Transparency or interpretability (how easily you can understand the reasons an algorithm makes its predictions)

Let's take a closer look at the most commonly used classification and regression algorithms.

Using larger training datasets often yield models that generalize well for new data.

Speed of training



Memory usage



Predictive accuracy



Interpretability

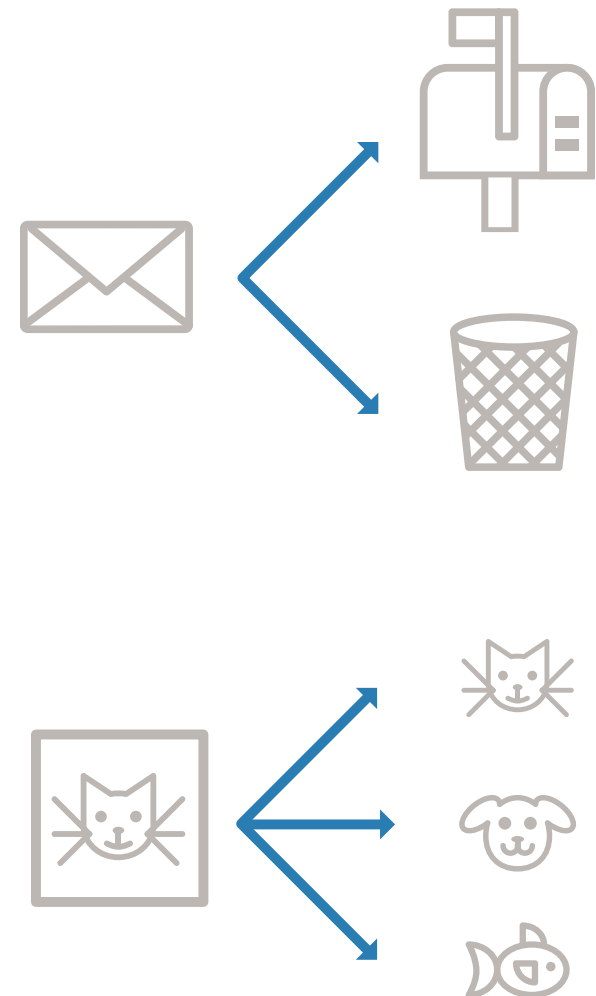


Binary vs. Multiclass Classification

When you are working on a classification problem, begin by determining whether the problem is binary or multiclass. In a binary classification problem, a single training or test item (instance) can only be divided into two classes—for example, if you want to determine whether an email is genuine or spam. In a multiclass classification problem, it can be divided into more than two—for example, if you want to train a model to classify an image as a dog, cat, or other animal.

Bear in mind that a multiclass classification problem is generally more challenging because it requires a more complex model.

Certain algorithms (for example, logistic regression) are designed specifically for binary classification problems. During training, these algorithms tend to be more efficient than multiclass algorithms.



Common Classification Algorithms

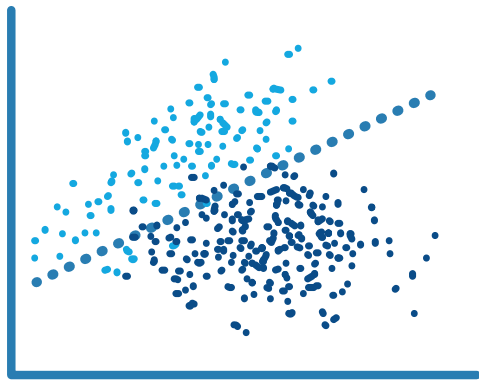
Logistic Regression

How it Works

Fits a model that can predict the probability of a binary response belonging to one class or the other. Because of its simplicity, logistic regression is commonly used as a starting point for binary classification problems.

Best Used...

- When data can be clearly separated by a single, linear boundary
- As a baseline for evaluating more complex classification methods



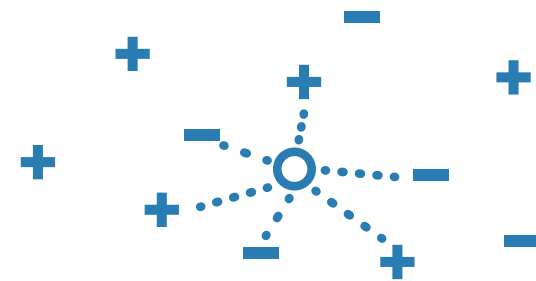
k Nearest Neighbor (kNN)

How it Works

kNN categorizes objects based on the classes of their nearest neighbors in the dataset. kNN predictions assume that objects near each other are similar. Distance metrics, such as Euclidean, city block, cosine, and Chebychev, are used to find the nearest neighbor.

Best Used...

- When you need a simple algorithm to establish benchmark learning rules
- When memory usage of the trained model is a lesser concern
- When prediction speed of the trained model is a lesser concern



Common Classification Algorithms *continued*

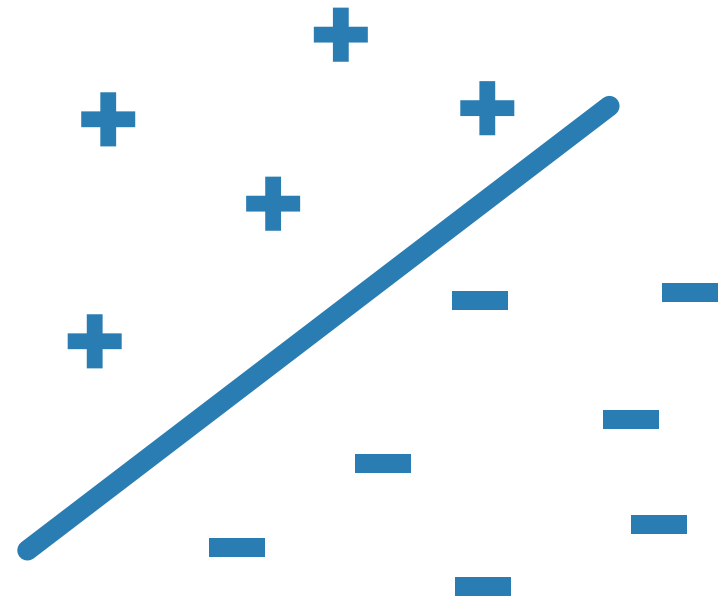
Support Vector Machine (SVM)

How It Works

Classifies data by finding the linear decision boundary (hyperplane) that separates all data points of one class from those of the other class. The best hyperplane for an SVM is the one with the largest margin between the two classes, when the data is linearly separable. If the data is not linearly separable, a loss function is used to penalize points on the wrong side of the hyperplane. SVMs sometimes use a kernel transform to transform nonlinearly separable data into higher dimensions where a linear decision boundary can be found.

Best Used...

- For data that has exactly two classes (you can also use it for multiclass classification with a technique called error-correcting output codes)
- For high-dimensional, nonlinearly separable data
- When you need a classifier that's simple, easy to interpret, and accurate



Common Classification Algorithms *continued*

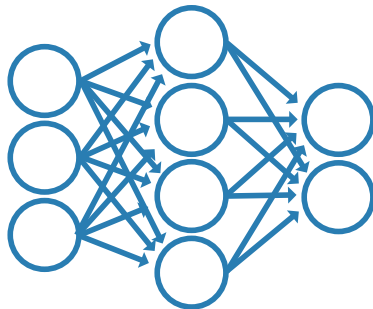
Neural Network

How it Works

Inspired by the human brain, a neural network consists of highly connected networks of neurons that relate the inputs to the desired outputs. The network is trained by iteratively modifying the strengths of the connections so that given inputs map to the correct response.

Best Used...

- For modeling highly nonlinear systems
- When data is available incrementally and you wish to constantly update the model
- When there could be unexpected changes in your input data
- When model interpretability is not a key concern



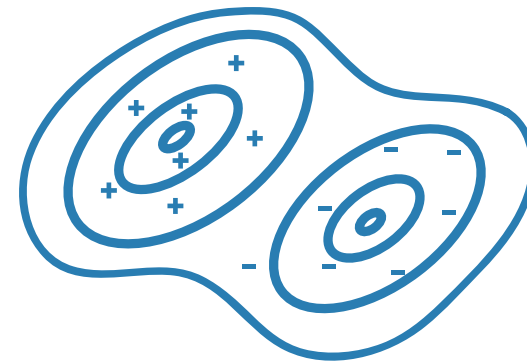
Naïve Bayes

How It Works

A naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. It classifies new data based on the highest probability of its belonging to a particular class.

Best Used...

- For a small dataset containing many parameters
- When you need a classifier that's easy to interpret
- When the model will encounter scenarios that weren't in the training data, as is the case with many financial and medical applications



Common Classification Algorithms *continued*

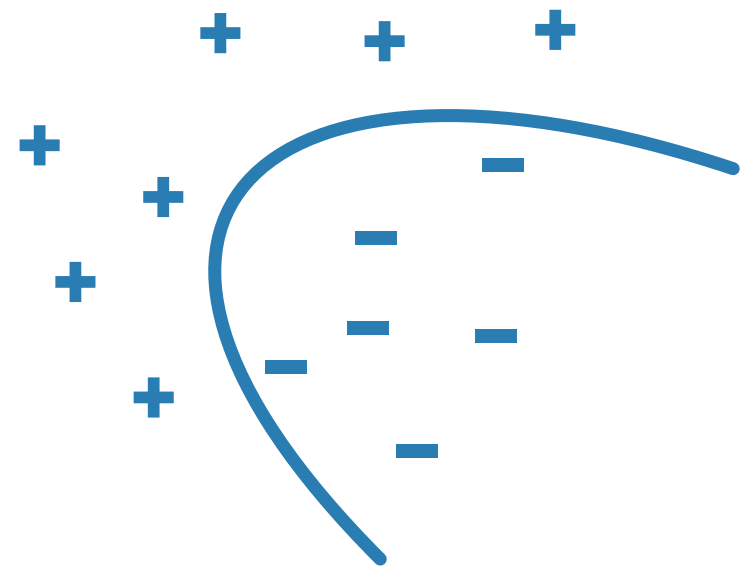
Discriminant Analysis

How It Works

Discriminant analysis classifies data by finding linear combinations of features. Discriminant analysis assumes that different classes generate data based on Gaussian distributions. Training a discriminant analysis model involves finding the parameters for a Gaussian distribution for each class. The distribution parameters are used to calculate boundaries, which can be linear or quadratic functions. These boundaries are used to determine the class of new data.

Best Used...

- When you need a simple model that is easy to interpret
- When memory usage during training is a concern
- When you need a model that is fast to predict



Common Classification Algorithms *continued*

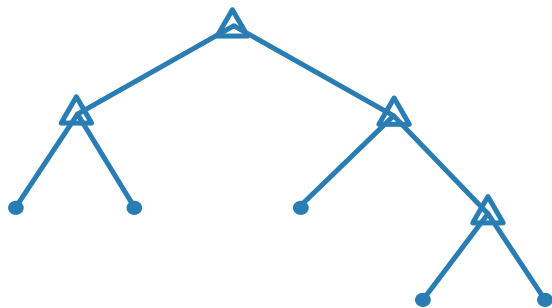
Decision Tree

How it Works

A decision tree lets you predict responses to data by following the decisions in the tree from the root (beginning) down to a leaf node. A tree consists of branching conditions where the value of a predictor is compared to a trained weight. The number of branches and the values of weights are determined in the training process. Additional modification, or pruning, may be used to simplify the model.

Best Used...

- When you need an algorithm that is easy to interpret and fast to fit
- To minimize memory usage
- When high predictive accuracy is not a requirement



Bagged and Boosted Decision Trees

How They Work

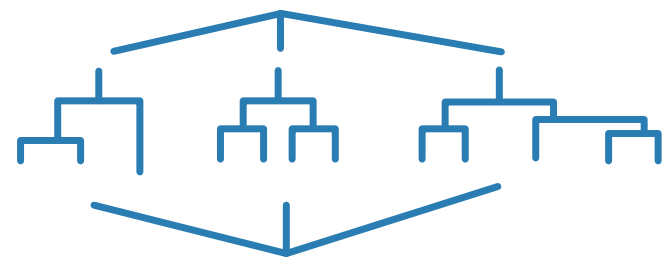
In these ensemble methods, several “weaker” decision trees are combined into a “stronger” ensemble.

A bagged decision tree consists of trees that are trained independently on data that is bootstrapped from the input data.

Boosting involves creating a strong learner by iteratively adding “weak” learners and adjusting the weight of each weak learner to focus on misclassified examples.

Best Used...

- When predictors are categorical (discrete) or behave nonlinearly
- When the time taken to train a model is less of a concern



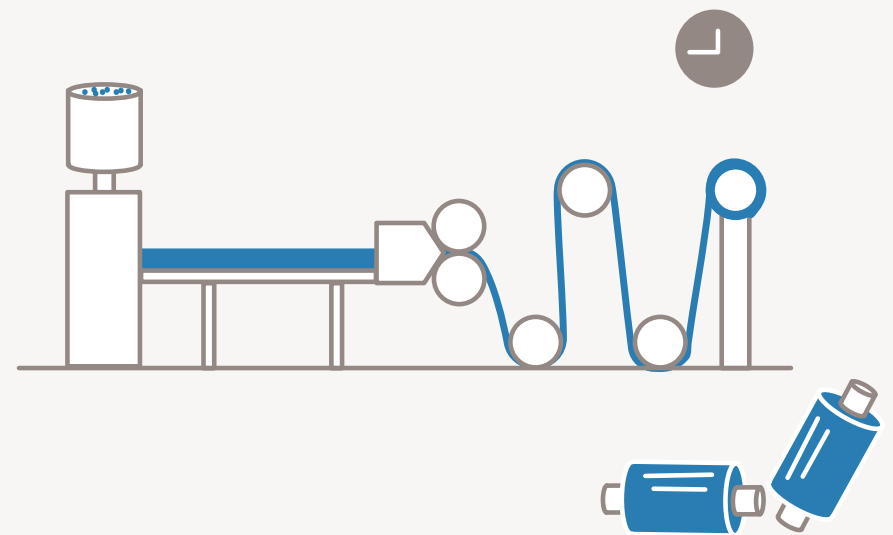
Common Classification Algorithms *continued*

Example: Predictive Maintenance for Manufacturing Equipment

A plastic production plant delivers about 18 million tons of plastic and thin film products annually. The plant's 900 workers operate 24 hours a day, 365 days a year.

To minimize machine failures and maximize plant efficiency, engineers develop a health monitoring and predictive maintenance application that uses advanced statistics and machine learning algorithms to identify potential issues with the machines so that operators can take corrective action and prevent serious problems from occurring.

After collecting, cleaning, and logging data from all the machines in the plant, the engineers evaluate several machine learning techniques, including neural networks, k-nearest neighbors, bagged decision trees, and support vector machines (SVMs). For each technique, they train a classification model using the logged machine data and then test the model's ability to predict machine problems. The tests show that an ensemble of bagged decision trees is the most accurate model for predicting the production quality.



Common Regression Algorithms

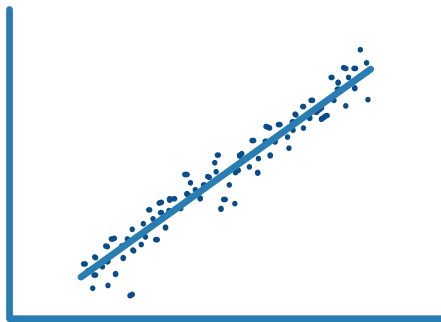
Linear Regression

How it Works

Linear regression is a statistical modeling technique used to describe a continuous response variable as a linear function of one or more predictor variables. Because linear regression models are simple to interpret and easy to train, they are often the first model to be fitted to a new dataset.

Best Used...

- When you need an algorithm that is easy to interpret and fast to fit
- As a baseline for evaluating other, more complex, regression models



Nonlinear Regression

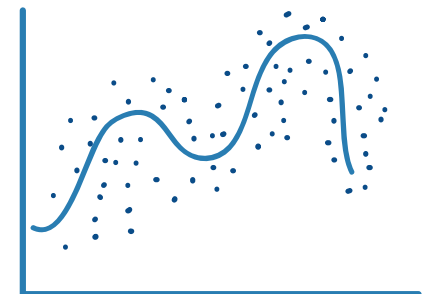
How It Works

Nonlinear regression is a statistical modeling technique that helps describe nonlinear relationships in experimental data. Nonlinear regression models are generally assumed to be parametric, where the model is described as a nonlinear equation.

“Nonlinear” refers to a fit function that is a nonlinear function of the parameters. For example, if the fitting parameters are b_0 , b_1 , and b_2 : the equation $y = b_0 + b_1x + b_2x^2$ is a linear function of the fitting parameters, whereas $y = (b_0x^{b_1}) / (x + b_2)$ is a nonlinear function of the fitting parameters.

Best Used...

- When data has strong nonlinear trends and cannot be easily transformed into a linear space
- For fitting custom models to data



Common Regression Algorithms *continued*

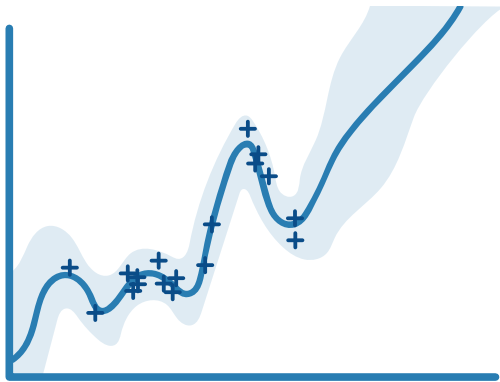
Gaussian Process Regression Model

How it Works

Gaussian process regression (GPR) models are nonparametric models that are used for predicting the value of a continuous response variable. They are widely used in the field of spatial analysis for interpolation in the presence of uncertainty. GPR is also referred to as Kriging.

Best Used...

- For interpolating spatial data, such as hydrogeological data for the distribution of ground water
- As a surrogate model to facilitate optimization of complex designs such as automotive engines



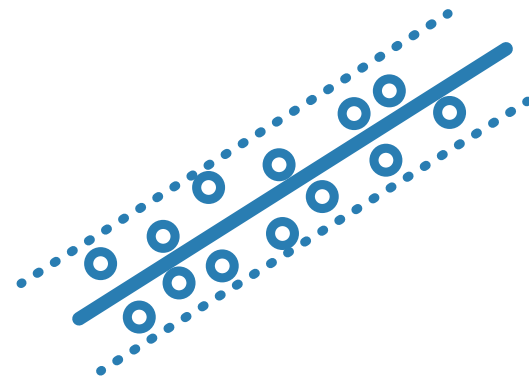
SVM Regression

How It Works

SVM regression algorithms work like SVM classification algorithms, but are modified to be able to predict a continuous response. Instead of finding a hyperplane that separates data, SVM regression algorithms find a model that deviates from the measured data by a value no greater than a small amount, with parameter values that are as small as possible (to minimize sensitivity to error).

Best Used...

- For high-dimensional data (where there will be a large number of predictor variables)



Common Regression Algorithms *continued*

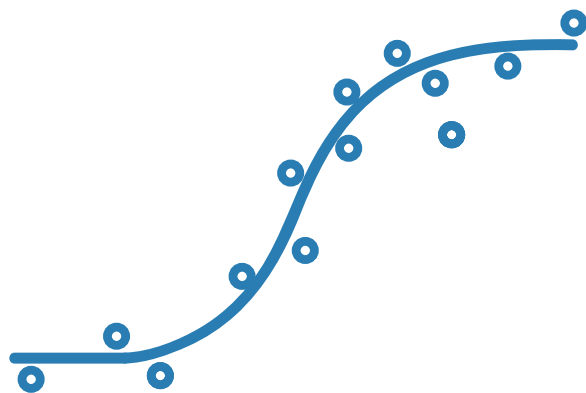
Generalized Linear Model

How it Works

A generalized linear model is a special case of nonlinear models that uses linear methods. It involves fitting a linear combination of the inputs to a nonlinear function (the link function) of the outputs.

Best Used...

- When the response variables have nonnormal distributions, such as a response variable that is always expected to be positive



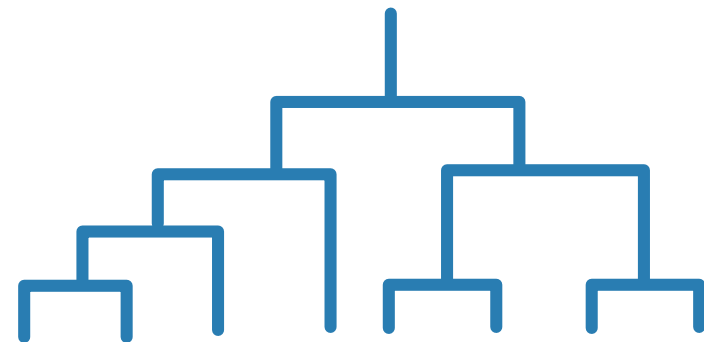
Regression Tree

How It Works

Decision trees for regression are similar to decision trees for classification, but they are modified to be able to predict continuous responses.

Best Used...

- When predictors are categorical (discrete) or behave nonlinearly

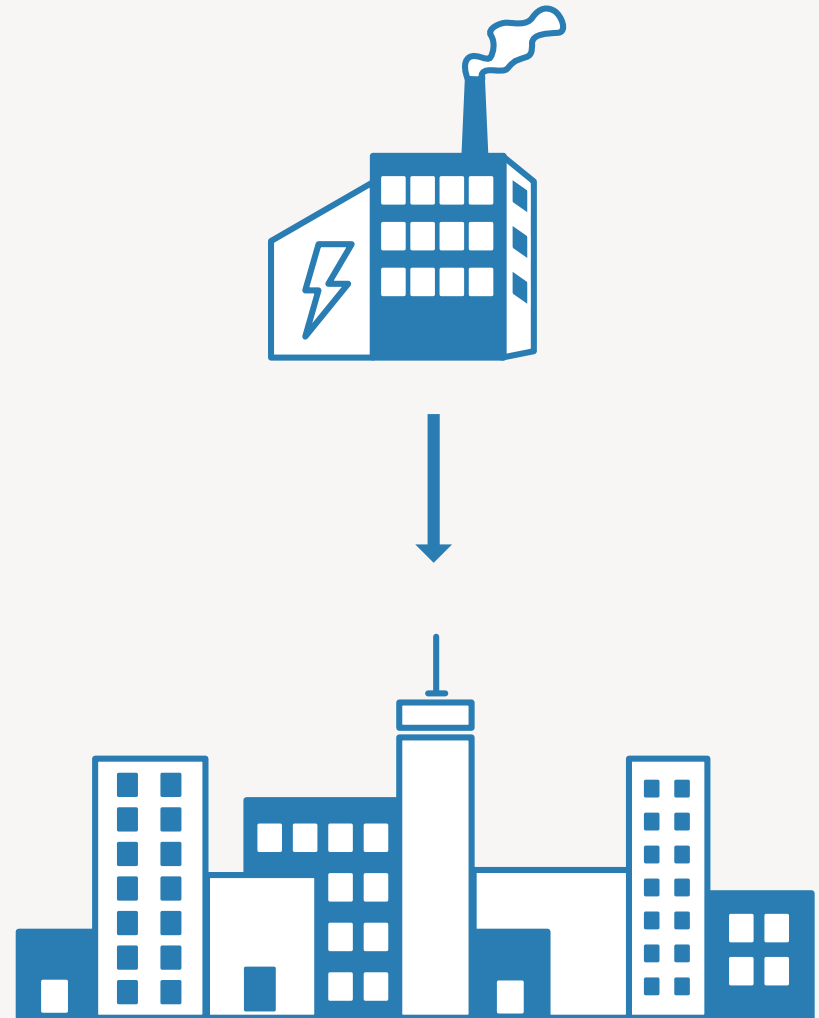


Common Regression Algorithms *continued*

Example: Forecasting Energy Load

Utility analysts at a large gas and electricity company developed models that predict energy demand for the following day. The models enable power grid operators to optimize resources and schedule power plant generation. Each model accesses a central database for historical power consumption and price data, weather forecasts, and parameters for each power plant, including maximum power out, efficiency, costs, and all the operation constraints that influence the plant dispatch.

Analysts looked for a model that provided a low mean absolute percent error (MAPE) to the testing data set. After trying several different types of regression models, it was determined that neural networks provided the lowest MAPE due to their ability to capture the nonlinear behavior of the system.



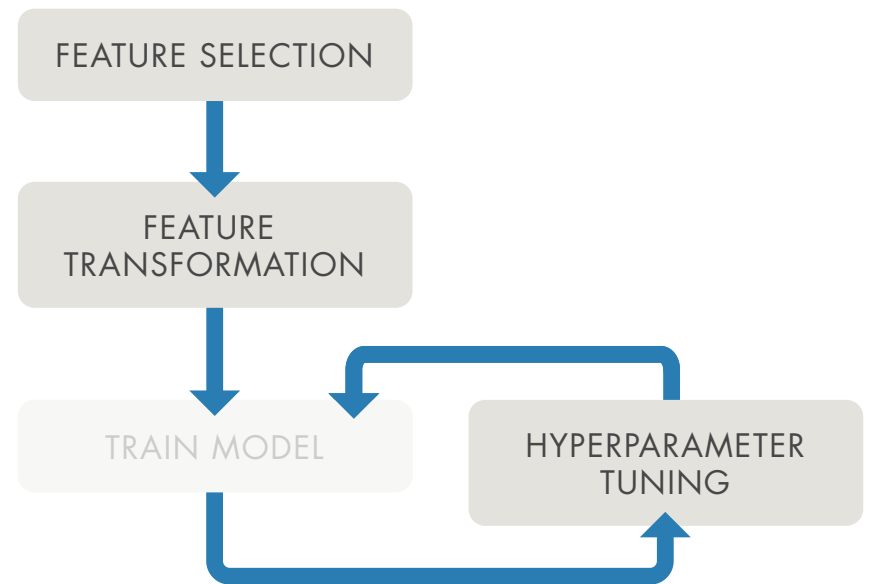
Improving Models

Improving a model means increasing its accuracy and predictive power and preventing overfitting (when the model cannot distinguish between data and noise). Model improvement involves feature engineering (feature selection and transformation) and hyperparameter tuning.

Feature selection: Identifying the most relevant features, or variables, that provide the best predictive power in modeling your data. This could mean adding variables to the model or removing variables that do not improve model performance.

Feature transformation: Turning existing features into new features using techniques such as principal component analysis, nonnegative matrix factorization, and factor analysis.

Hyperparameter tuning: The process of identifying the set of parameters that provides the best model. Hyperparameters control how a machine learning algorithm fits the model to the data.



Feature Selection

Feature selection is one of the most important tasks in machine learning. It's especially useful when you're dealing with high-dimensional data or when your dataset contains a large number of features and a limited number of observations. Reducing features also saves storage and computation time and makes your results easier to understand.

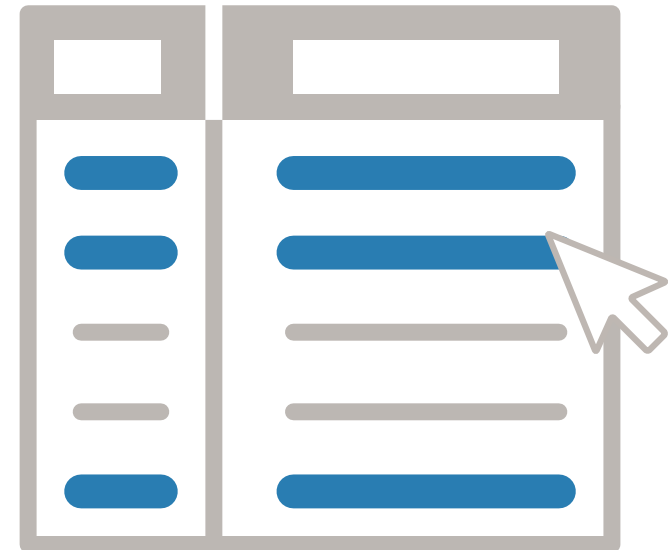
Common feature selection techniques include:

Stepwise regression: Sequentially adding or removing features until there is no improvement in prediction accuracy.

Sequential feature selection: Iteratively adding or removing predictor variables and evaluating the effect of each change on the performance of the model.

Regularization: Using shrinkage estimators to remove redundant features by reducing their weights (coefficients) to zero.

Neighborhood component analysis (NCA): Finding the weight each feature has in predicting the output, so that features with lower weights can be discarded.



A model is only as good as the features you select to train it.

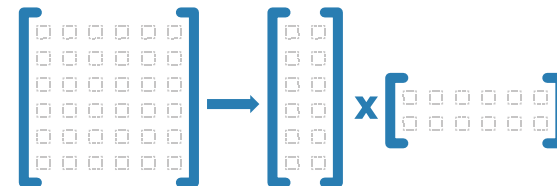
Feature Transformation

Feature transformation is a form of dimensionality reduction. As we saw in section 3, the three most commonly used dimensionality reduction techniques are:

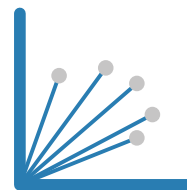
Principal component analysis (PCA): Performs a linear transformation on the data so that most of the variance or information in your high-dimensional dataset is captured by the first few principal components. The first principal component will capture the most variance, followed by the second principal component, and so on.



Nonnegative matrix factorization: Used when model terms must represent nonnegative quantities, such as physical quantities.



Factor analysis: Identifies underlying correlations between variables in your dataset to provide a representation in terms of a smaller number of unobserved latent factors, or common factors.

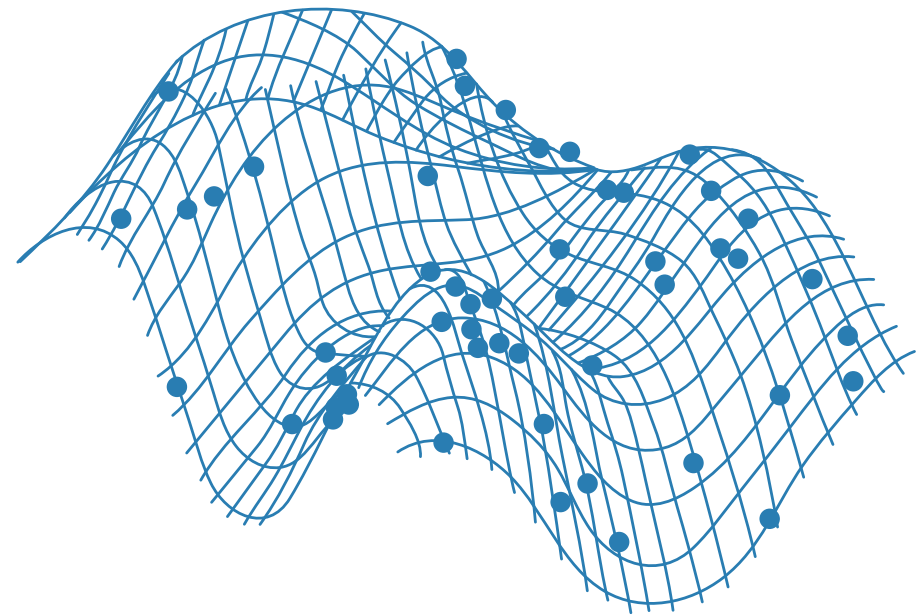


Hyperparameter Tuning

Like many machine learning tasks, parameter tuning is an iterative process. You begin by setting parameters based on a “best guess” of the outcome. Your goal is to find the “best possible” values—those that yield the best model. As you adjust parameters and model performance begins to improve, you see which parameter settings are effective and which still require tuning.

Three common parameter tuning methods are:

- Bayesian optimization
- Grid search
- Gradient-based optimization



A simple algorithm with well-tuned parameters often produces a better model than an inadequately tuned complex algorithm.

Learn More

Ready for a deeper dive? Explore these machine learning methods, examples, and tools.

[Getting Started with Supervised Learning](#)

Classification

[Machine Learning with MATLAB:
Getting Started with Classification](#)

[Introductory Classification Examples](#)

[Bayesian Brain Teaser](#)

[Explore Decision Trees Interactively](#)

[Support Vector Machines](#)

[K-Nearest Neighbor Classification](#)

[Train a Classification Ensemble](#)

[Predicting Tumor Class from Gene Expression
Data Using Bagged Decision Trees](#)

Regression

[Linear Regression](#)

[What are Generalized Linear Models?](#)

[Regression Trees](#)

[Train a Regression Ensemble to Predict the
Fuel Economy of a Car](#)

Feature Selection

[Selecting Features for Classifying High-Dimensional Data](#)