0) Theory

(For "https://archive.ics.uci.edu/ml/datasets")

- "Multivariate" means a dataset that has multiple variables
- "Associated tasks" are tasks the data is good for, like classification and regression
- "Number of instances" so like, if there is data about students' performance, that is the number of students
- "Number of attributes" ie how many attributes there are for every instance

- Deep learning is a subset of ML which is a subset of AI
- AI: A program that can sense, reason, act, and adapt. A means of simulating intelligent behaviour
- Deep learning: Subset of ML where multilayered neural network learn from vast amount of data. It involves using very complicated models called "deep neural networks"
- In deep learning/neural networks, it will identify the features/attributes itself (unlike in classic ml where we have to feed features)
- AI breakthroughs: image classification and machine translation, aka natural language processing
- **Estimation:** It is the application of an algorithm (like taking an average)
- A good model omits unimportant details while retaining whats important
- Ω means parameters (Maybe these parameters represent coefficients relating the features x with expected target values in a regression model)
- In classification we need the following:
    - Features/attributes that can be quantified
    - Labels that are known
    - Method to measure similarity

ML

It is said to be the process through which computers are said to "learn" and infer predictions from data. Algorithms whose performance is improved as they're exposed to data overtime. The study and construction of programs that aren't explicitly programmed, but learn patterns as they are exposed to more data overtime. These programs learn from repeatedly seeing data rather than explicitly programmed by humans. Two types of ML, supervised and unsupervised

Aspect	Supervised	Unsupervised
Target Column	We will have a target column	We will not have a target column

Aspect	Supervised	Unsupervised
Goal	Be able to predict the label of an entry given its features	Find structure in data
Example	Fraud detection	Customer segmentation

Two main modelling approaches in supervised learning is regression (predicting a value) and classification (predicting a category)
Unsupervised learning entails two things: clustering and dimensionality reduction

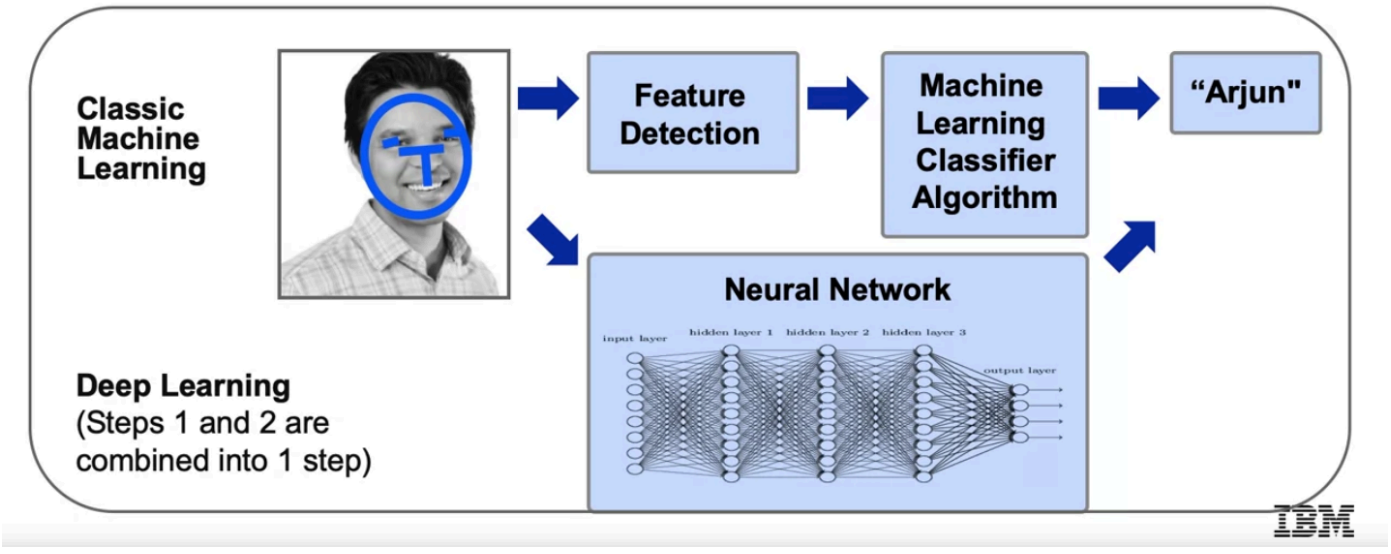
ML Workflow

1. **Problem statement:** What problem are we trying to solve
2. **Data collection:** What data is needed to solve it
3. **Data exploration and pre-processing:** How to clean data and make it usable
4. **Modeling:** Building a model to solve the problem
5. **Validation:** Was the problem solved
6. **Decision making and deployement:** Communicate to stakeholders or put into production

Supervised ML

Two types of supervised ML:

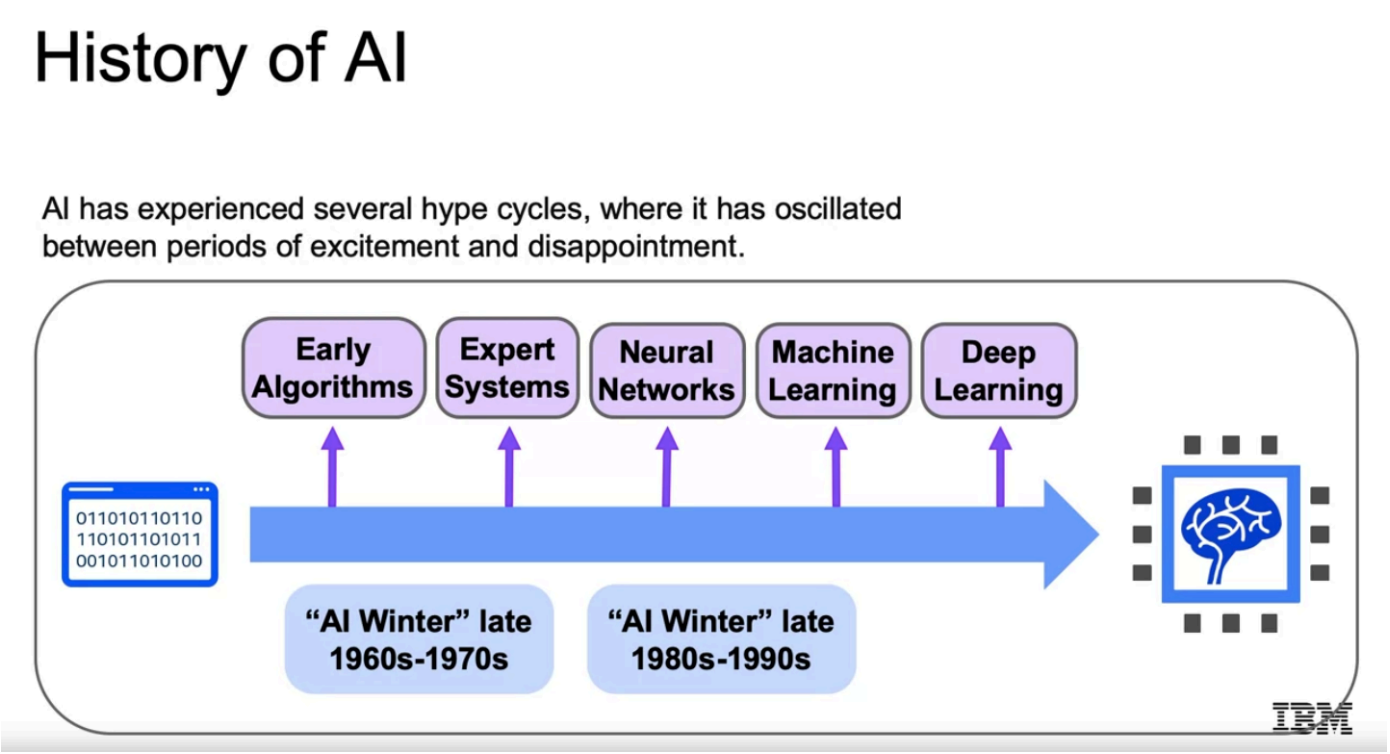
- **Regression:** The outcome in continuous (numerical), For example house prices, box office revenues, event attendance, etc
- **Classification:** Outcome is a category. For example detecting fraud, customer churn, etc. For classification, we need quantified features, labels that are known, and a method to measure similarity



History of AI

- This era of AI is different because we have bigger datasets, more storage, faster and powerful computers, and neural networks
- Some uses of AI in this era:
  - **Health:** Enhanced diagnostics, drug discovery, patient care, research, sensory aids
  - **Industrial:** Factory automation, predictive maintenance, precision agriculture, field automation
  - **Finance:** Algorithmic trading, fraud detection, research, risk mitigation, personal finance
  - **Energy:** Oil and gas exploration, smart grid, operational movement, conservation
  - **Government:** Defence, safety & security, data insights, engagement, smarter cities
  - **Transport:** Autonomous cars, automated trucking, aerospace, shipping, search & rescue
  - **Other:** Personalised content & advertising, education, gaming, service industries, sports
- Real life examples:
  - Finding fastest route in navigation
  - Uber and Lyft predict real time demand
  - Social medias use AI for personalised and targeted content and ads
  - Siri and alexa use NLP
  - Object detection in self driving cars or to determine when to take a photo

(Pink things are breakthroughs, blue things are period where development slowed down)



1950s: Early AI

1950: Alan Turing developed the Turing test, to test a machine's ability to exhibit intelligent behavior.

1956: Artificial Intelligence was accepted as a field at the Dartmouth Conference.

1957: Frank Rosenblatt invented the perceptron algorithm. This was the precursor to modern neural networks.

1959: Arthur Samuel published an algorithm for a checkers program using machine learning.

The First “AI Winter”

1966: ALPAC committee evaluated AI techniques for machine translation and determined there was little yield from the investment.

1969: Marvin Minsky published a book on the limitations of the Perceptron algorithm which slowed research in neural networks.

1973: The Lighthill report highlights AI’s failure to live up to promises.

The two reports led to cuts in government funding for AI research, leading to the first “AI Winter”.

1980’s AI Boom

Expert Systems - systems with programmed rules designed to mimic human experts.

- Ran on mainframe computers with specialized programming languages (e.g. LISP).
- Were the first widely-used AI technology, with two-thirds of "Fortune 500" companies using them at their peak.
- 1986: The “Backpropagation” algorithm is able to train multi-layer perceptrons, leading to new successes and interest in neural network research.

Another AI Winter (1980s –1990s)

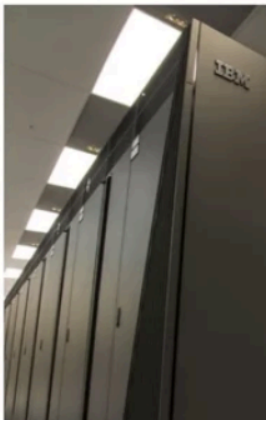
Expert systems' progress on solving business problems slowed.

- Expert systems began to be melded into software suites of general business applications (e.g. SAP®, Oracle®) that could run on PCs instead of mainframes.
- Neural networks didn't scale to large problems.
- Interest in AI in business declined.

1990s - 2000s: Machine Learning

AI solutions had successes in speech recognition, medical diagnosis, robotics, and many other areas.

- AI algorithms were integrated into larger systems and became useful throughout industry.
- The Deep Blue chess system beat world chess champion Garry Kasparov.
- Google's search engine launched using artificial intelligence technology.



2006: Rise of Deep Learning

2006: Geoffrey Hinton publishes a paper on unsupervised pre-training that allowed deeper neural networks to be trained.

- Neural networks are rebranded as deep learning.

2009: The ImageNet database of human-tagged images is presented at the CVPR conference.

2010: Algorithms compete on several visual recognition tasks at the first ImageNet competition.

Deep Learning Breakthroughs (2012 – ?)

2012: Deep learning beats previous benchmark on the ImageNet competition.

2013: Deep learning is used to understand "conceptual meaning" of words.

2014: Similar breakthroughs appeared in language translation.

These have led to advancements in Web Search, Document Search, Document Summarization, and Machine Translation.

Deep Learning Breakthroughs (2012 – ?)

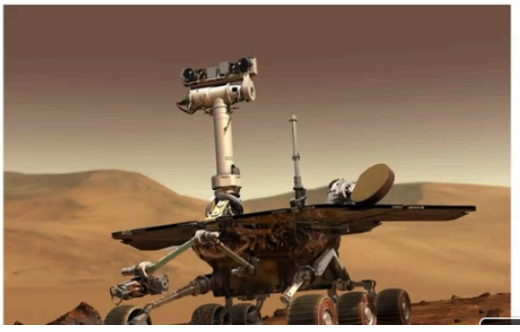
2014: Stanford team creates computer vision algorithm that can describe photos.

2015: Deep learning platform TensorFlow is developed.

2016: DeepMind's AlphaGo, developed by Aja Huang, beats Go master Lee Se-dol.

2018: Waymo launches commercial self-driving-car service in suburbs of Phoenix

2019: IBM Project Debater is able to have a full debate with rebuttal with champion human debater



Data cleansing

- **Outlier:** An observation in data that is distant from other observations. If we do not identify and deal with outliers, they can have a significant impact on the model. It is important to remember that some outliers are informative and provide insight to the data. Some policies for dealing with outliers are:
    - **Remove the outlier** (Pro: Dont have to deal with the outlier | Con: We may lose the whole observation)
    - **Replace the outlier w/ mean or median** (Pro: The whole observation is not lost | Con: We may lose an important value)
    - **Predict what the value would have been** (by using similar observations or using regression)
    - **Keep the value**
- **Residual:** It is the difference between actual and predicted value of label. It represents model failure. Some approaches to calculating/detecting residuals are:
    - Standardized: Residual divided by standard error
    - Deleted: Remove the observation from the dataframe and see the improvement
    - Studentized: Deleted residuals divided by residual standard error
- **Problems faced while data collection**: Lack of data, too much data, or bad data
- **Messy data** Data having duplicate or unnecessary data, inconsistent text and typos, missing data, outliers, data sourcing issues

Policies of missing data:

- **Remove the row(s) entirely** (Pro: is easy to do | Con: If too many rows are removed, we can end up with a biased dataset)
- **Imputing the data:** Substitute empty values, like filling it with most common value, average value, etc (Pro: We don't lose any row | Con: We add a level of uncertainty)
- **Mask the data:** Create a category for missing values (Pro: We don't lose any row | Con: We add a level of uncertainty since we assume that all our missing values are alike)

Exploratory data analysis: It is an approach to analyse datasets to summarise their main characteristics, often with visual methods. It allows us to get an initial feel for the data. It lets us determine if the data makes sense, or if further cleaning of data is needed. It also helps us to identify patterns and trends in the data.
Some summary statistics: Avg, min, max, correlations, etc
Visualisations: Histograms, scatter plots, box plots, etc (using matplotlib, pandas-matplotlib, or seaborn)

Transformation

Variable selection involved choosing the set of features to be included in the model. Variables must often be transformed before they can be included in models. In addition to log and polynomial transformations, it can include encoding (converting non-numeric features to numeric features) and scaling (converting the scale of numeric data so it is comparable)

Encoding

Encoding can be of 2 types:

- **Nominal:** Categorical values take values in unordered categories (red, blue, green, etc)
- **Ordinal:** Categorical values take values in ordered categories (low, medium, high, etc)

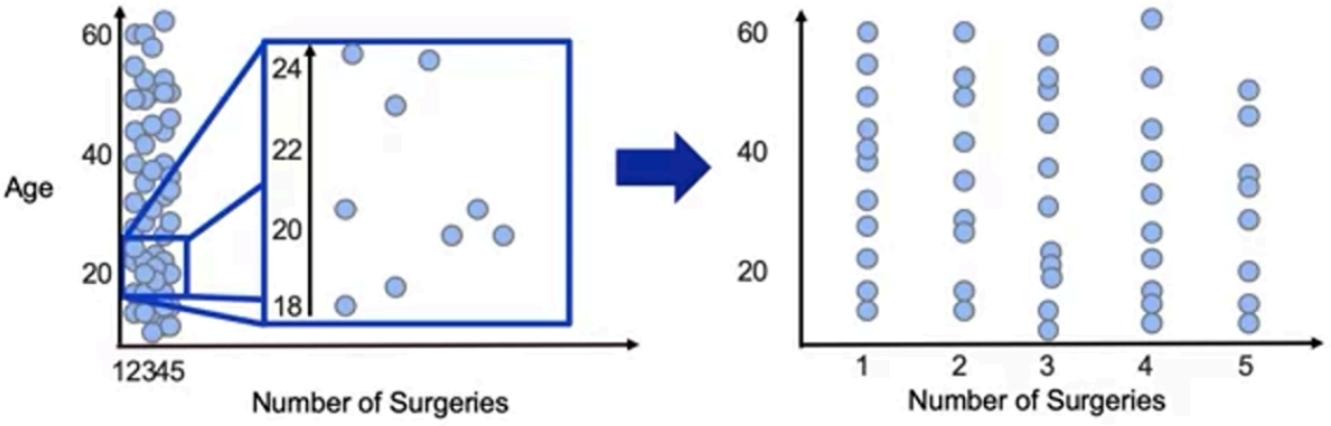
Ways to do encoding are:

- **Binary encoding:** Converts variables to 0 or 1
- **One hot encoding:** Converts variables that take multiple values into binary variables, one for each category. This creates several new categories
- **Ordinal encoding:** Converts ordered categories into numerical values

```
from sklearn.preprocessing import LabelEncoder, LabelBinarizer, OneHotEncoder, OrdinalEncoder
# OR
from pandas import get_dummies
```

Scaling

Useful in proximity/distance based algorithms like KNN



- **Standard scaling:** Converts features to standard normal variables (by subtracting the mean and dividing by standard error)
- **Minmax scaling:** Converts variables to continuous variables in the (0,1) interval by mapping the min value to 0 and max value to 1. This is very sensitive to outliers

- **Robust scaling:** Similar to minmax scaling but instead maps the interquartile range (75^{th} percentile value minus the 25^{th} percentile value) to (0,1). This means that the variable itself takes values outside of the (0,1) interval

```
from sklearn.preprocessing import StandardScaler, MixMaxScaler, RobustScaler
```

Parametric model

- **Inference:** It involves putting an accuracy on the estimate (like standard error of avg)
- If inference is about trying o find out the data generation process (DGP), we can say that a statistical model is a set of possible distributions or maybe even regressions
- **Parametric model:** It is a particular type of statistical model. It also has a set of distributions or regressions, but they have a finite number of parameters
- A value of a non parametric inference is creating a distribution of a data using a histogram. In this case, we aren't specifying parameters

For example, in a parametric model, data related to the customer might include:

- The expected length of time as a customer
- The expected amount spent over time
  To estimate lifetime value, we make assumptions about the data. These assumptions can be parametric (assuming a specific distribution) or non parametric

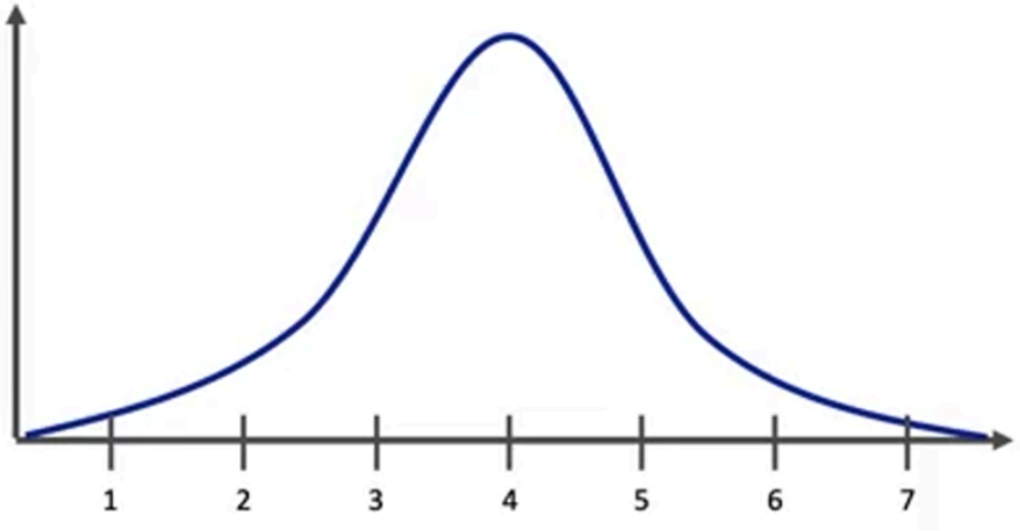
The most common way of estimating parameters (in a parametric) is through maximum likelihood estimation (MLE)

The **likelihood function** is related to probability and is a function of the **parameters** of the model:

$$\mathcal{L}_n(\theta) = \Pi_{i=1}^{n} f(X_i, \theta)$$

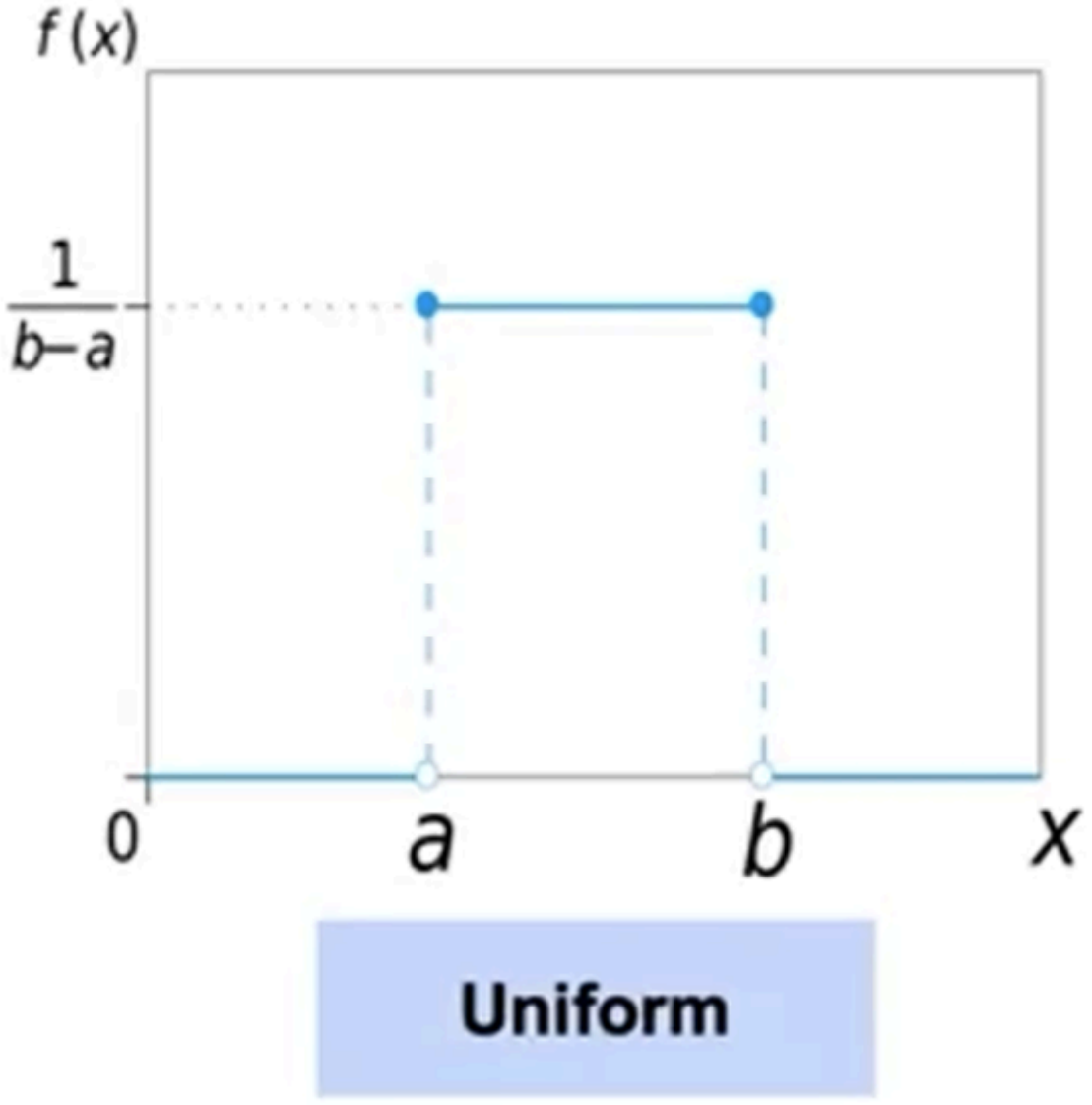
Function of the parameters

We choose the value of θ (parameters) that maximizes the likelihood function.

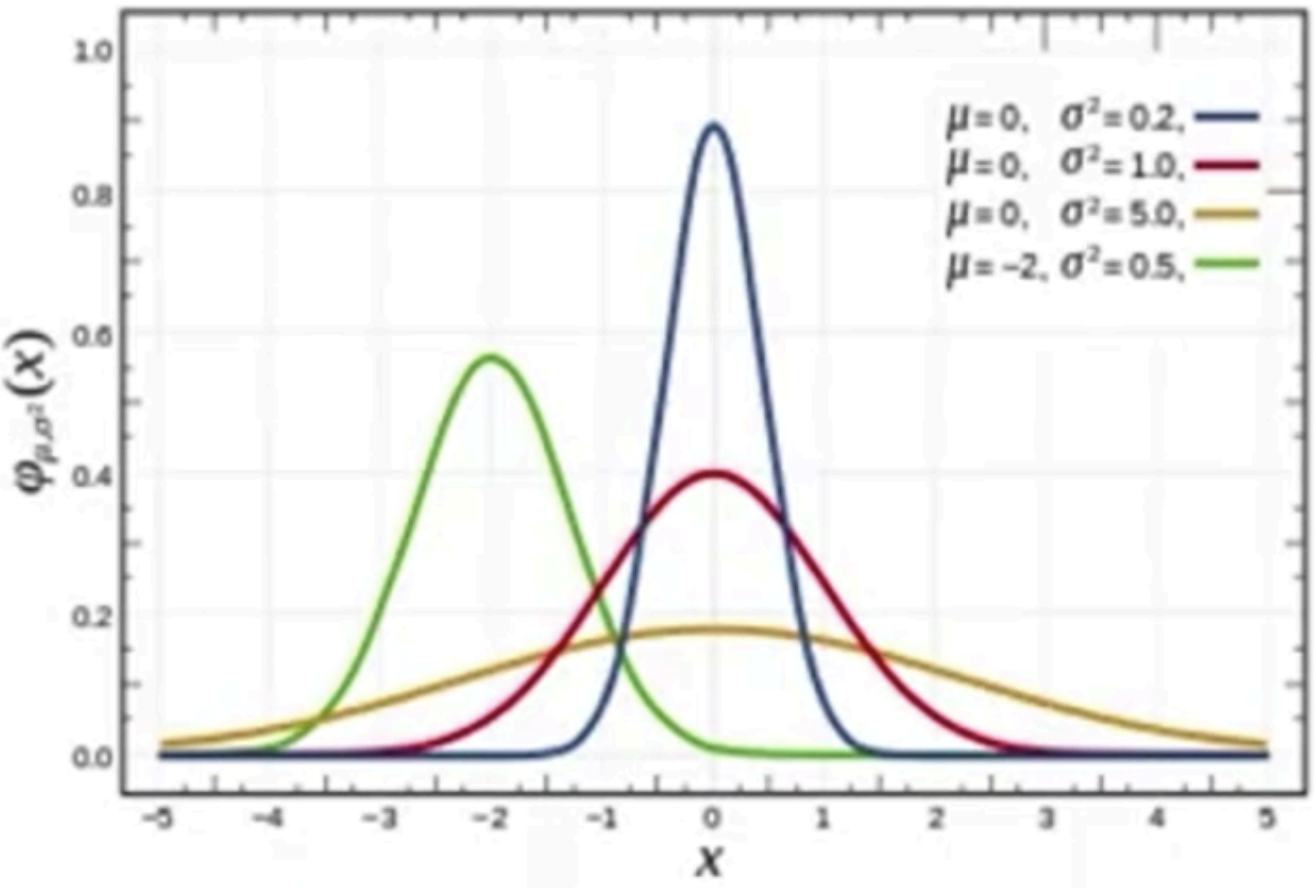


Some commonly used distributions are:

Uniform

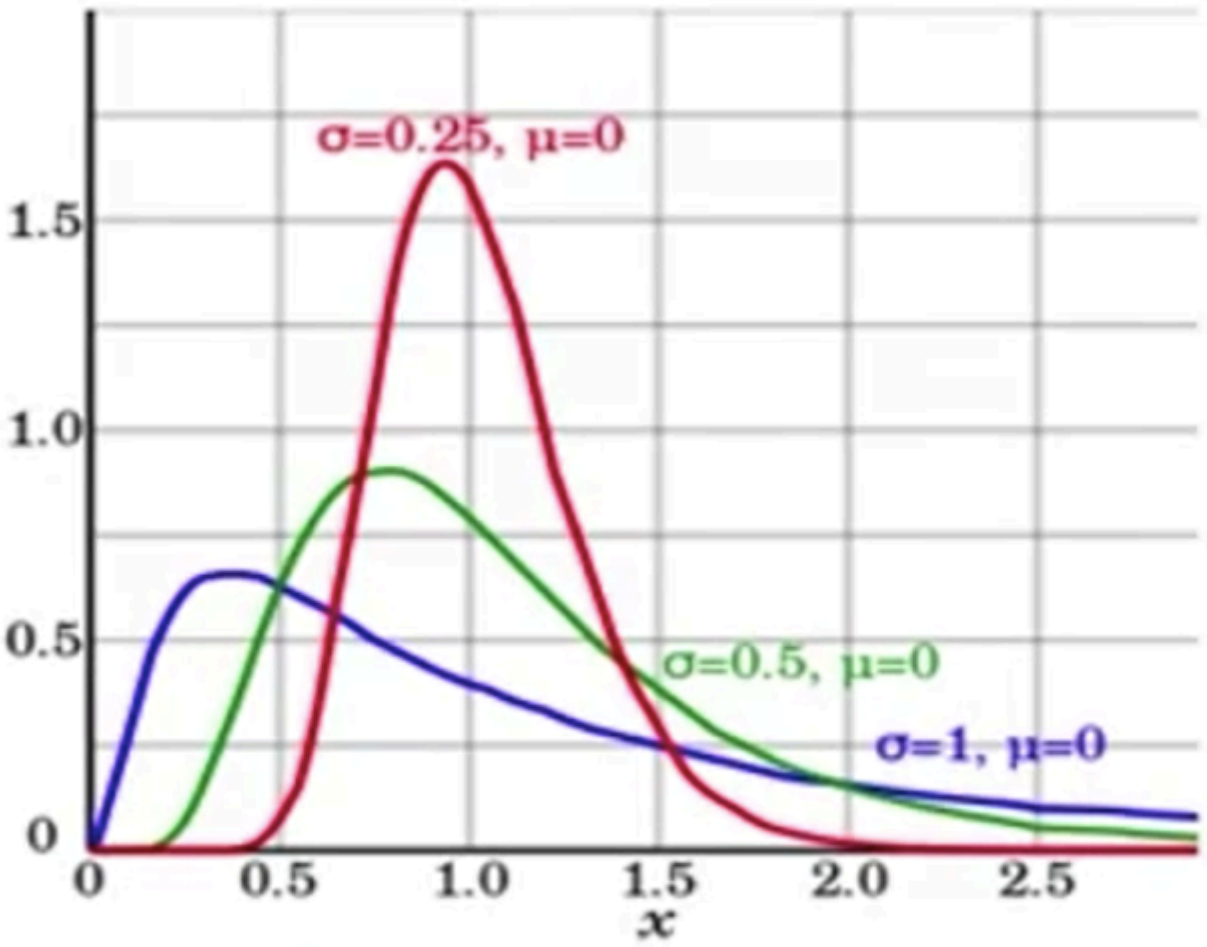


Normal/Gaussian



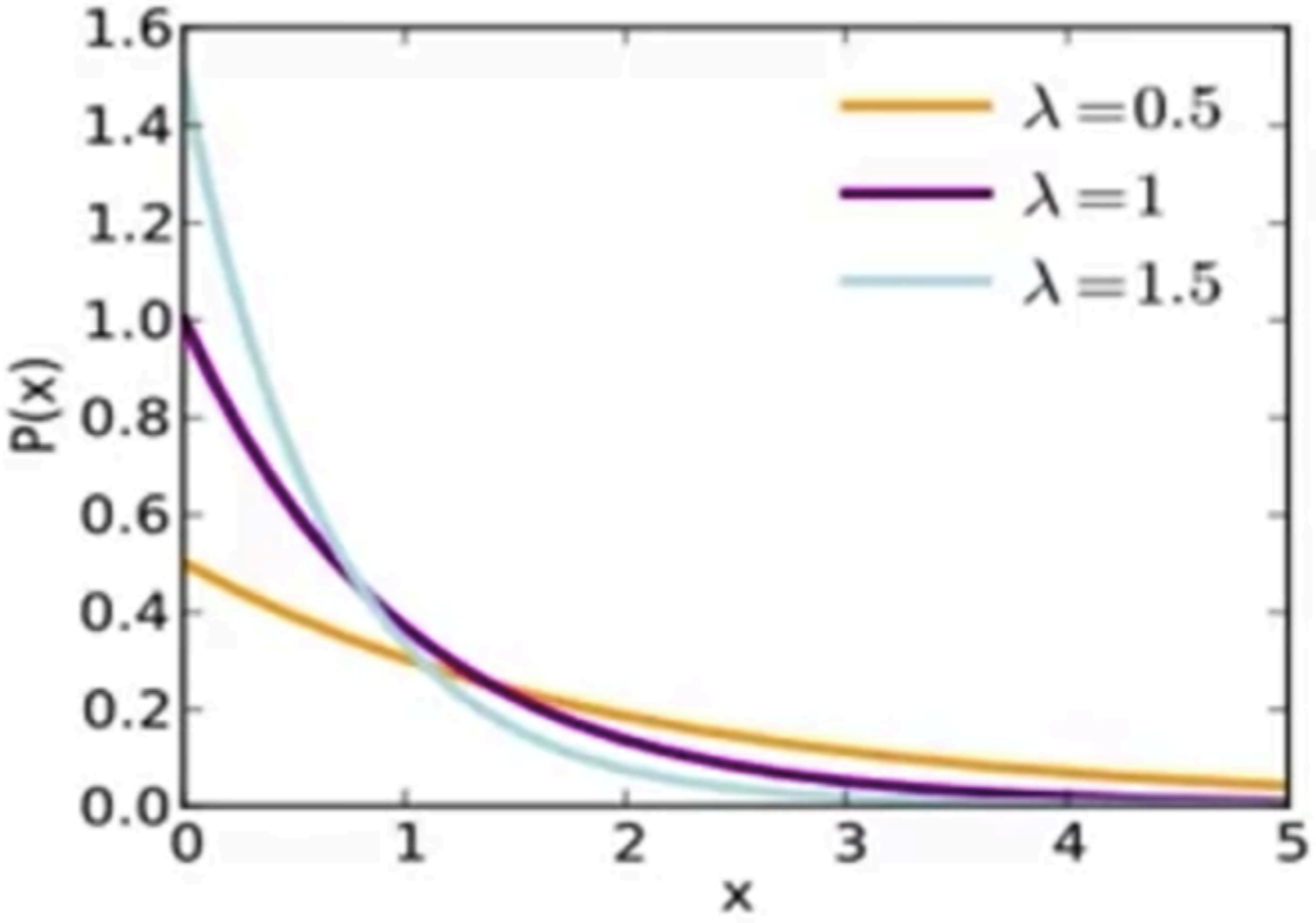
Gaussian / Normal

Log Normal



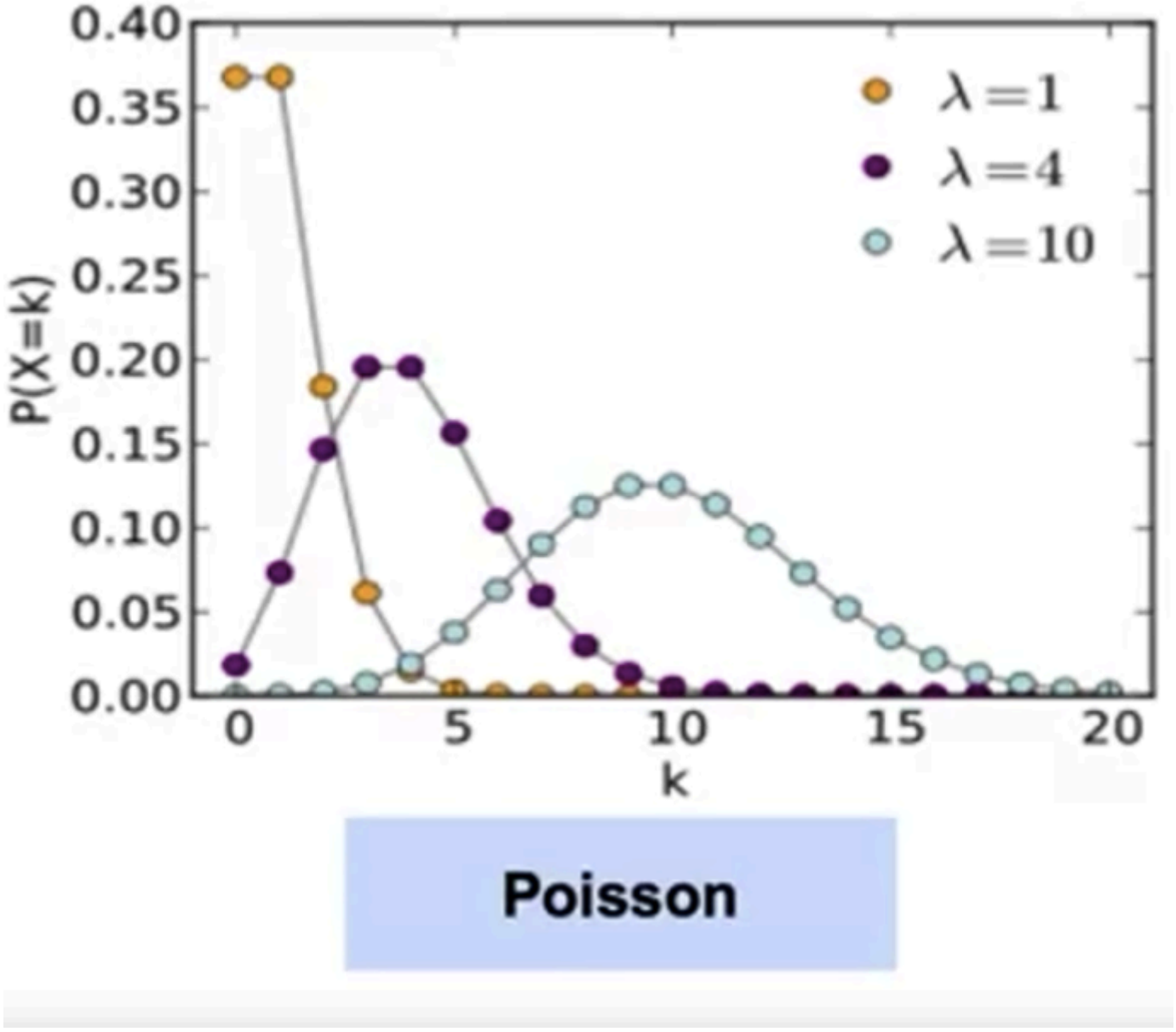
Log Normal

Exponential



Exponential

Poisson



Frequentist vs Bayesian statistics

In both, frequentist and bayesian statistics, we use much of the same math and formula. The main element that differs is the interpretation

A **frequentist** is concerned with repeated observations in the limit. The frequentist approach is:

1. **Derive** the probabilistic property of a procedure
2. **Apply** the probability directly to the applied data

A **bayesian** describes parameters by probability distributions. Before seeing any data, a prior distribution (like an initial guess) is formulated. This prior distribution is then updated after seeing the data. The resultant distribution is known as the posterior distribution

Type I vs Type II error

- **Hypothesis:** It is a statement of a popular parameter. We create two hypothesis, the null hypothesis (H_0) and the alternative hypothesis (H_1 or H_A). We decide which one to call the null hypothesis depending on how the problem is set up
- A **hypothesis testing procedure** gives us a rule to decide, for which values of the test test statistic do we accept H_0, and for which do we reject H_0 and accept H_1
- The **likelihood ratio** is called a test statistic, we use it to decide wether to accept or reject H_0
- **Rejection region:** It is the set of values of the test statistics that lead to the rejection of H_0
- **Acceptance region:** It is the set of values of the test statistics that lead to the acceptance of H_0
- **Null distribution:** It is the test statistic's distribution when the null is true
- **Significance level:** It is a probability threshold below which the null hypothesis will be rejected
- **p-value:** It is the smallest significance level at which the null hypothesis would be rejected

Significance level and p-values

A significance level is a probability threshold below which the null hypothesis can be rejected. You must choose the significance level before computing the test statistic. It is usually .01 or .05.

A p-value is the smallest significance level at which the null hypothesis would be rejected. The confidence interval contains the values of the statistic for which we accept the null hypothesis.

Correlations are useful as effects can help predict an outcome, but correlation does not imply causation.

When making recommendations, one should take into consideration confounding variables and the fact that correlation across two variables do not imply that an increase or decrease in one of them will drive an increase or decrease of the other.

Spurious correlations happen in data. They are just coincidences given a particular data sample.

		Decision	
		Accept H_0	Reject H_0
Truth	H_0	Correct	Type I error
	H_1	Type II error	Correct

Power of a test = 1 – P(Type II Error)

Suppose we use data on customer characteristics to predict who will churn over the next year.

In our data, customers who have been with the company for longer are less likely to churn.

This could be due to an underlying effect, or due to chance:

- A **Type I Error** occurs when this effect is due to chance, but we find it to be significant in the model.
- A **Type II Error** occurs when we ascribe the effect to chance, but the effect is non-coincidental.

Interpretation vs Prediction

Interpretation

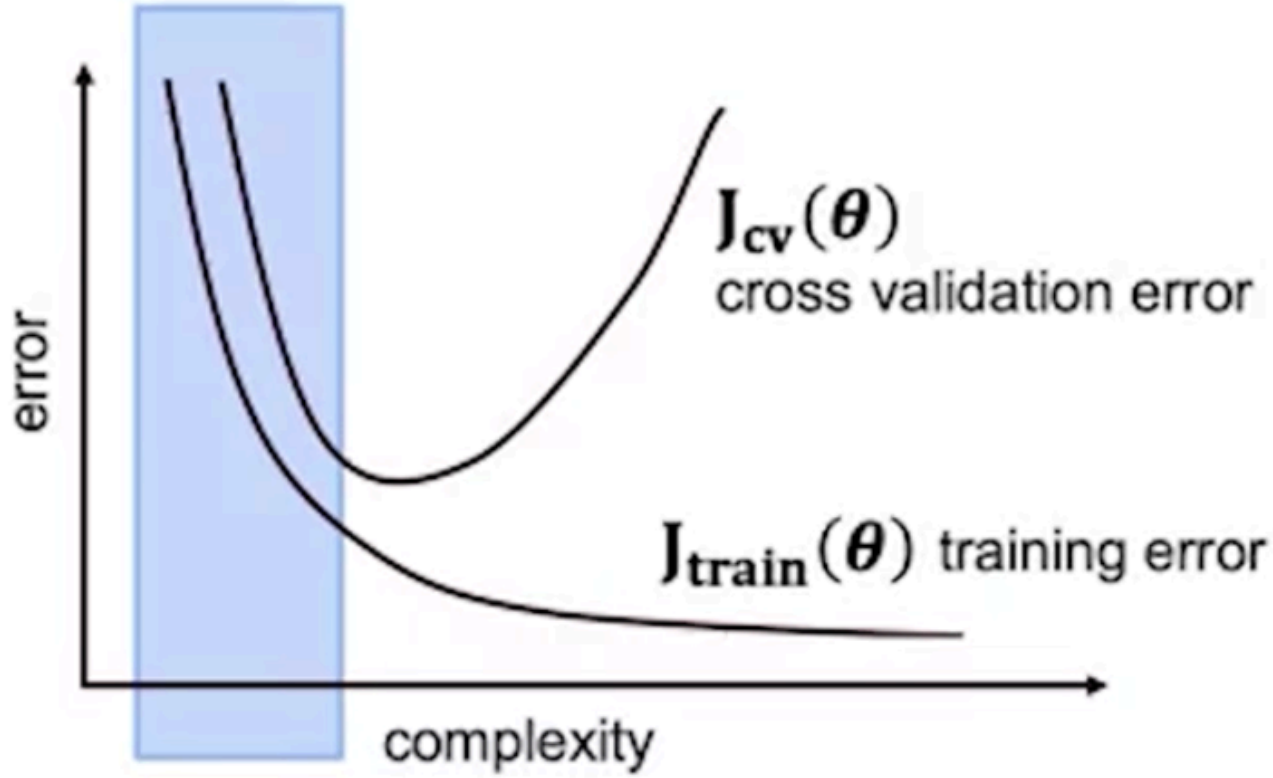
- In some cases, the primary objective is to train a model and find insights from the data (like which features affect the label the most)
- In $y_p = f(\Omega,x)$, the interpretation approach uses Ω (parameters) to give us insight into a system
- Common workflow:
  - Gather x,y; Train model by finding Ω that gives best prediction $y_p = f(\Omega,x)$
  - Focus on Ω rather than y_p to give insights
- Example interpretation exercises:
  - x = customer demographics; y = sales data; examine Ω to understand loyalty by segment
  - x = car safety features; y = traffic accidents; examine Ω to understand what features make a car safer
  - x = marketing budget; y = movie revenue; examine Ω to understand marketing effectiveness

Prediction

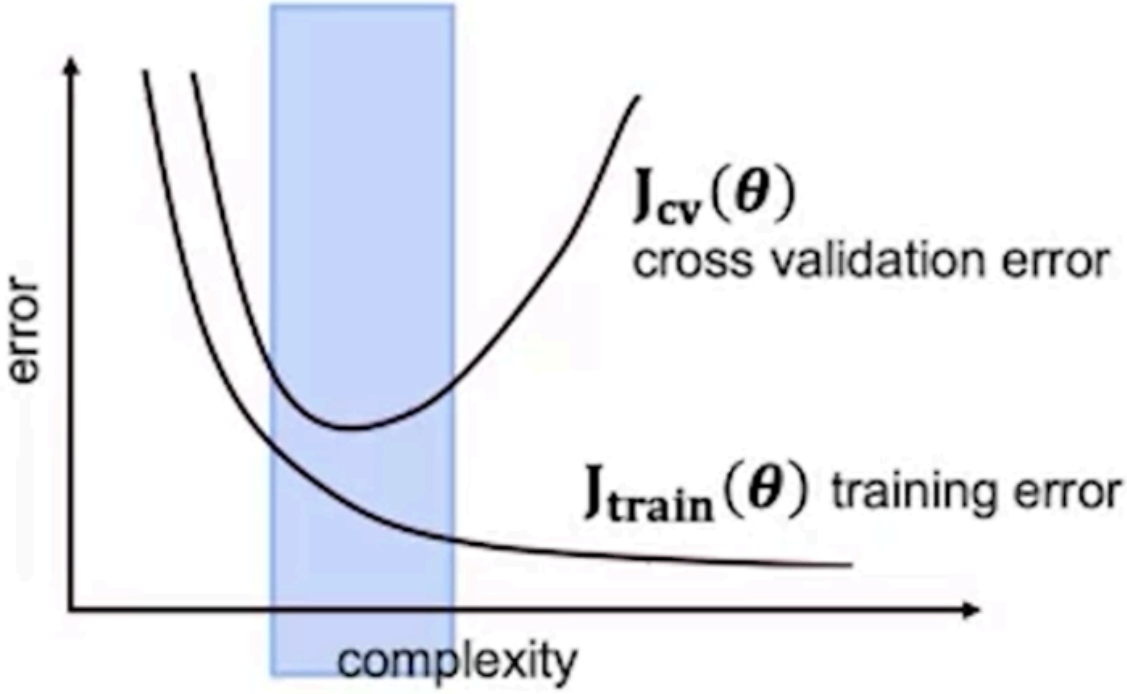
- In some cases, the primary objective is to make the best prediction
- In $y_p = f(\Omega,x)$, the interpretation approach compares y_p (predicted value) with y (actual value)
- The focus is on performance metrics, which measure he quality of a model's prediction
  - Performance metrics usually involve some measure of closeness between y_p and y
  - Without focusing on interpretability, we risk having a black-box model
- Example interpretation exercises:
  - x = customer purchase history; y = customer churn; focus on predicting customer churn
  - x = financial information; y = flagged default/non-default; focus on predicting loan default
  - x = purchase history; y = next purchase; focus on predicting the next purchase

Cross validation

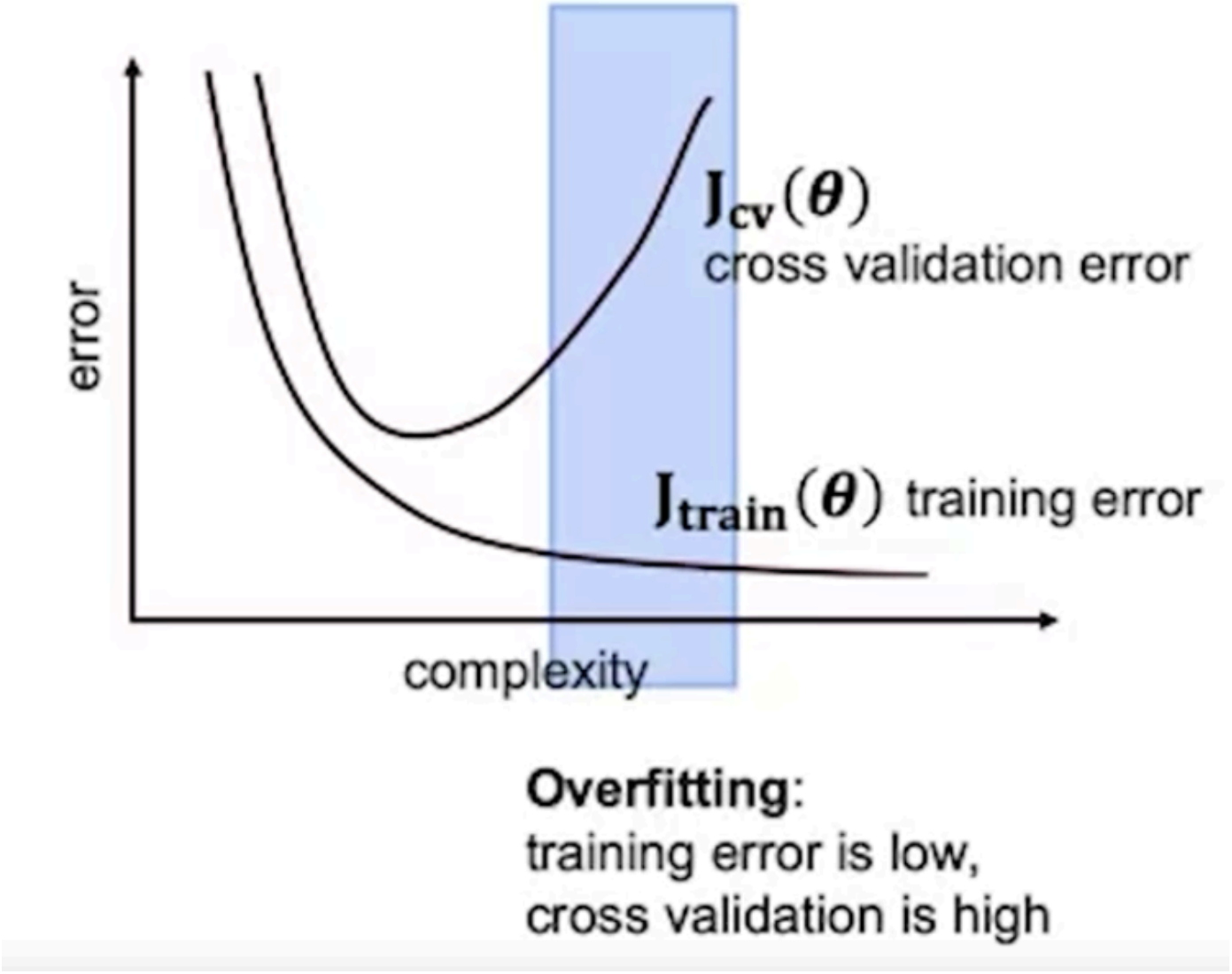
Basically we make many test-train split sets where like, for example there are rows 1-20. So the test data in the sets would be 1-5, 6-10, 11-15, and 16-20. The mean of the error resulting in the models made will be referred to as the cross validation error



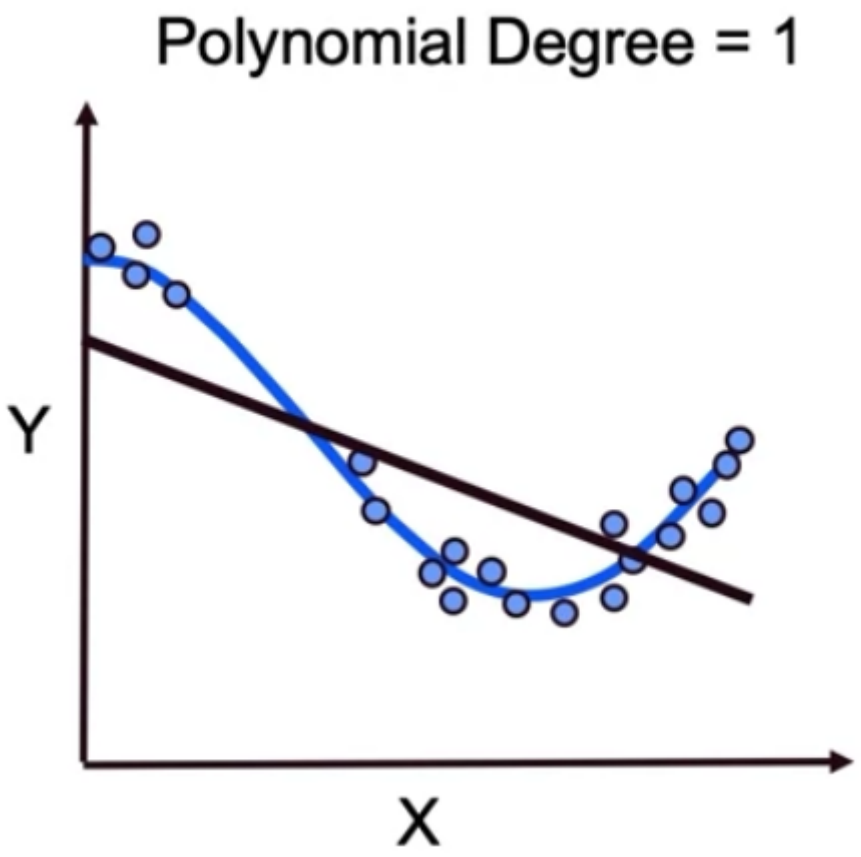
Underfitting:
training and cross validation error are both high

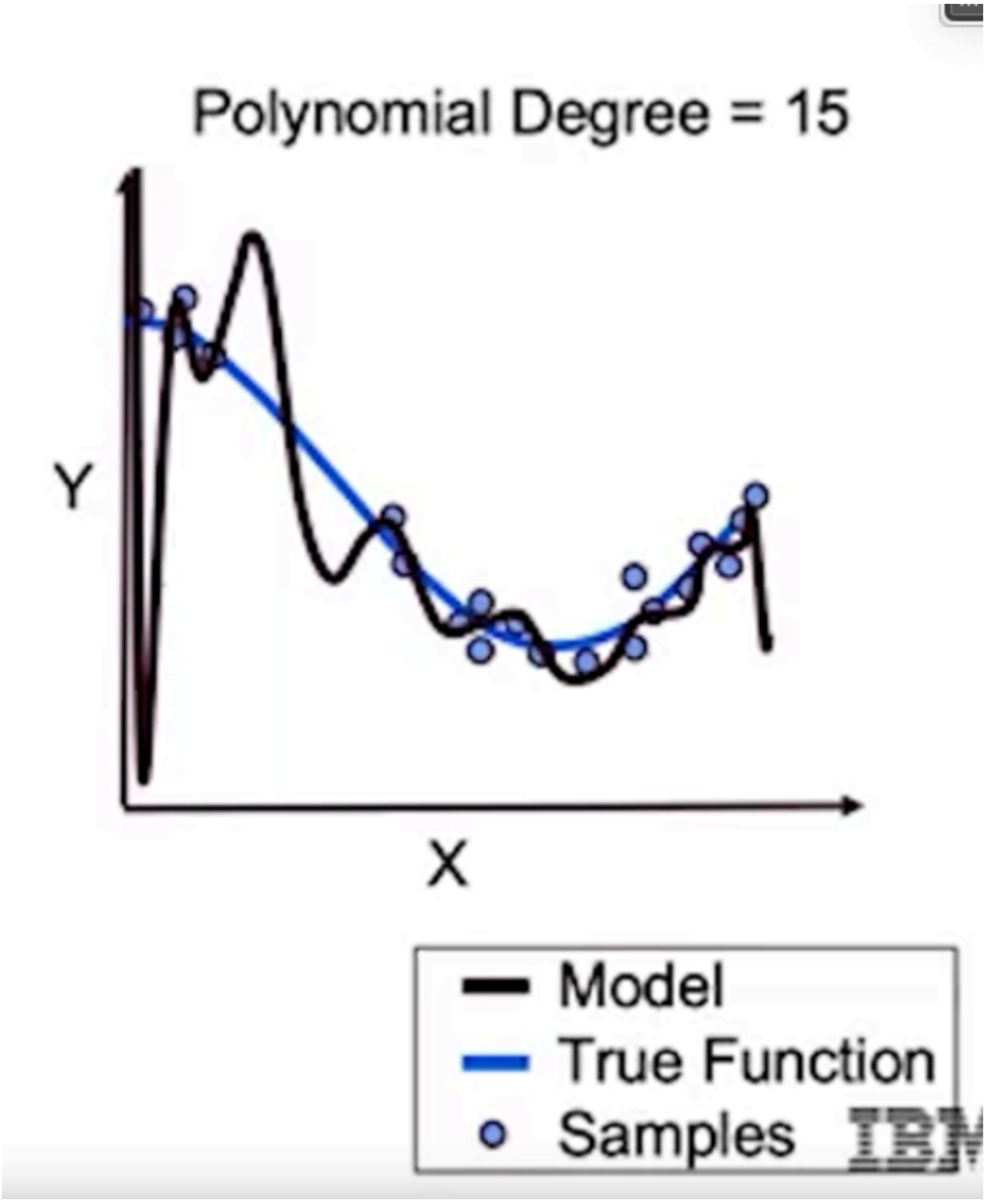


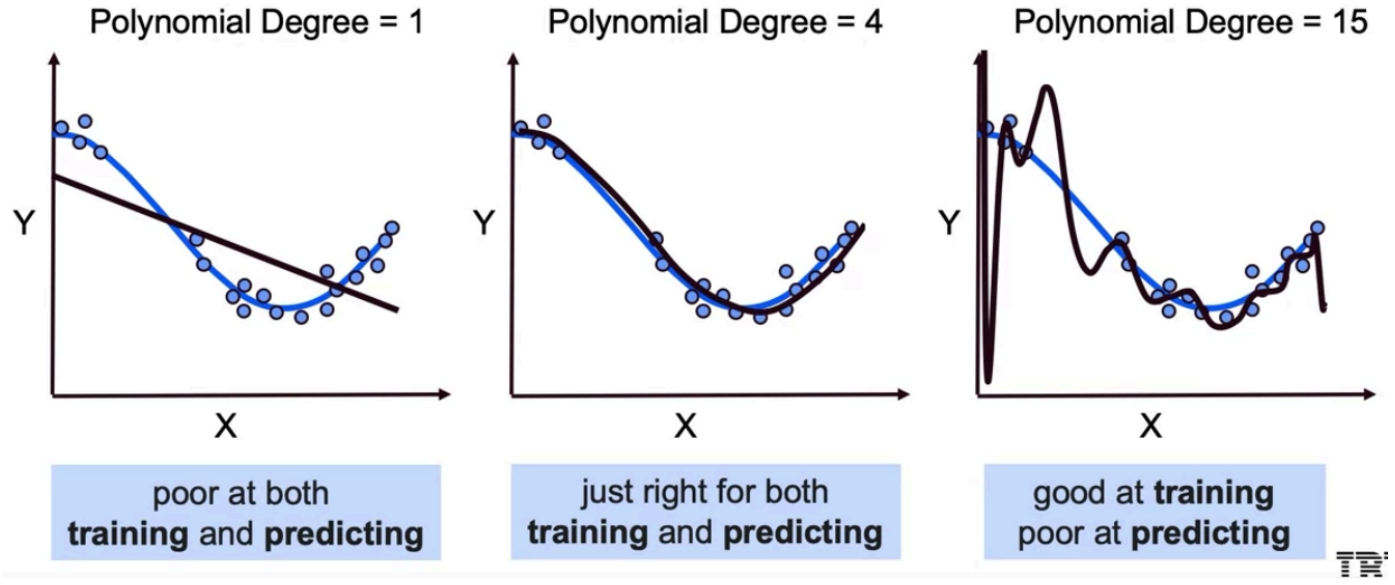
Just right:
training and cross validation errors are both low



An example of underfitting and overfitting are:

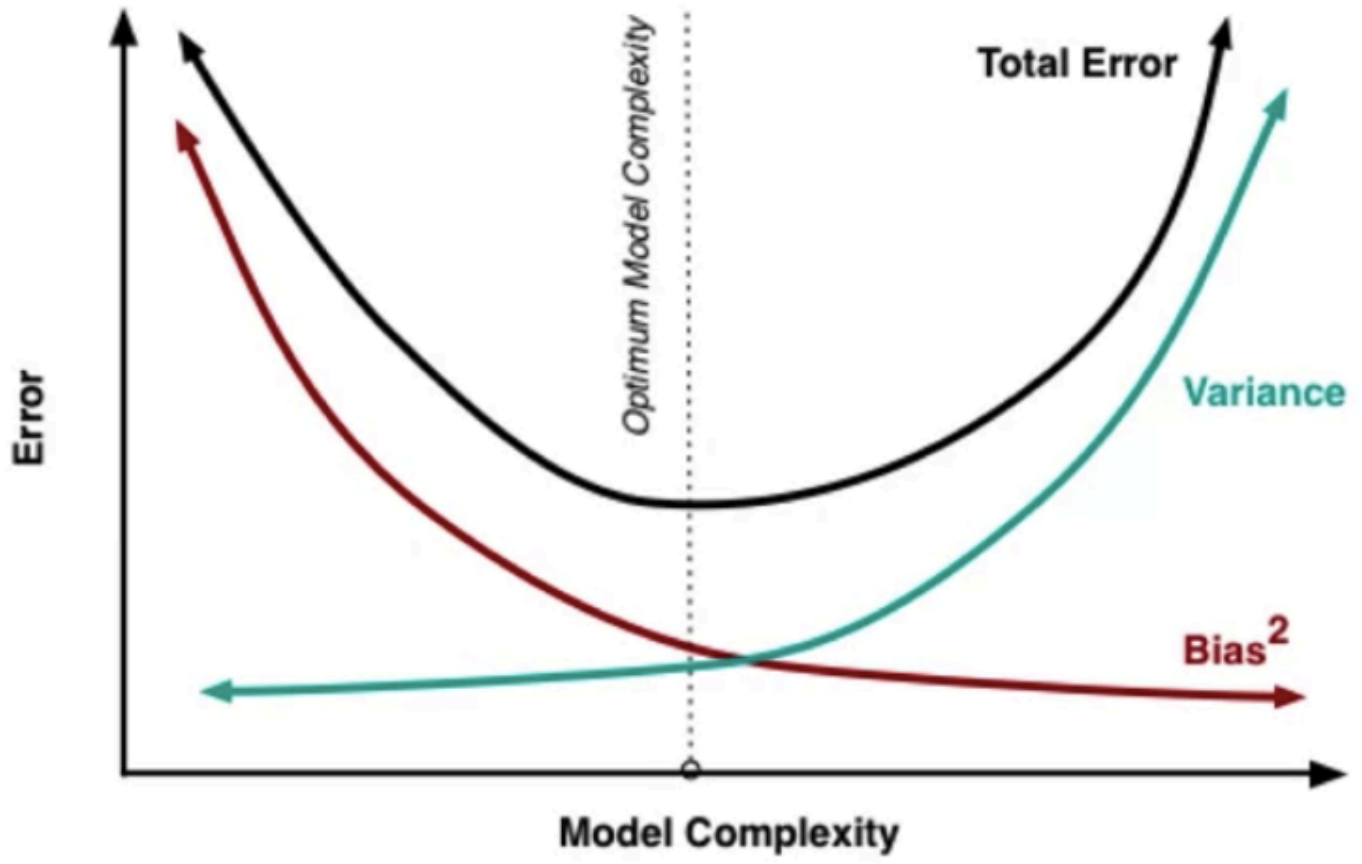

Polynomial Degree = 1
Y
X


Polynomial Degree = 15
Y
X
Model
True Function
Samples


Polynomial Degree = 1
Y
X
poor at both
training and predicting
Polynomial Degree = 4
Y
X
just right for both
training and predicting
Polynomial Degree = 15
Y
X
good at training
poor at predicting

Bias and variance

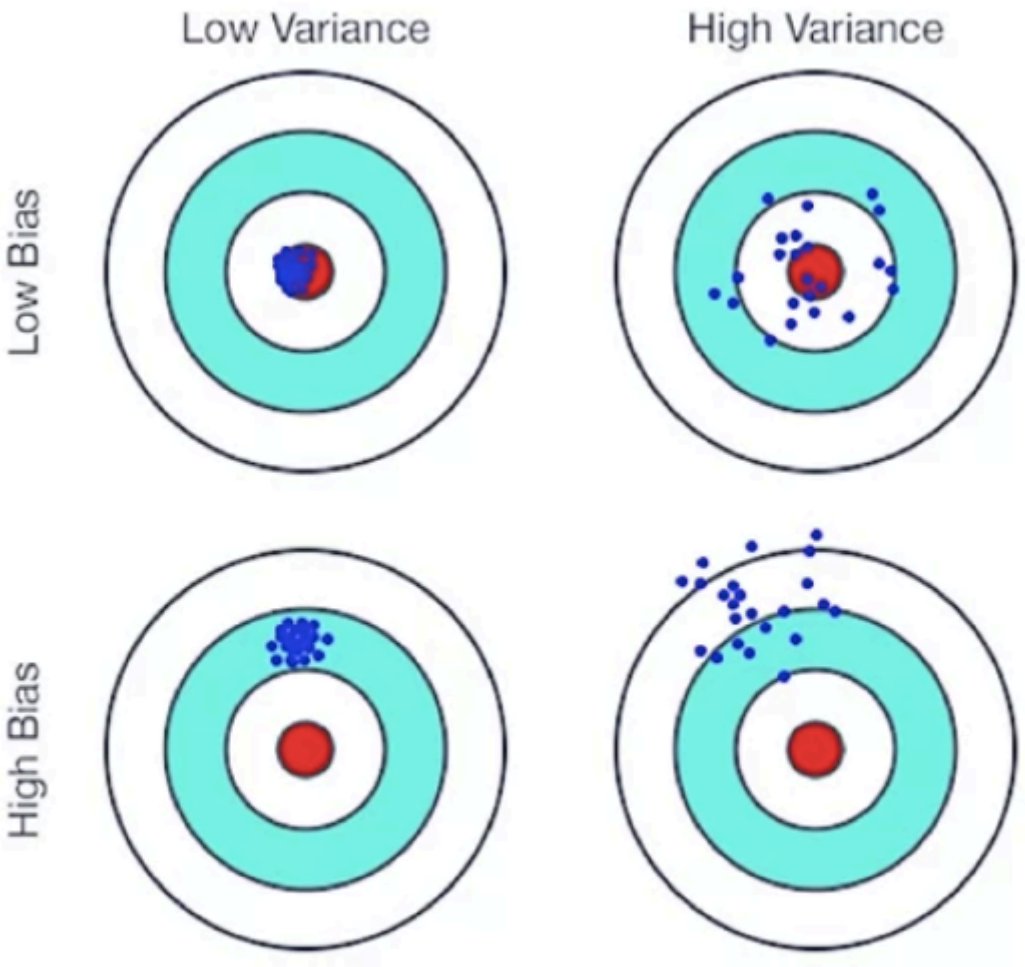
- **Bias:** Tendency to miss true values. Can be caused by missing information, overly simplistic assumptions, missing real patterns (underfitting)
- **Variance:** Tendency to be inconsistent, or of predictions to fluctuate. Can be caused by sensitivity to output to small changes in data, overly complex or poorly fit models
- **Tendency:** Expectation of out of sample behaviour over many training test samples
- **Bias variance tradeoff:** Model adjustments that decrease bias often increase variance, and vice versa. Finding the best model means choosing the right amount of complexity. We want a model elaborate enough not to underfit, but not so elaborate that it overfits



Visualizing the complexity tradeoff

3 sources of model error:

- Being wrong (high bias)
- Being unstable (high variance)
- Unavoidable randomness (Irreducible error) (Present in even the best models)



Intuitive view of **bias** and **variance**

How Correlations are Important

We should be careful about changing *X* with the hope of changing *Y*.

X and *Y* can be correlated for different reasons:

- *X* causes *Y* (what we want).
- *Y* causes *X* (mixing up cause-and-effect).
- *X* and *Y* are both caused by something else (confounding).
- *X* and *Y* aren't really related, we just got unlucky in the sample (spurious).