**Random variable:** Uncertainty about the outcome

**Probability**: measures the likelihood / frequency of occurrence of a

random variable (Value between [0,1])

**Expected value/mean:** average value of a random variable

**Variance:** measures the deviation from the mean value

Probability distribution

**Gaussian/ Normal**

distribution is most common (Fully characterized by mean and variance)

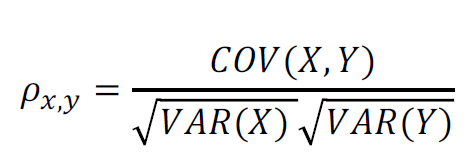
**Conditional probability:**

Given that an event has occurred, what is the probability that another event will occur

**Covariance:**

COV(X,Y) = E[(X-mean(X))(Y-mean(Y)]

**Correlation coefficient:**



**Machine learning**

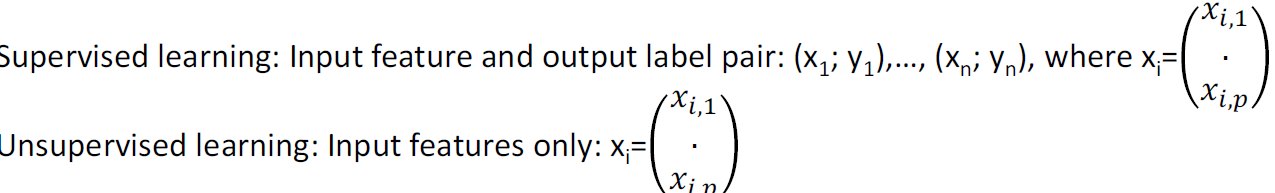
Field of study that gives computers the ability to learn without being explicitly programmed

• Machine learns with respect to a particular task T, performance metric P and experience E, if the performance P on task T improves with experience E.

• **Supervised Learning**: Learn to predict from labeled data (correct answers are given in learning phase)

• **Unsupervised learning**: Find structure in unlabeled data

• Others: Reinforcement learning



**Parametric approach**

First, assume function form

Second, use training to fit the model (ordinary least square)

**Non-parametric approach (KNN)**

No explicit form of function is assumed

Advantage: Could work well for a wider range of possible shapes of f

Disadvantage:

Needs large number of data points

Difficult to understand relationship between output and features

**Liner regression** (easy to interpret)

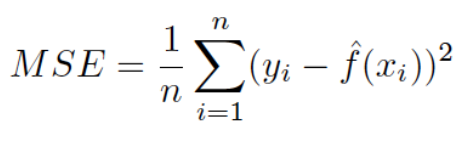
Less flexible models, more restrictive, less complex

**SVM** (hard to interpret)

More flexible, more complex

**Assessing the model accuracy**

In regression setting, a common measure is mean squared error(MSE)



Test:

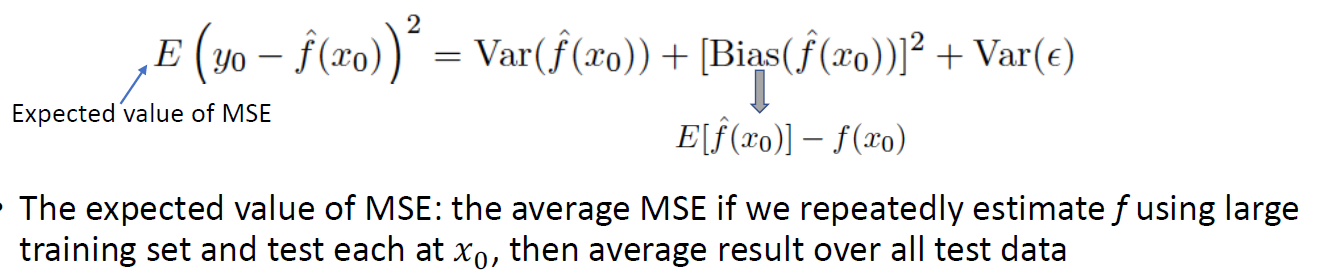


**Overfitting:** Building a model that is too complex, fits training data very

well, but fail to generalize to new data (e.g. large test MSE)

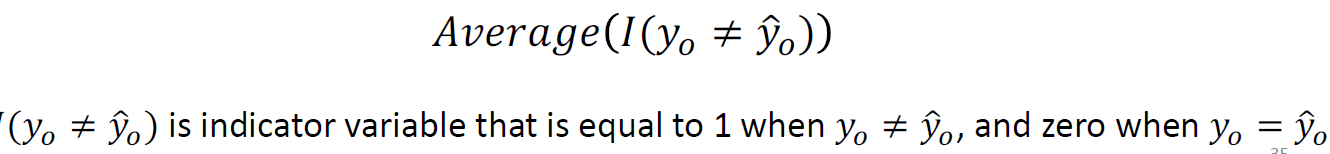
**Underfitting**: build simple model that is unable to capture variability

in data



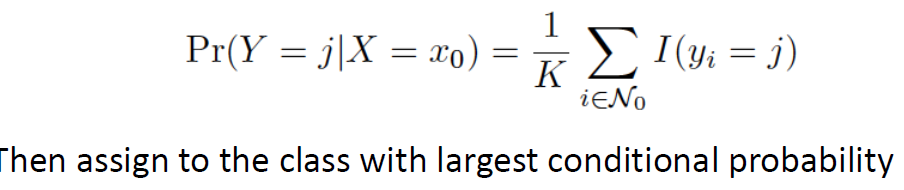
**Variance**: amount by which !" changes if we made the estimation by different training set

**Bias**: Errors from approximating real-life problems by a simpler model



**Bayes classifier** assigns each observation to the most likely class given the feature values. (decision boundary)

Sometime hard to calculate conditional probability (KNN)

Typically, **Euclidean distance** is used to find close neighbors

**Feature scaling** (do in preprocessing step)

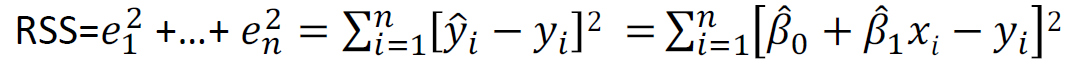
MinMaxScaler (scales features in range of 0-1)

StandardScaler (scales featues so that they are all with zero mean and unit variance)

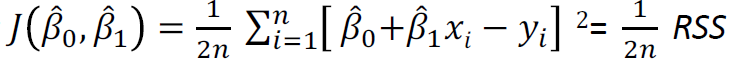
**Linear regression**

One method to estimate coefficients is the **least square method**

RSS(residual sum of squares)



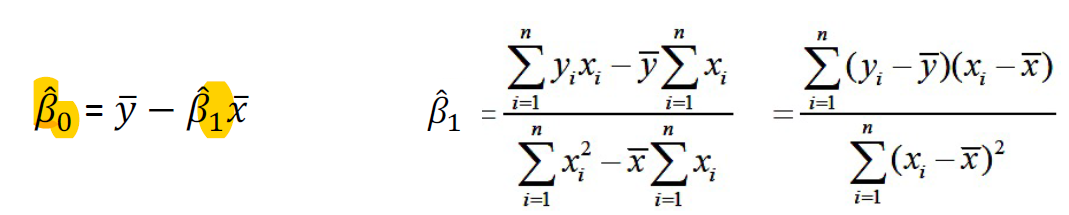
MSE=1/n \* RSS



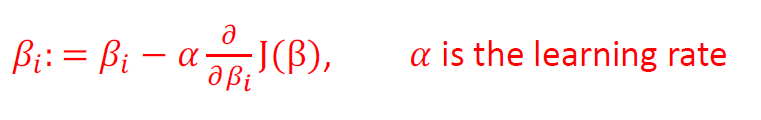
**How to find coefficient?**

1. Get derivative of the cost function then set to zero –> **closed form**

**method**

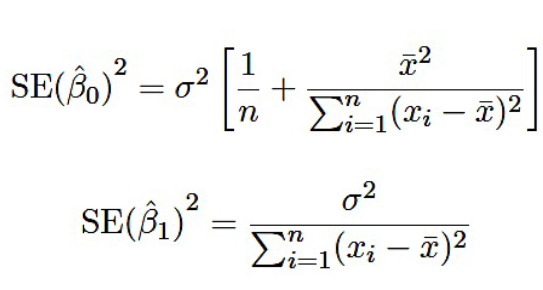


1. gradient descent (Iterative method to find the optimal values of coefficients)

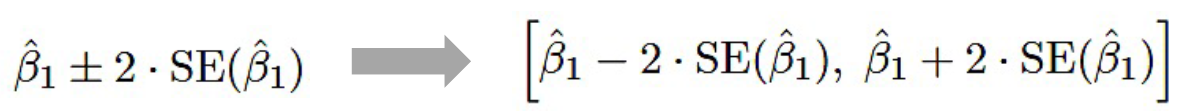


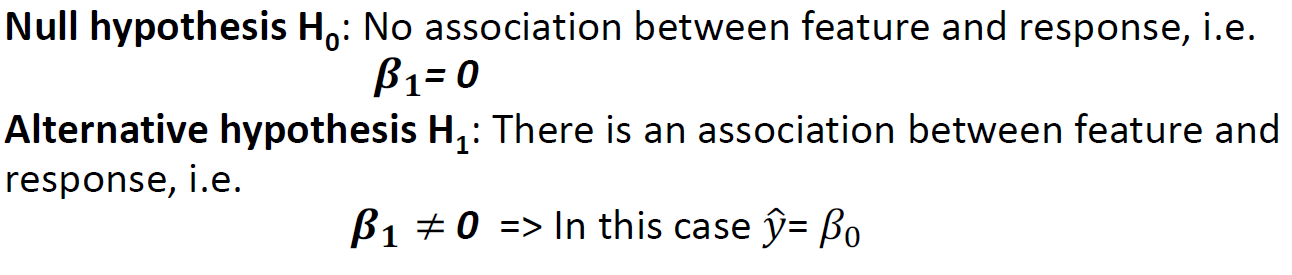
disadvantage: may coverage at a local minima

**Variance** (Reflects how the coefficient varies under repeated sampling)

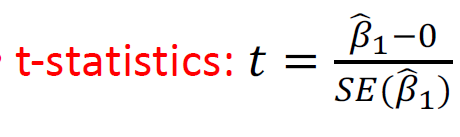


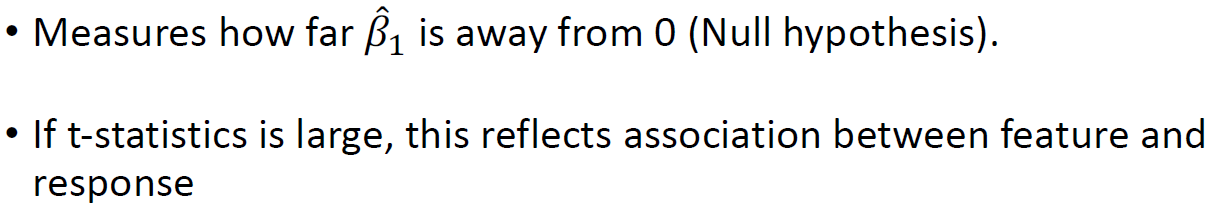
**confidence interval** is the range of values that with 95% probability the range will contain the **true unknown value** of the coefficient.

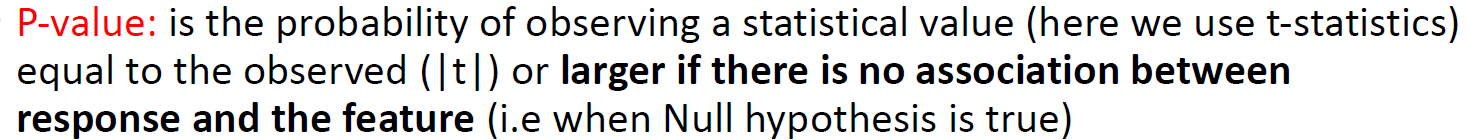




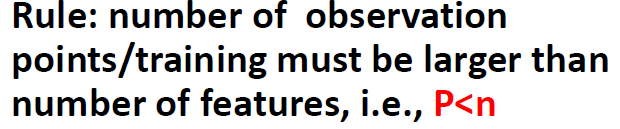
**Test null hypothesis**







**Multiple linear regression**



**Forward selection:**

• Start with null hypothesis

• Fit p simple linear regression models, then add to the null model

the **feature that results in lowest RSS**

• Add to that model the feature that results in lowest RSS among all

two-feature models

• And so on until stopping criteria is met

**Backward selection:**

• Start with all variables/features in the model

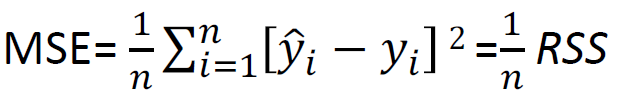
• **Remove variable with largest p-value**

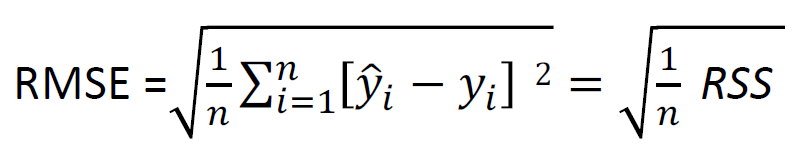
• The new (p-1 features) model is fitted, and feature with largest p-value is removed

• Repeat until stopping criteria is met: all variables have p-value below some threshold

**Mixed selection (may remove after adding it)**

**Feature scaling and normalization (essential when using gradient descent)**





**R2** measures the proportion of variability in Y that can be explained using feature (X)

**Additive assumption**: the change in the response due to one-unit change in feature ! is constant ("#), and is independent of other features

If the interaction term (e.g. X1 X2) is important (has low p-value), then we also include the individual terms (X1 ) and (X2) regardless of their p-value: **hierarchy principle**

**Parametric**

• Make strong assumption about f(x). e.g. linear model

• Easy to fit and understand

**Non-paramteric methods**

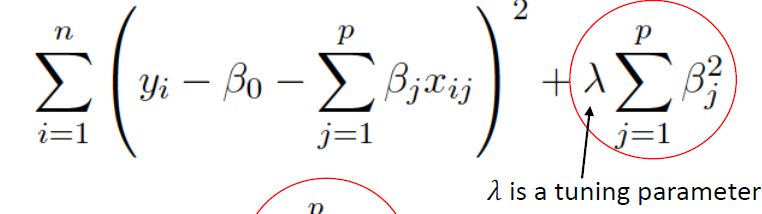
• Do not assume any form of f(x), hence are more flexible, e.g. KNN

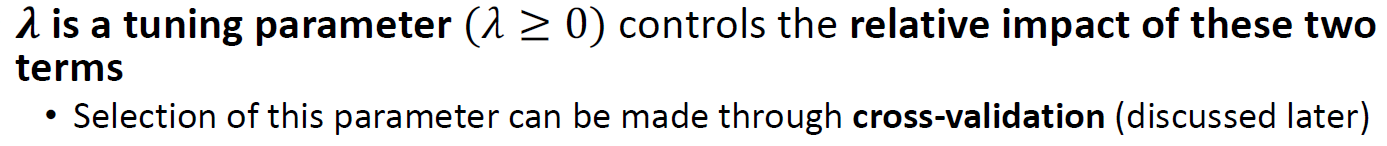
• Typically, the parametric approach outperforms the nonparametric

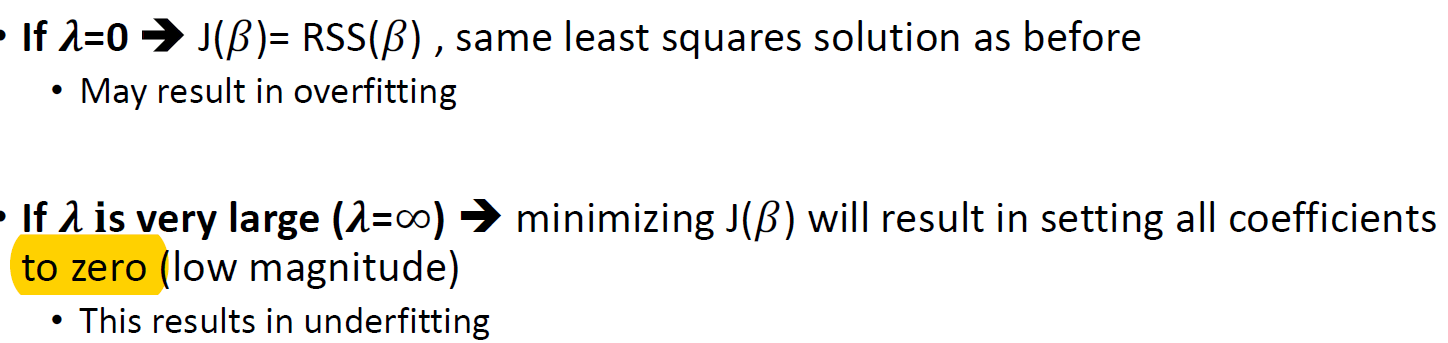
one if the model selected is close to the true one

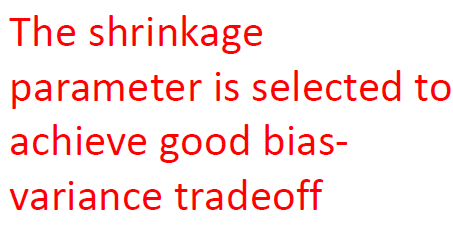
**Curse of dimensionality**: Number of training data needed grows exponentially with the number of features.

**Ridge and Lasso regression** fit and single model and include all features , but use a technique that shrinks some coefficient estimates towards zero.









pron: reduce variance (avoid overfitting); fit single model

conn: All coefficient shrink to zero

challenge in model interpretation

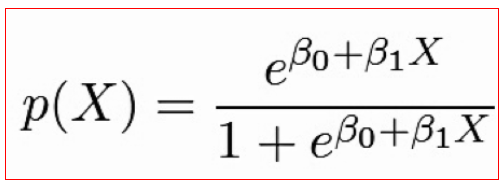
**Lasso regression**

When the tuning parameter (!) is sufficiently large, some coefficients will be forced to be zero

**Ridge** performs better when the response is a function of all features

**Lasso** performs better when small number of features are in fact related to the response

**Logistic model**



Get coefficients that maximizes the likelihood, then use them for predictions

