Lecture 2: ML Fundamental Concepts

Foundations of Machine Learning

Introduction

- Data plays a big part in machine learning. It is important to understand and use the right terminology when talking about data.
- In this session you will learn exactly how to describe and talk about data in machine learning.
  - Standard data terminology used in general when talking about spreadsheets of data.
  - Data terminology used in statistics and the statistical view of machine learning.
  - Data terminology used in the computer science perspective of machine learning.
- This will greatly help you with understanding machine learning algorithms in general.

Data As you KnowIt

- How do you think about data? Think of a spreadsheet. You have columns, rows, and cells.

	A	B	C	D
1		Column 1	Column 2	Column 3
2	Row 1	2.2	2.3	1
3	Row 2	2.3	2.6	0
4	Row 3	2.1	2	1
5				

- **Column**: A column describes data of a single type. For example, you could have a column of weights or heights or prices. All the data in one column will have the same scale and have meaning relative to each other.
- **Row:** A row describes a single entity or observation and the columns describe properties about that entity or observation. The more rows you have, the more examples from the problem domain that you have.

Data As you KnowIt

- Cell: A cell is a single value in a row and column. It may be a real value (1.5) an integer (2) or a category (red).
- This is how you probably think about data, columns, rows and cells.
- Generally, we can call this type of data: tabular data. This form of data is easy to work with in machine learning.
- There are different flavors of machine learning that give different perspectives on the field.
- For example there is the statistical perspective and the computer science perspective.
- Next we will look at the different terms used to refer to data as you know it.

Statistical Learning Perspective

- The statistical perspective frames data in the context of a hypothetical function (f) that the machine learning algorithm is trying to learn.
- That is, given some input variables (input), what is the predicted output variable (output).

$$Output = f(Input)$$

- Those columns that are the inputs are referred to as input variables.
- Whereas the column of data that you may not always have and that you would like to predict for new input data in the future is called the output variable.
- It is also called the response variable. $OutputVariable = f(InputVariables)$

	A	B	C
1	X1	X2	Y
2	2.2	2.3	1
3	2.3	2.6	0
4	2.1	2	1
5			

- Typically, you have more than one input variable. In this case the group of input variables are referred to as the input vector.

$$OutputVariable = f(InputVector)$$

- If you have done a little statistics in your past you may know of another more traditional terminology. For example, a statistics text may talk about the input variables as independent variables and the output variable as the dependent variable.
- This is because in the framing of the prediction problem the output is dependent (a function of) the input (also called the independent variables).

$$DependentVariable = f(IndependentVariables)$$

Statistical Learning Perspective

- The data is described using a short hand in equations and descriptions of machine learning algorithms.
- The standard shorthand used in the statistical perspective is to refer to the input variables as capital x (X) and the output variables as capital y (Y).

$$Y = f(X)$$

- When you have multiple input variables they may be dereferenced with an integer to indicate their ordering in the input vector, for example X1, X2 and X3 for data in the first three columns.

Computer Science Perspective

- There is a lot of overlap in the computer science terminology for data with the statistical perspective. We will look at the key differences.
- A row often describes an entity (like a person) or an observation about an entity.
- As such, the columns for a row are often referred to as attributes of the observation.
- When modeling a problem and making predictions, we may refer to input attributes and output attributes

$$OutputAttribute = Program(InputAttributes)$$

Computer Science Perspective

	A	B	C	D
1		Attribute 1	Attribute 2	Output Attribute
2	Instance 1	2.2	2.3	1
3	Instance 2	2.3	2.6	0
4	Instance 3	2.1	2	1
5				

- Another name for columns is features, used for the same reason as attribute, where a feature describes some property of the observation.
- This is more common when working with data where features must be extracted from the raw data in order to construct an observation.
- Exampl $Output = Program(InputFeatures)$ like images, audio and video.

Computer Science Perspective

- Another computer science phrasing is that for a row of data or an observation as an instance.
- This is used because a row may be considered a single example or single instance of data observed or generated by the problem domain.

$$Prediction = Program(Instance)$$

Models and Algorithms

- There is one final note of clarification that is important and that is between algorithms and models.
- This can be confusing as both algorithm and model can be used interchangeably.
- A perspective that I like is to think of the model as the specific representation learned from data and the algorithm as the process for learning it.

- For example, a decisio $Model = Algorithm(Data)$ ents are a model and the C5.0 and Least Squares Linear Regression are algorithms to learn those respective models.

Algorithms Learn a Mapping From Input to Output

- How do machine learning algorithms work? There is a common principle that underlies all supervised machine learning algorithms for predictive modeling.
- It is necessary to know how machine learning algorithms actually work by understanding the common principle that underlies all algorithms.
  - The mapping problem that all supervised machine learning algorithms aim to solve.
  - That the subfield of machine learning focused on making predictions is called predictive modeling.
  - That different machine learning algorithms represent different strategies for learning the mapping function.

Learning a Function

- Machine learning algorithms are described as learning a target function (f) that best maps input variables (X) to an output variable (Y).

$$Y = f(X)$$

- This is a general learning task where we would like to make predictions in the future (Y) given new examples of input variables (X).
- We don't know what the function (f) looks like or its form. If we did, we would use it directly and we would not need to learn it from data using machine learning algorithms. It is harder than you think.
- There is also error (e) that is independent of the input data (X).

$$Y = f(X) + e$$

- This error might be error such as not having enough attributes to sufficiently characterize the best mapping from X to Y .
- This error is called irreducible error because no matter how good we get at estimating the target function (f), we cannot reduce this error.
- This is to say, that the problem of learning a function from data is a difficult problem and this is the reason why the field of machine learning and machine learning algorithms exist.

Learning a Function To Make Predictions

- The most common type of machine learning is to learn the mapping Y = f(X) to make predictions of Y for new X. This is called predictive modeling or predictive analytics and our goal is to make the most accurate predictions possible.
- As such, we are not really interested in the shape and form of the function (f) that we are learning, only that it makes accurate predictions. We could learn the mapping of Y = f(X) to learn more about the relationship in the data and this is called statistical inference.
- If this were the goal, we would use simpler methods and value understanding the learned model and form of (f) above making accurate predictions.

Learning a Function To Make Predictions

- When we learn a function (f) we are estimating its form from the data that we have available. As such, this estimate will have error.
- It will not be a perfect estimate for the underlying hypothetical best mapping from Y given X.
- Much time in applied machine learning is spent attempting to improve the estimate of the underlying function and in turn improve the performance of the predictions made by the model.

Techniques For Learning a Function

- Machine learning algorithms are techniques for estimating the target function (f) to predict the output variable (Y) given input variables (X).
- Different representations make different assumptions about the form of the function being learned, such as whether it is linear or nonlinear.

Techniques For Learning a Function

- Different machine learning algorithms make different assumptions about the shape and structure of the function and how best to optimize a representation to approximate it.
- This is why it is so important to try a suite of different algorithms on a machine learning problem, because we cannot know beforehand which approach will be best at estimating the structure of the underlying function we are trying to approximate.

Parametric and Nonparametric Machine Learning Algorithms

- What is a parametric machine learning algorithm and how is it different from a nonparametric machine learning algorithm?
- In this session you will discover the difference between parametric and nonparametric machine learning algorithms. Things to know:
  - That parametric machine learning algorithms simplify the mapping to a known functional form.
  - That nonparametric algorithms can learn any mapping from inputs to outputs.
  - That all algorithms can be organized into parametric or nonparametric groups.

Parametric Machine Learning Algorithms

- Assumptions can greatly simplify the learning process, but can also limit what can be learned.
- Algorithms that simplify the function to a known form are called parametric machine learning algorithms.
  - *A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a parametric model. No matter how much data you throw at a parametric model, it won't change its mind about how many parameters it needs-*
    - **Artificial Intelligence: A Modern Approach, page 737**

Parametric Machine Learning Algorithms

- A parametric algorithm involve two steps:
- 1. Select a form for the function.
- 2. Learn the coefficients for the function from the training data.
- An easy to understand functional form for the mapping function is a line, as is used in linear regression:

$$B0 + B1 \times X1 + B2 \times X2 = 0$$

Parametric Machine Learning Algorithms

- Where B0, B1 and B2 are the coefficients of the line that control the intercept and slope, and X1 and X2 are two input variables.
- Assuming the functional form of a line greatly simplifies the learning process.
- Now, all we need to do is estimate the coefficients of the line equation and we have a predictive model for the problem.

Parametric Machine Learning Algorithms

- Often the assumed functional form is a linear combination of the input variables and as such parametric machine learning algorithms are often also called linear machine learning algorithms.
- The problem is, the actual unknown underlying function may not be a linear function like a line.
- It could be almost a line and require some minor transformation of the input data to work right.
- Or it could be nothing like a line in which case the assumption is wrong and the approach will produce poor results.

Parametric Machine Learning Algorithms

- Some more examples of parametric machine learning algorithms include:
  - Logistic Regression
  - Linear Discriminant Analysis
  - Perceptron
- Benefits of Parametric Machine Learning Algorithms:
  - Simpler: These methods are easier to understand and interpret results.
  - Speed: Parametric models are very fast to learn from data.
  - Less Data: They do not require as much training data and can work well even if the fit to the data is not perfect.
- Limitations of Parametric Machine Learning Algorithms:
  - Constrained: By choosing a functional form these methods are highly constrained to the specified form.
  - Limited Complexity: The methods are more suited to simpler problems.
  - Poor Fit: In practice the methods are unlikely to match the underlying mapping function.

Nonparametric Machine Learning Algorithms

- Algorithms that do not make strong assumptions about the form of the mapping function are called nonparametric machine learning algorithms.
- By not making assumptions, they are free to learn any functional form from the training data.
  - *Nonparametric methods are good when you have a lot of data and no prior knowledge, and when you don't want to worry too much about choosing just the right features.*
    - *Artificial Intelligence: A Modern Approach, page 757*

Nonparametric Machine Learning Algorithms

- Nonparametric methods seek to best fit the training data in constructing the mapping function, whilst maintaining some ability to generalize to unseen data.
- As such, they are able to fit a large number of functional forms. An easy to understand nonparametric model is the k-nearest neighbors algorithm that makes predictions based on the k most similar training patterns for a new data instance.
- The method does not assume anything about the form of the mapping function other than patterns that are close are likely have a similar output variable.

Nonparametric Machine Learning Algorithms

- Some more examples of popular nonparametric machine learning algorithms are:
  - Decision Trees like CART and C4.5
  - Naive Bayes
  - Support Vector Machines
  - Neural Networks
- Benefits of Nonparametric Machine Learning Algorithms:
  - Flexibility: Capable of fitting a large number of functional forms.
  - Power: No assumptions (or weak assumptions) about the underlying function.
  - Performance: Can result in higher performance models for prediction.
- Limitations of Nonparametric Machine Learning Algorithms:
  - More data: Require a lot more training data to estimate the mapping function.
  - Slower: A lot slower to train as they often have far more parameters to train.
  - Overfitting: More of a risk to overt the training data and it is harder to explain why specific predictions are made.

Supervised, Unsupervised and Semi-Supervised Learning

- What is supervised machine learning and how does it relate to unsupervised machine learning?
- In this session you will discover supervised learning, unsupervised learning and semi-supervised learning. You should know:
  - About the classification and regression supervised learning problems.
  - About the clustering and association unsupervised learning problems.
  - Example algorithms used for supervised and unsupervised problems.

A problem that sits in between supervised and unsupervised learning called semi-supervised learning.

Supervised Machine Learning

- The majority of practical machine learning uses supervised learning.
- Supervised learning is where you have input variables (X) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

$$Y = f(X)$$

Supervised Machine Learning

- The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (Y) for that data.
- It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process.
- We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher.
- Learning stops when the algorithm achieves an acceptable level of performance.

Supervised Machine Learning

- Supervised learning problems can be further grouped into regression and classification problems.
  - Classification: A classification problem is when the output variable is a category, such as red or blue or disease and no disease.
  - Regression: A regression problem is when the output variable is a real value, such as dollars or weight.
  - Some common types of problems built on top of classification and regression include recommendation and time series prediction respectively.
- Some popular examples of supervised machine learning algorithms are:
  - Linear regression for regression problems.
  - Random forest for classification and regression problems.
  - Support vector machines for classification problems.

Unsupervised Machine Learning

- Unsupervised learning is where you only have input data (X) and no corresponding output variables.
- The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.
- This is called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher.
- Algorithms are left to their own devises to discover and present the interesting structure in the data.

Unsupervised Machine Learning

- Unsupervised learning problems can be further grouped into clustering and association problems.
  - Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
  - Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy A also tend to buy B.
- Some popular examples of unsupervised learning algorithms are:
  - k-means for clustering problems.
  - Apriori algorithm for association rule learning problems.

Semi-Supervised Machine Learning

- Problems where you have a large amount of input data (X) and only some of the data is labeled (Y) are called semi-supervised learning problems.
- These problems sit in between both supervised and unsupervised learning.
- A good example is a photo archive where only some of the images are labeled, (e.g. dog, cat, person) and the majority are unlabeled.
- Many real world machine learning problems fall into this area.
- This is because it can be expensive or time consuming to label data as it may require access to domain experts. Whereas unlabeled data is cheap and easy to collect and store.

Semi-Supervised Machine Learning

- You can use unsupervised learning techniques to discover and learn the structure in the input variables.
- You can also use unsupervised learning techniques to make best guess predictions for the unlabeled data, feed that data back into the supervised learning algorithm as training data and use the model to make predictions on new unseen data.

You now know that:

- Supervised: All data is labeled and the algorithms learn to predict the output from the input data.
- Unsupervised: All data is unlabeled and the algorithms learn to inherent structure from the input data.
- Semi-supervised: Some data is labeled but most of it is unlabeled and a mixture of supervised and unsupervised techniques can be used.
- In the next session you will discover the two biggest sources of error when learning from data, namely bias and variance and the tension between these two concerns.

The Bias-Variance Trade-Off

- Supervised machine learning algorithms can best be understood through the lens of the bias-variance trade-off.
- In this session you will discover the Bias-Variance Trade-Off and how to use it to better understand machine learning algorithms and get better performance on your data.
- Things you need to know.
  - That all learning error can be broken down into bias or variance error.
  - That bias refers to the simplifying assumptions made by the algorithm to make the problem easier to solve.
  - That variance refers to the sensitivity of a model to changes to the training data.
  - That all of applied machine learning for predictive model is best understood through the framework of bias and variance.

Overview of Bias and Variance

- In supervised machine learning an algorithm learns a model from training data.
- The goal of any supervised machine learning algorithm is to best estimate the mapping function (f) for the output variable (Y) given the input data (X).
- The mapping function is often called the target function because it is the function that a given supervised machine learning algorithm aims to approximate.

Overview of Bias and Variance

- The prediction error for any machine learning algorithm can be broken down into three parts:
  - Bias Error
  - Variance Error
  - Irreducible Error

Overview of Bias and Variance

- The irreducible error cannot be reduced regardless of what algorithm is used.
- It is the error introduced from the chosen framing of the problem and may be caused by factors like unknown variables that influence the mapping of the input variables to the output variable.
- In this session we will focus on the two parts we can influence with our machine learning algorithms.
- The bias error and the variance error.

Bias Error

- Bias are the simplifying assumptions made by a model to make the target function easier to learn.
- Generally parametric algorithms have a high bias making them fast to learn and easier to understand but generally less flexible.
- In turn they have lower predictive performance on complex problems that fail to meet the simplifying assumptions of the algorithms bias.
  - Low Bias: Suggests less assumptions about the form of the target function.
  - High-Bias: Suggests more assumptions about the form of the target function.
- Examples of low-bias machine learning algorithms include: Decision Trees, k-Nearest Neighbors and Support Vector Machines.
- Examples of high-bias machine learning algorithms include: Linear Regression, Linear Discriminant Analysis and Logistic Regression.

Variance Error

- Variance is the amount that the estimate of the target function will change if different training data was used.
- The target function is estimated from the training data by a machine learning algorithm, so we should expect the algorithm to have some variance.
- Ideally, it should not change too much from one training dataset to the next, meaning that the algorithm is good at picking out the hidden underlying mapping between the inputs and the output variables.

Variance Error

- Machine learning algorithms that have a high variance are strongly influenced by the specifics of the training data.
- This means that the specifics of the training data influences the number and types of parameters used to characterize the mapping function.
  - Low Variance: Suggests small changes to the estimate of the target function with changes to the training dataset.
  - High Variance: Suggests large changes to the estimate of the target function with changes to the training dataset.

Variance Error

- Generally nonparametric machine learning algorithms that have a lot of flexibility have a high variance.
- For example decision trees have a high variance, that is even higher if the trees are not pruned before use.
- Examples of low-variance machine learning algorithms include: Linear Regression, Linear Discriminant Analysis and Logistic Regression.
- Examples of high-variance machine learning algorithms include: Decision Trees, k-Nearest Neighbors and Support Vector Machines.

Bias-Variance Trade-Off

- The goal of any supervised machine learning algorithm is to achieve low bias and low variance.
- In turn the algorithm should achieve good prediction performance. You can see a general trend in the examples above:
  - Parametric or linear machine learning algorithms often have a high bias but a low variance.
  - Nonparametric or nonlinear machine learning algorithms often have a low bias but a high variance.

Bias-Variance Trade-Off

- The parameterization of machine learning algorithms is often a battle to balance out bias and variance.
- Below are two examples of configuring the bias-variance trade-off for specific algorithms:
  - The k-nearest neighbors algorithm has low bias and high variance, but the trade-off can be changed by increasing the value of k which increases the number of neighbors that contribute to the prediction and in turn increases the bias of the model.
  - The support vector machine algorithm has low bias and high variance, but the trade-off can be changed by increasing the C parameter that influences the number of violations of the margin allowed in the training data which increases the bias but decreases the variance.

Bias-Variance Trade-Off

- There is no escaping the relationship between bias and variance in machine learning.
  - Increasing the bias will decrease the variance.
  - Increasing the variance will decrease the bias.
- There is a trade-off at play between these two concerns and the algorithms you choose and the way you choose to configure them are finding different balances in this trade-off for your problem.
- In reality we cannot calculate the real bias and variance error terms because we do not know the actual underlying target function.
- Nevertheless, as a framework, bias and variance provide the tools to understand the behavior of machine learning algorithms in the pursuit of predictive performance.

You now know that:

- Bias is the simplifying assumptions made by the model to make the target function easier to approximate.
- Variance is the amount that the estimate of the target function will change given different training data.
- Trade-off is tension between the error introduced by the bias and the variance.

- You now know about bias and variance, the two sources of error when learning from data.
- In the next session you will learn the practical implications of bias and variance when applying machine learning to problems, namely overfitting and underfitting.

Overfitting and Underfitting

- The cause of poor performance in machine learning is either overfitting or underfitting the data.
- In this session you will learn the concept of generalization in machine learning and the problems of overfitting and underfitting that go along with it.
- Things to know:
  - That overfitting refers to learning the training data too well at the expense of not generalizing well to new data.
  - That underfitting refers to failing to learn the problem from the training data sufficiently.
  - That overfitting is the most common problem in practice and can be addressed by using resampling methods and a held-back verification dataset.

Generalization in Machine Learning

- In machine learning we describe the learning of the target function from training data as inductive learning.
- Induction refers to learning general concepts from specific examples which is exactly the problem that supervised machine learning problems aim to solve.
- This is different from deduction that is the other way around and seeks to learn specific concepts from general rules.

Generalization in Machine Learning

- Generalization refers to how well the concepts learned by a machine learning model apply to specific examples not seen by the model when it was learning.
- The goal of a good machine learning model is to generalize well from the training data to any data from the problem domain.
- This allows us to make predictions in the future on data the model has never seen.
- There is a terminology used in machine learning when we talk about how well a machine learning model learns and generalizes to new data, namely overfitting and underfitting.
- Overtting and underfitting are the two biggest causes for poor performance of machine learning algorithms.

Statistical Fit

- In statistics a fit refers to how well you approximate a target function.
- This is good terminology to use in machine learning, because supervised machine learning algorithms seek to approximate the unknown underlying mapping function for the output variables given the input variables.

Statistical Fit

- Statistics often describe the goodness of fit which refers to measures used to estimate how well the approximation of the function matches the target function.
- Some of these methods are useful in machine learning (e.g. calculating the residual errors), but some of these techniques assume we know the form of the target function we are approximating, which is not the case in machine learning.
- If we knew the form of the target function, we would use it directly to make predictions, rather than trying to learn an approximation from samples of noisy training data.

Overfitting in Machine Learning

- Overfitting refers to a model that models the training data too well.
- Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance on the model on new data.
- This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model.
- The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

Overfitting in Machine Learning

- Overfitting is more likely with nonparametric and nonlinear models that have more flexibility when learning a target function.
- As such, many nonparametric machine learning algorithms also include parameters or techniques to limit and constrain how much detail the model learns.
- For example, decision trees are a nonparametric machine learning algorithm that is very flexible and is subject to overfitting training data.
- This problem can be addressed by pruning a tree after it has learned in order to remove some of the detail it has picked up.

Underfitting in Machine Learning

- Underfitting refers to a model that can neither model the training data nor generalize to new data.
- An underfit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data.
- Underfitting is often not discussed as it is easy to detect given a good performance metric.
- The remedy is to move on and try alternate machine learning algorithms.
- Nevertheless, it does provide good contrast to the problem of concept of overfitting.

A Good Fit in MachineLearning

- Ideally, you want to select a model at the sweet spot between underfitting and overfitting. This is the goal, but is very difficult to do in practice.
- To understand this goal, we can look at the performance of a machine learning algorithm over time as it is learning a training data.
- We can plot both the skill on the training data and the skill on a test dataset we have held back from the training process.
- Over time, as the algorithm learns, the error for the model on the training data goes down and so does the error on the test dataset.

A Good Fit in MachineLearning

- If we train for too long, the error on the training dataset may continue to decrease because the model is overfitting and learning the irrelevant detail and noise in the training dataset.
- At the same time the error for the test set starts to rise again as the model's ability to generalize decreases.
- The sweet spot is the point just before the error on the test dataset starts to increase where the model has good skill on both the training dataset and the unseen test dataset.

A Good Fit in MachineLearning

- You can perform this experiment with your favorite machine learning algorithms.
- This is often not useful technique in practice, because by choosing the stopping point for training using the skill on the test dataset it means that the testset is no longer unseen or a standalone objective measure.
- Some knowledge (a lot of useful knowledge) about that data has leaked into the training procedure.
- There are two additional techniques you can use to help find the sweet spot in practice: resampling methods and a validation dataset.

How To Limit Overfitting

- Both overfitting and underfitting can lead to poor model performance. But by far the most common problem in applied machine learning is overfitting.
- Overfitting is such a problem because the evaluation of machine learning algorithms on training data is different from the evaluation we actually care the most about, namely how well the algorithm performs on unseen data.
- There are two important techniques that you can use when evaluating machine learning algorithms to limit overfitting:
  - 1. Use a resampling technique to estimate model accuracy.
  - 2. Hold back a validation dataset.

How To Limit Overfitting

- The most popular resampling technique is k-fold cross-validation.
- It allows you to train and test your model k-times on different subsets of training data and build up an estimate of the performance of a machine learning model on unseen data.
- A validation dataset is simply a subset of your training data that you hold back from your machine learning algorithms until the very end of your project.

How To Limit Overfitting

- After you have selected and tuned your machine learning algorithms on your training dataset you can evaluate the learned models on the validation dataset to get a final objective idea of how the models might perform on unseen data.
- Using cross-validation is a gold standard in applied machine learning for estimating model accuracy on unseen data. If you have the data, using a validation dataset is also an excellent practice.

Summary

- In this session you discovered that machine learning is solving problems by the method of induction. You learned that generalization is a description of how well the concepts learned by a model apply to new data.
- Finally you learned about the terminology of generalization in machine learning of overfitting and underfitting:
  - Overfitting: Good performance on the training data, poor generalization to other data.
  - Underfitting: Poor performance on the training data and poor generalization to other data.

Summary

- You now know about the risks of overfitting and underfitting data. This session draws your background on machine learning algorithms to an end.
- In the next part you will start learning about machine learning algorithms, starting with linear algorithms.

The End