Practical issues and tips

(avoid machine learning going mad)

John Joseph Valletta

University of Exeter, Penryn Campus, UK

April 2019

Overview

- Feature selection
- The wrong way to do cross-validation
- Imbalanced data
- Machine learning gone mad
- Practical tips in machine learning

Feature selection

Often one wants to determine **which** features are best at predicting the outcome. There are three main types of feature selection methods:

1. **Filter**: Use univariate measures (e.g correlation) to assess the feature's relevance
2. **Wrapper**: Employ a greedy strategy to search for the best subset of features (e.g. stepwise selection)
3. **Embedded**: Feature selection is an implicit aspect of the learning alogrithm (e.g. LASSO)

Feature selection

Often one wants to determine **which** features are best at predicting the outcome. There are three main types of feature selection methods:

1. **Filter**: Use univariate measures (e.g correlation) to assess the feature's relevance
2. **Wrapper**: Employ a greedy strategy to search for the best subset of features (e.g. stepwise selection)
3. **Embedded**: Feature selection is an implicit aspect of the learning alogrithm (e.g. LASSO)

Feature selection

Often one wants to determine **which** features are best at predicting the outcome. There are three main types of feature selection methods:

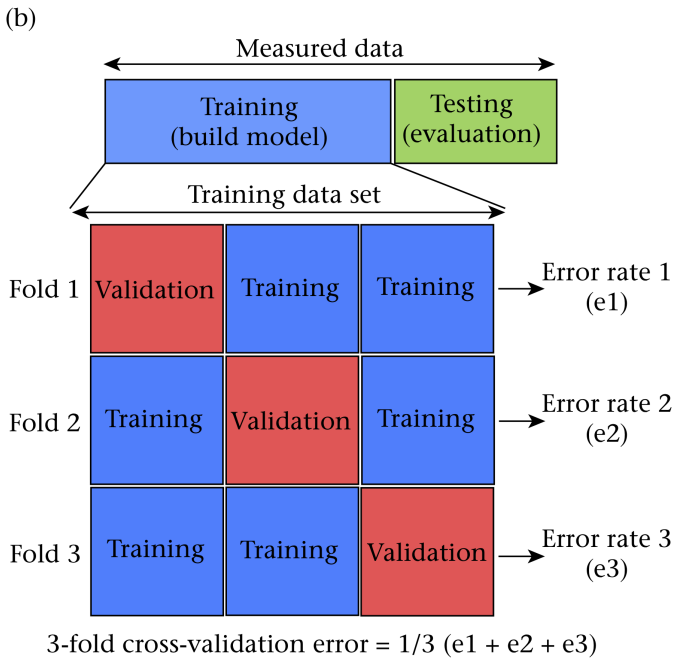
1. **Filter**: Use univariate measures (e.g correlation) to assess the feature's relevance
2. **Wrapper**: Employ a greedy strategy to search for the best subset of features (e.g. stepwise selection)
3. **Embedded**: Feature selection is an implicit aspect of the learning alogrithm (e.g. LASSO)

Feature selection

Often one wants to determine **which** features are best at predicting the outcome. There are three main types of feature selection methods:

1. **Filter**: Use univariate measures (e.g correlation) to assess the feature's relevance
2. **Wrapper**: Employ a greedy strategy to search for the best subset of features (e.g. stepwise selection)
3. **Embedded**: Feature selection is an implicit aspect of the learning alogrithm (e.g. LASSO)

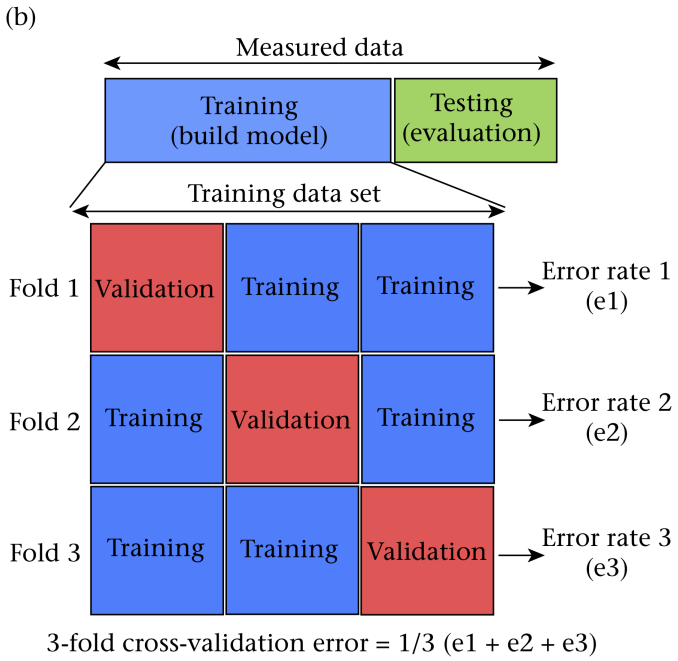
The wrong way to do cross-validation



1. Filter out features by their univariate association with the outcome using all of the training data
2. Use the selected features to build a predictive model
3. Employ k-fold cross-validation to tune the hyperparameters and estimate predictive performance

See Section 7.10.2 of "The Elements of Statistical Learning" by T. Hastie, R. Tibshirani and J. Friedman

The right way to do cross-validation



1. Filter out features by their univariate association with the outcome using all of the training data, **except** those in fold k
2. Use the selected features to build a predictive model using all of the training data **except** those in fold k
3. Use the model to predict the outcome for fold k to estimate the predictive accuracy

See Section 7.10.2 of "The Elements of Statistical Learning" by T. Hastie, R. Tibshirani and J. Friedman

Imbalanced data

- **Imbalanced data** arises in applications where one class dominates over the other. For example:
  - Easier to collect "healthy" samples than rare disease ones
  - Legitimate transactions far outweight the fraudelent ones
- **Balancing** the class distribution through **resampling** is one of the most popular solutions:

Imbalanced data

- **Imbalanced data** arises in applications where one class dominates over the other. For example:
  - Easier to collect "healthy" samples than rare disease ones
  - Legitimate transactions far outweight the fraudelent ones

- **Balancing** the class distribution through **resampling** is one of the most popular solutions:
  - **Undersampling**: Put to one side observations from the *majority* class
  - **Oversampling**: Replicate observations from the *minority* class

Imbalanced data

- **Imbalanced data** arises in applications where one class dominates over the other. For example:
  - Easier to collect "healthy" samples than rare disease ones
  - Legitimate transactions far outweight the fraudelent ones

- **Balancing** the class distribution through **resampling** is one of the most popular solutions:
  - **Undersampling**: Put to one side observations from the *majority* class
  - **Oversampling**: Replicate observations from the *minority* class

Imbalanced data

- **Imbalanced data** arises in applications where one class dominates over the other. For example:
  - Easier to collect "healthy" samples than rare disease ones
  - Legitimate transactions far outweight the fraudelent ones

- **Balancing** the class distribution through **resampling** is one of the most popular solutions:
  - **Undersampling**: Put to one side observations from the *majority* class
  - **Oversampling**: Replicate observations from the *minority* class

Machine learning gone mad - The US tank experiment

- 1980s: Pentagon got excited about artificial neural networks
- Wanted a classifier to detect whether a tank is hiding behind trees
- Collected photos of trees with/without tanks hiding in them
- Trained neural network performed *excellently* on the testing dataset
  Champagne to all the scientists!

Machine learning gone mad - The US tank experiment

- 1980s: Pentagon got excited about artificial neural networks
- Wanted a classifier to detect whether a tank is hiding behind trees
- Collected photos of trees with/without tanks hiding in them
- Trained neural network performed *excellently* on the testing dataset
- **Champagne to all the scientists!**

Machine learning gone mad - The US tank experiment

- 1980s: Pentagon got excited about artificial neural networks
- Wanted a classifier to detect whether a tank is hiding behind trees
- Collected photos of trees with/without tanks hiding in them
- Trained neural network performed *excellently* on the testing dataset
  Champagne to all the scientists!

Machine learning gone mad - The US tank experiment

- 1980s: Pentagon got excited about artificial neural networks
- Wanted a classifier to detect whether a tank is hiding behind trees
- Collected photos of trees with/without tanks hiding in them
- Trained neural network performed *excellently* on the testing dataset **Champagne to all the scientists!**

Machine learning gone mad - The US tank experiment

- Another set of tank/no tank photos was commissioned
- The classifier was now useless and no better than randomly guessing
- Someone noted that all previous photos were taken on:
  - no tank: **sunny blue** skies day
  - tank: **cloudy grey** skies day
- Neural network had learnt whether it was a sunny or cloudy day

God bless the United States of America!

Machine learning gone mad - The US tank experiment

- Another set of tank/no tank photos was commissioned
- The classifier was now useless and no better than randomly guessing
- Someone noted that all previous photos were taken on:
  no tank: **sunny blue** skies day
  tank: **cloudy grey** skies day
- Neural network had learnt whether it was a sunny or cloudy day
  God bless the United States of America!

Machine learning gone mad - The US tank experiment

- Another set of tank/no tank photos was commissioned
- The classifier was now useless and no better than randomly guessing
- Someone noted that all previous photos were taken on:

  no tank: **sunny blue** skies day

  tank: **cloudy grey** skies day

- Neural network had learnt whether it was a sunny or cloudy day

God bless the United States of America!

Machine learning gone mad - The US tank experiment

- Another set of tank/no tank photos was commissioned
- The classifier was now useless and no better than randomly guessing
- Someone noted that all previous photos were taken on:

  no tank: **sunny blue** skies day

  tank: **cloudy grey** skies day

- Neural network had learnt whether it was a sunny or cloudy day
  God bless the United States of America!

Machine learning gone mad - Google flu trends

- 2008: Google decided to showcase the power of big data
- Built a predictive model to estimate influenza activity
- Training data were queries containing terms such as cough and fever
- Used IP address to break data by states
- Outcome (label) was influenza-like illness (ILI) physician visits collected by the CDC (Centers for Disease Control and Prevention)
- Model was successful, it predicted a spike in the mid-Atlantic region of the US two weeks prior to the CDC
  Google is great!

Machine learning gone mad - Google flu trends

- 2008: Google decided to showcase the power of big data
- Built a predictive model to estimate influenza activity
- Training data were queries containing terms such as cough and fever
- Used IP address to break data by states
- Outcome (label) was influenza-like illness (ILI) physician visits collected by the CDC (Centers for Disease Control and Prevention)
- Model was successful, it predicted a spike in the mid-Atlantic region of the US two weeks prior to the CDC
  Google is great!

Machine learning gone mad - Google flu trends

- 2008: Google decided to showcase the power of big data
- Built a predictive model to estimate influenza activity
- Training data were queries containing terms such as cough and fever
- Used IP address to break data by states
- Outcome (label) was influenza-like illness (ILI) physician visits collected by the CDC (Centers for Disease Control and Prevention)
- Model was successful, it predicted a spike in the mid-Atlantic region of the US two weeks prior to the CDC

  Google is great!

Machine learning gone mad - Google flu trends

- 2008: Google decided to showcase the power of big data
- Built a predictive model to estimate influenza activity
- Training data were queries containing terms such as cough and fever
- Used IP address to break data by states
- Outcome (label) was influenza-like illness (ILI) physician visits collected by the CDC (Centers for Disease Control and Prevention)
- Model was successful, it predicted a spike in the mid-Atlantic region of the US two weeks prior to the CDC

  Google is great!

Machine learning gone mad - Google flu trends

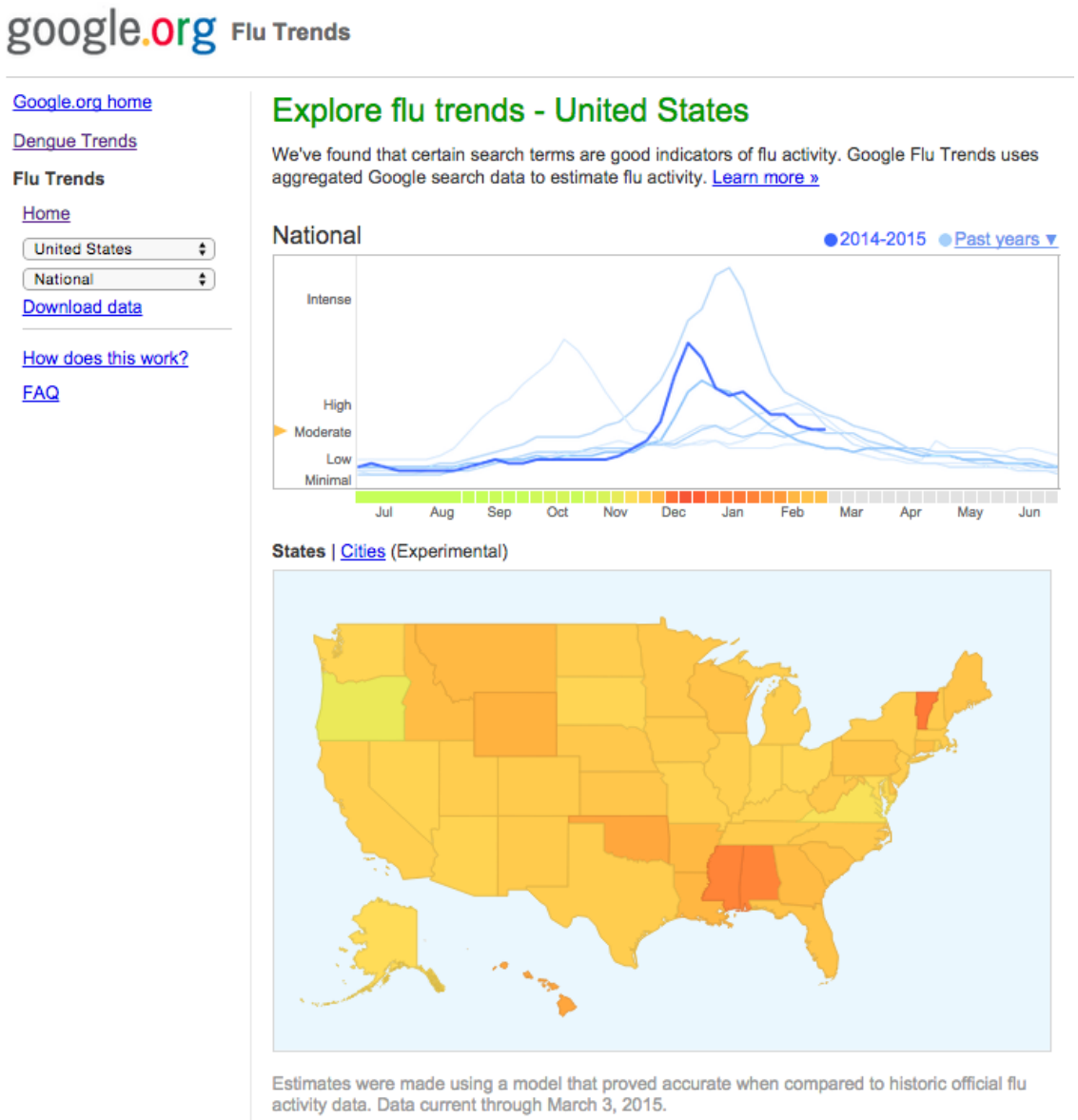
- 2008: Google decided to showcase the power of big data
- Built a predictive model to estimate influenza activity
- Training data were queries containing terms such as cough and fever
- Used IP address to break data by states
- Outcome (label) was influenza-like illness (ILI) physician visits collected by the CDC (Centers for Disease Control and Prevention)
- Model was successful, it predicted a spike in the mid-Atlantic region of the US two weeks prior to the CDC

  Google is great!

Machine learning gone mad - Google flu trends

- 2008: Google decided to showcase the power of big data
- Built a predictive model to estimate influenza activity
- Training data were queries containing terms such as cough and fever
- Used IP address to break data by states
- Outcome (label) was influenza-like illness (ILI) physician visits collected by the CDC (Centers for Disease Control and Prevention)
- Model was successful, it predicted a spike in the mid-Atlantic region of the US two weeks prior to the CDC
  Google is great!

Machine learning gone mad - Google flu trends



FINAL FINAL

POLICYFORUM

BIG DATA

The Parable of Google Flu: Traps in Big Data Analysis

Large errors in flu prediction were largely avoidable, which offers lessons for the use of big data.

David Lazer,[1,2*] Ryan Kennedy,[1,3,4] Gary King,[3] Alessandro Vespignani[3,5,6]

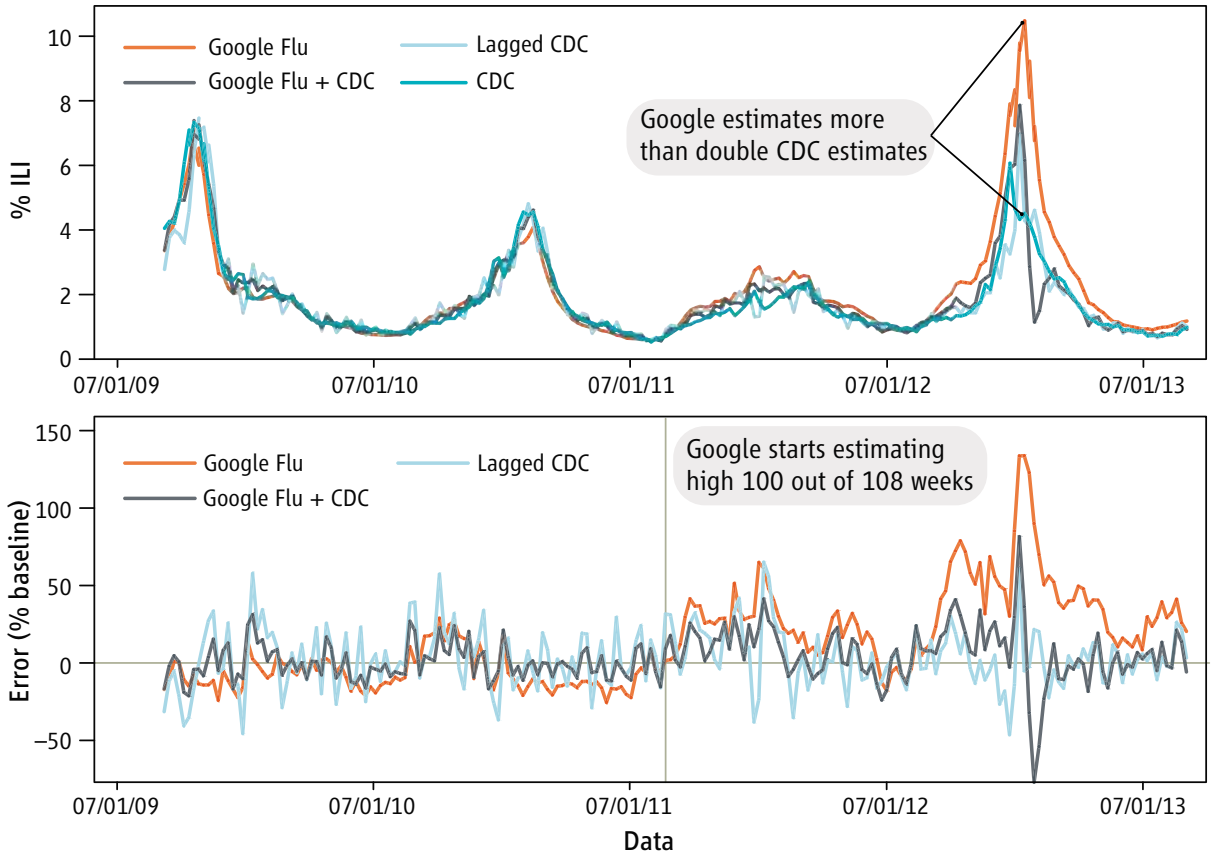
In February 2013, Google Flu Trends (GFT) made headlines but not for a reason that Google executives or the creators of the flu tracking system would have hoped. *Nature* reported that GFT was predicting more than double the proportion of doctor visits for influenza-like illness (ILI) than the Centers for Disease Control and Prevention (CDC), which bases its estimates on surveillance reports from laboratories across the United States (*1*, *2*). This happened despite the fact that GFT was built to predict CDC reports. Given that GFT is often held up as an exemplary use of big data (*3*, *4*), what lessons can we draw from this error?

The problems we identify are not limited to GFT. Research on whether search or social media can predict *x* has become common-

the algorithm in 2009, and this model has run ever since, with a few changes announced in October 2013 (*10*, *15*).

Although not widely reported until 2013, the new GFT has been persistently overestimating flu prevalence for a much longer time. GFT also missed by a very large margin in the 2011–2012 flu season and has missed high for 100 out of 108 weeks starting with August 2011 (see the graph). These errors are not randomly distributed. For example, last week's errors predict this week's errors (temporal autocorrelation), and the direction and magnitude of error varies with the time of year (seasonality). These patterns mean that GFT overlooks considerable information that could be extracted by traditional statistical methods.

Machine learning gone mad - Google flu trends



GFT overestimation. GFT overestimated the prevalence of flu in the 2012–2013 season and overshot the actual level in 2011–2012 by more than 50%. From 21 August 2011 to 1 September 2013, GFT reported overly high flu prevalence 100 out of 108 weeks. (**Top**) Estimates of doctor visits for ILI. "Lagged CDC" incorporates 52-week seasonality variables with lagged CDC data. "Google Flu + CDC" combines GFT, lagged CDC estimates, lagged error of GFT estimates, and 52-week seasonality variables. (**Bottom**) Error [as a percentage {[Non-CDC estmate)−(CDC estimate)]/(CDC) estimate)}. Both alternative models have much less error than GFT alone. Mean absolute error (MAE) during the out-of-sample period is 0.486 for GFT, 0.311 for lagged CDC, and 0.232 for combined GFT and CDC. All of these differences are statistically significant at $P < 0.05$. See SM.

Google flu trends, so what went wrong?

- Google had to mine 50 million search terms that best correlated with 1,152 "true" data points collected by CDC (Centers for Disease Control and Prevention). They retained 45 queries
- Winter is coming: Correlation is not causation
- Are these correlates stable and comparable over time?
- Google's search algorithm changes very frequently. Autosuggest feature might have implicitly caused more people to search flu-related terms
- Is it time to recalibrate the model and/or hybridise both data sources?
- In 2014, Google retrained and updated the model significantly, but cancelled the project in 2015

Google flu trends, so what went wrong?

- Google had to mine 50 million search terms that best correlated with 1,152 "true" data points collected by CDC (Centers for Disease Control and Prevention). They retained 45 queries
- Winter is coming: Correlation is not causation
- Are these correlates stable and comparable over time?
- Google's search algorithm changes very frequently. Autosuggest feature might have implicitly caused more people to search flu-related terms
- Is it time to recalibrate the model and/or hybridise both data sources?
- In 2014, Google retrained and updated the model significantly, but cancelled the project in 2015

Google flu trends, so what went wrong?

- Google had to mine 50 million search terms that best correlated with 1,152 "true" data points collected by CDC (Centers for Disease Control and Prevention). They retained 45 queries
- Winter is coming: Correlation is not causation
- Are these correlates stable and comparable over time?
- Google's search algorithm changes very frequently. Autosuggest feature might have implicitly caused more people to search flu-related terms
- Is it time to recalibrate the model and/or hybridise both data sources?
- In 2014, Google retrained and updated the model significantly, but cancelled the project in 2015

Google flu trends, so what went wrong?

- Google had to mine 50 million search terms that best correlated with 1,152 "true" data points collected by CDC (Centers for Disease Control and Prevention). They retained 45 queries
- Winter is coming: Correlation is not causation
- Are these correlates stable and comparable over time?
- Google's search algorithm changes very frequently. Autosuggest feature might have implicitly caused more people to search flu-related terms
- Is it time to recalibrate the model and/or hybridise both data sources?
- In 2014, Google retrained and updated the model significantly, but cancelled the project in 2015

Google flu trends, so what went wrong?

- Google had to mine 50 million search terms that best correlated with 1,152 "true" data points collected by CDC (Centers for Disease Control and Prevention). They retained 45 queries
- Winter is coming: Correlation is not causation
- Are these correlates stable and comparable over time?
- Google's search algorithm changes very frequently. Autosuggest feature might have implicitly caused more people to search flu-related terms
- Is it time to recalibrate the model and/or hybridise both data sources?
- In 2014, Google retrained and updated the model significantly, but cancelled the project in 2015

Google flu trends, so what went wrong?

- Google had to mine 50 million search terms that best correlated with 1,152 "true" data points collected by CDC (Centers for Disease Control and Prevention). They retained 45 queries
- Winter is coming: Correlation is not causation
- Are these correlates stable and comparable over time?
- Google's search algorithm changes very frequently. Autosuggest feature might have implicitly caused more people to search flu-related terms
- Is it time to recalibrate the model and/or hybridise both data sources?
- In 2014, Google retrained and updated the model significantly, but cancelled the project in 2015

Laws of data analysis

1. Shite in, shite out
   Anonymous

2. If you torture the data long enough it will confess to anything
   Ronald Coase (1910 - 2013)

3. A sufficiently elaborate analysis process can always lend an air of legitimacy
   Chris Laws, ex-McLaren boss

Laws of data analysis

1. Shite in, shite out
   Anonymous
2. If you torture the data long enough it will confess to anything
   Ronald Coase (1910 - 2013)
3. A sufficiently elaborate analysis process can always lend an air of legitimacy
   Chris Laws, ex-McLaren boss

Laws of data analysis

1. Shite in, shite out
   Anonymous

2. If you torture the data long enough it will confess to anything
   Ronald Coase (1910 - 2013)

3. A sufficiently elaborate analysis process can always lend an air of legitimacy
   Chris Laws, ex-McLaren boss

Laws of data analysis

1. Shite in, shite out
   Anonymous

2. If you torture the data long enough it will confess to anything
   Ronald Coase (1910 - 2013)

3. A sufficiently elaborate analysis process can always lend an air of legitimacy
   Chris Laws, ex-McLaren boss

A few important tips

- **Data exploration** is key to understanding the data's structure. Use dimensionality reduction techniques to visualise high-dimensional data
- Features with **little variability** can be safely ignored
- **Impute**/ignore missing data (are they missing completely at random?)
- **Standardise**/**normalise** features with widely varying ranges
- **Features** extracted from the data need to be directly relevant to the question you're asking
- Use **expert application knowledge** where possible over automatic feature extraction
- **Do not** trust predictions for inputs outside the training dataset range (i.e avoid extrapolation)

A few important tips

- **Data exploration** is key to understanding the data's structure. Use dimensionality reduction techniques to visualise high-dimensional data
- Features with **little variability** can be safely ignored
- **Impute**/ignore missing data (are they missing completely at random?)
- **Standardise**/**normalise** features with widely varying ranges
- **Features** extracted from the data need to be directly relevant to the question you're asking
- Use **expert application knowledge** where possible over automatic feature extraction
- **Do not** trust predictions for inputs outside the training dataset range (i.e avoid extrapolation)

A few important tips

- **Data exploration** is key to understanding the data's structure. Use dimensionality reduction techniques to visualise high-dimensional data
- Features with **little variability** can be safely ignored
- **Impute**/ignore missing data (are they missing completely at random?)
- **Standardise**/**normalise** features with widely varying ranges
- **Features** extracted from the data need to be directly relevant to the question you're asking
- Use **expert application knowledge** where possible over automatic feature extraction
- **Do not** trust predictions for inputs outside the training dataset range (i.e avoid extrapolation)

A few important tips

- **Data exploration** is key to understanding the data's structure. Use dimensionality reduction techniques to visualise high-dimensional data
- Features with **little variability** can be safely ignored
- **Impute**/ignore missing data (are they missing completely at random?)
- **Standardise**/**normalise** features with widely varying ranges
- **Features** extracted from the data need to be directly relevant to the question you're asking
- Use **expert application knowledge** where possible over automatic feature extraction
- **Do not** trust predictions for inputs outside the training dataset range (i.e avoid extrapolation)

A few important tips

- **Data exploration** is key to understanding the data's structure. Use dimensionality reduction techniques to visualise high-dimensional data
- Features with **little variability** can be safely ignored
- **Impute**/ignore missing data (are they missing completely at random?)
- **Standardise**/**normalise** features with widely varying ranges
- **Features** extracted from the data need to be directly relevant to the question you're asking
- Use **expert application knowledge** where possible over automatic feature extraction
- **Do not** trust predictions for inputs outside the training dataset range (i.e avoid extrapolation)

A few important tips

- **Data exploration** is key to understanding the data's structure. Use dimensionality reduction techniques to visualise high-dimensional data
- Features with **little variability** can be safely ignored
- **Impute**/ignore missing data (are they missing completely at random?)
- **Standardise**/**normalise** features with widely varying ranges
- **Features** extracted from the data need to be directly relevant to the question you're asking
- Use **expert application knowledge** where possible over automatic feature extraction
- **Do not** trust predictions for inputs outside the training dataset range (i.e avoid extrapolation)

A few important tips

- **Data exploration** is key to understanding the data's structure. Use dimensionality reduction techniques to visualise high-dimensional data
- Features with **little variability** can be safely ignored
- **Impute**/ignore missing data (are they missing completely at random?)
- **Standardise**/**normalise** features with widely varying ranges
- **Features** extracted from the data need to be directly relevant to the question you're asking
- Use **expert application knowledge** where possible over automatic feature extraction
- **Do not** trust predictions for inputs outside the training dataset range (i.e avoid extrapolation)

A few important tips

- **Data exploration** is key to understanding the data's structure. Use dimensionality reduction techniques to visualise high-dimensional data
- Features with **little variability** can be safely ignored
- **Impute**/ignore missing data (are they missing completely at random?)
- **Standardise**/**normalise** features with widely varying ranges
- **Features** extracted from the data need to be directly relevant to the question you're asking
- Use **expert application knowledge** where possible over automatic feature extraction
- **Do not** trust predictions for inputs outside the training dataset range (i.e avoid extrapolation)

Practical advice

- Your fitted predictive model has poor accuracy on the testing data What should you do next?

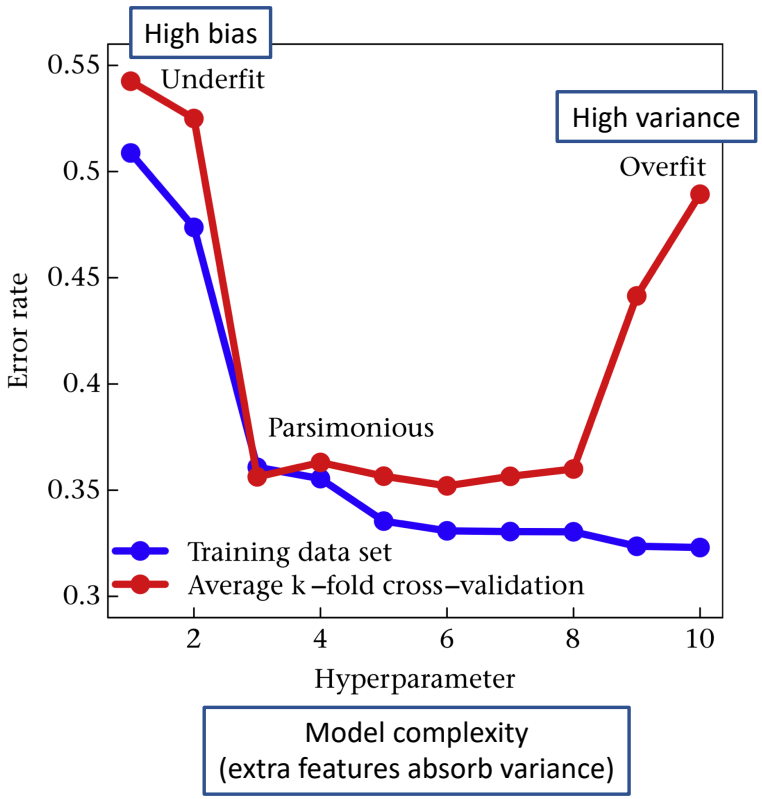
1. **High bias/underfit**
   Training *and* validation errors are large

2. **High variance/overfit**
   Validation error $\gg$ Training error

Practical advice

- Your fitted predictive model has poor accuracy on the testing data What should you do next?

Recall the bias-variance tradeoff:

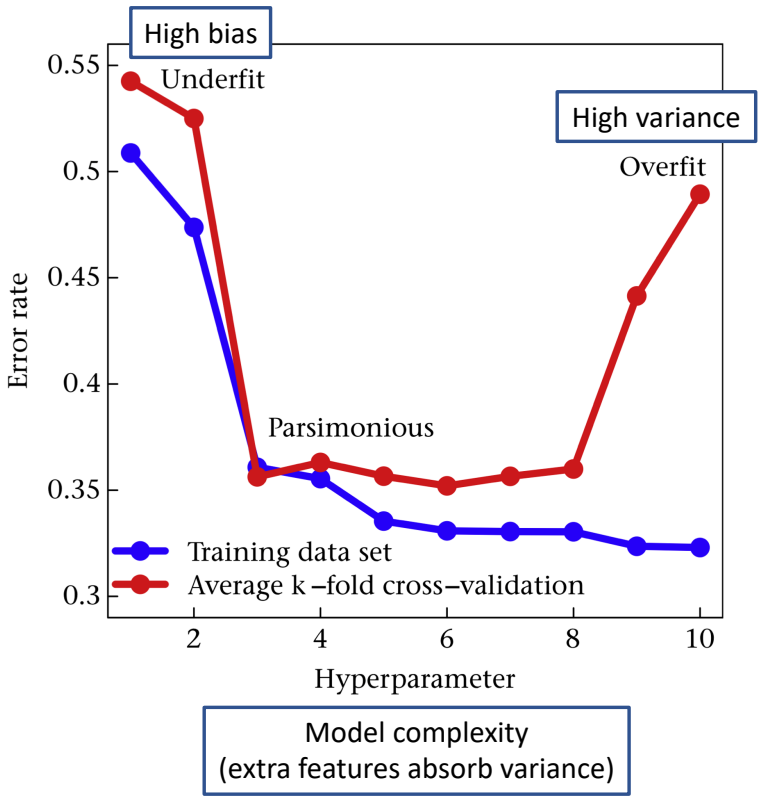


1. **High bias/underfit**
   Training *and* validation errors are large
2. **High variance/overfit**
   Validation error $\gg$ Training error

Practical advice

- Your fitted predictive model has poor accuracy on the testing data
  What should you do next?

Recall the bias-variance tradeoff:



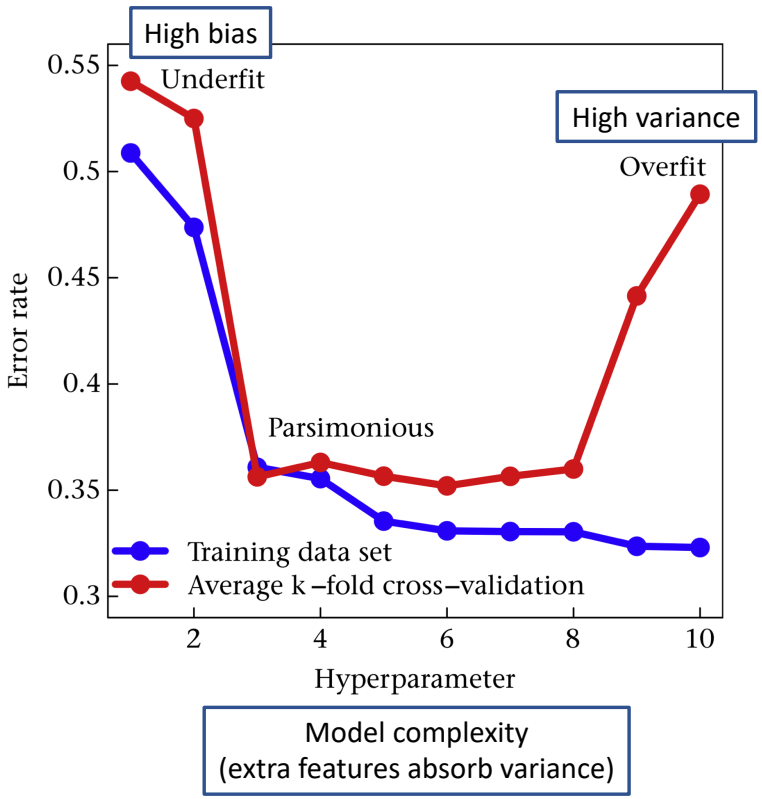
1. **High bias/underfit**
   Training *and* validation errors are large

2. **High variance/overfit**
   Validation error $\gg$ Training error

Practical advice

- Your fitted predictive model has poor accuracy on the testing data
  What should you do next?

Recall the bias-variance tradeoff:

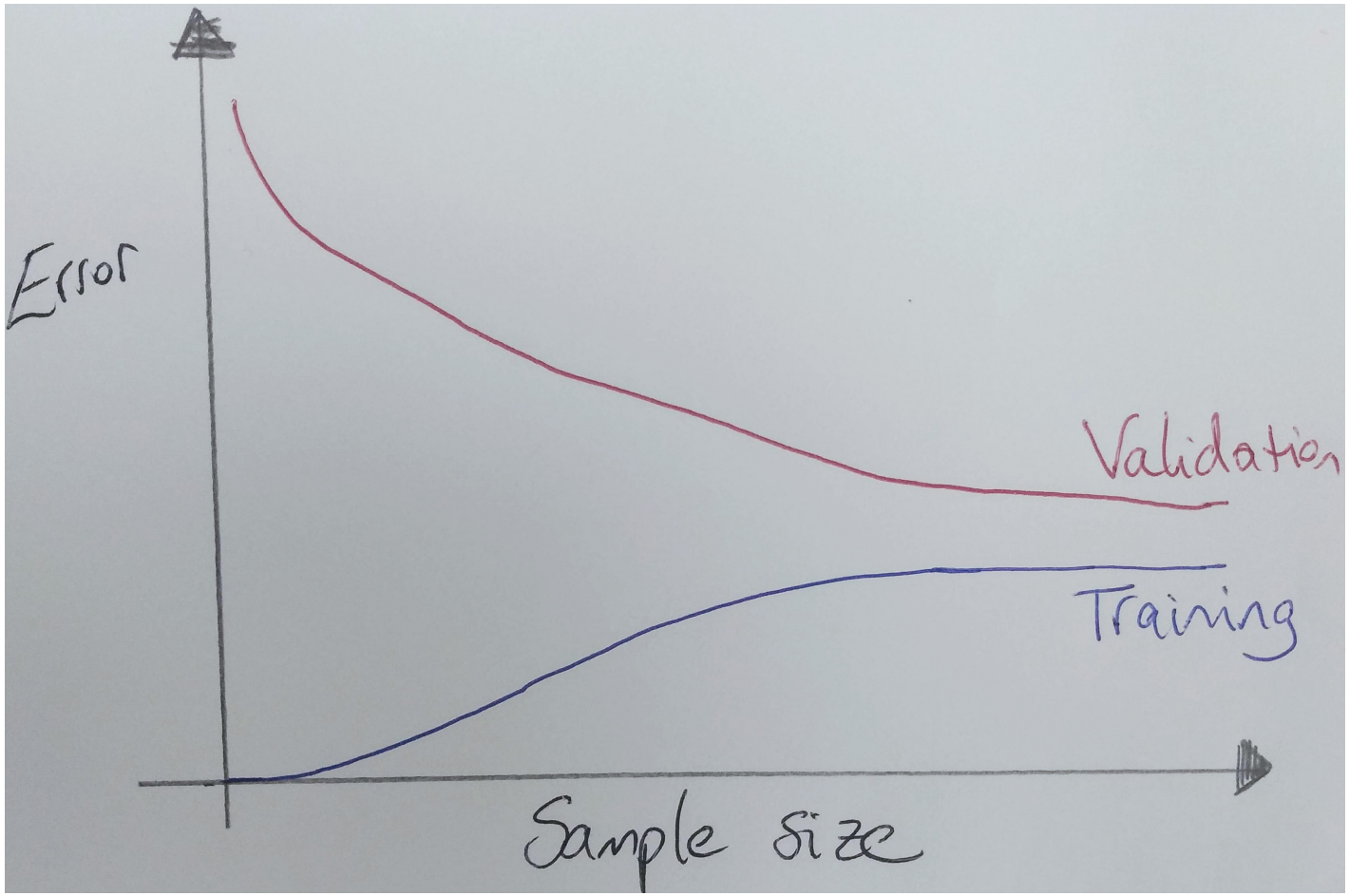


1. **High bias/underfit**
   Training *and* validation errors are large

2. **High variance/overfit**
   Validation error $\gg$ Training error

Learning curves

- **Learning curves** are useful diagnostic plots for learning algorithms
- It's a plot of model performance/error rate vs experience/sample size



High variance

- Collect more data
- Try a smaller set of "hand-crafted" features

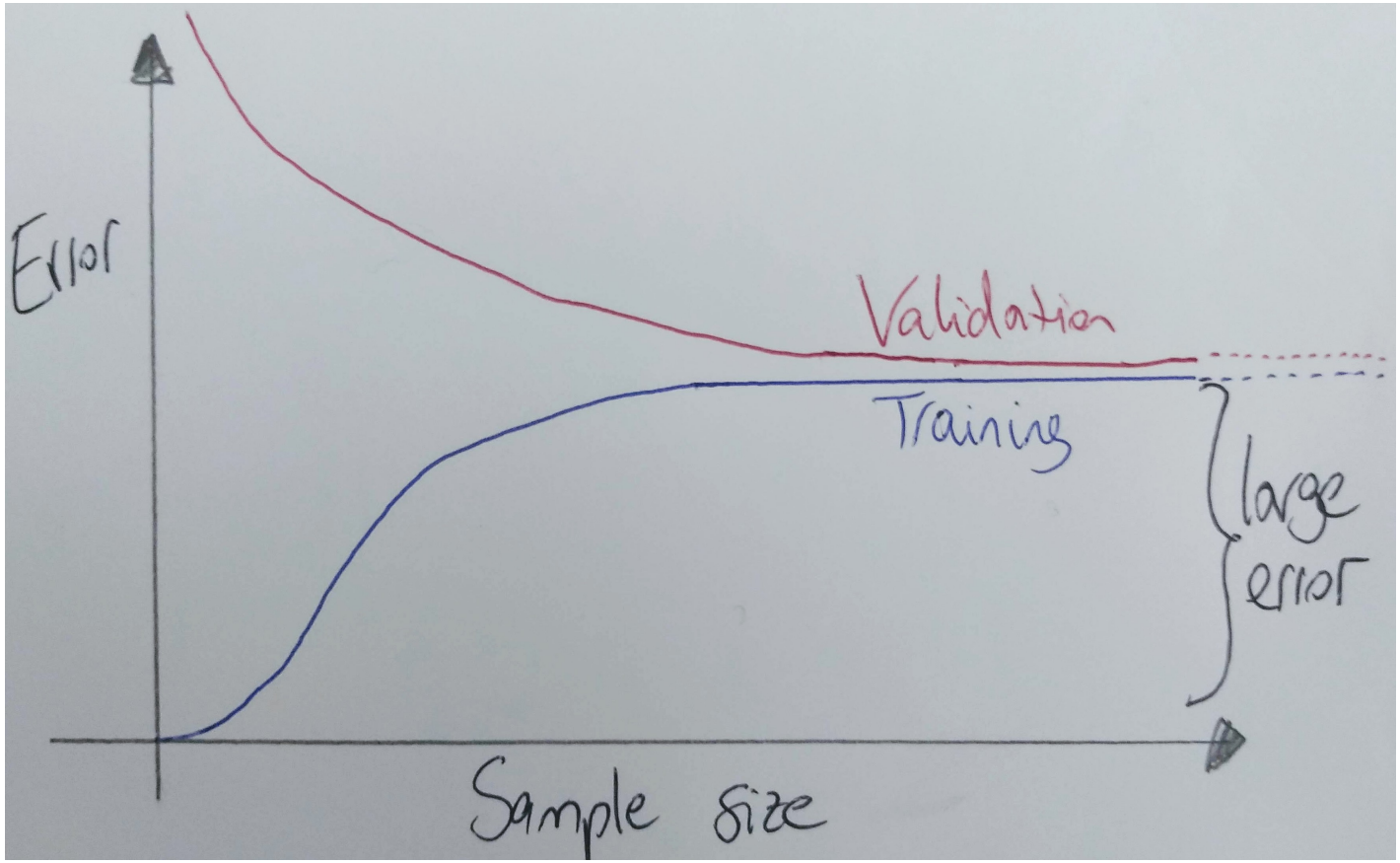
High variance

- Collect more data
- Try a smaller set of "hand-crafted" features

High variance

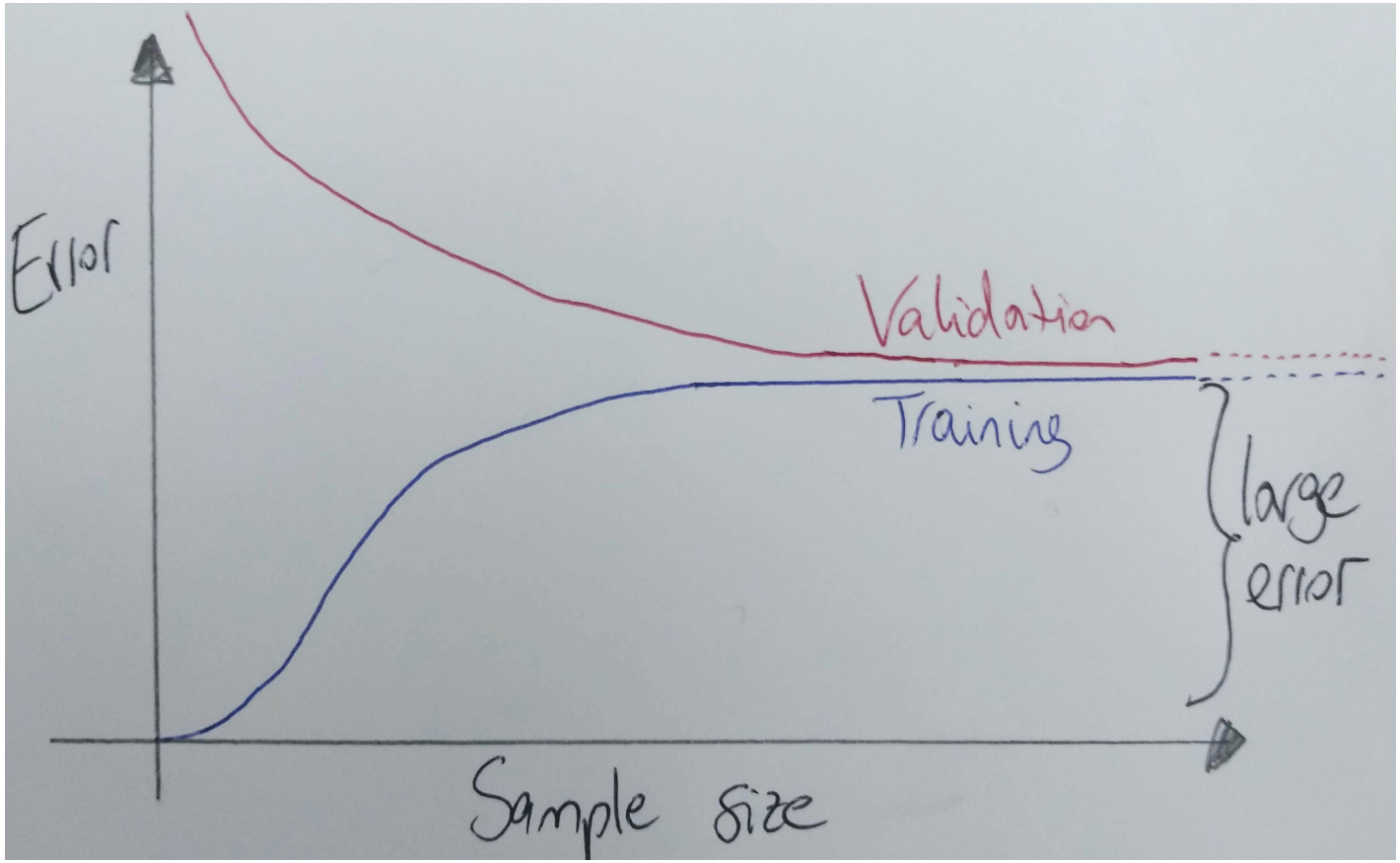
- Collect more data
- Try a smaller set of "hand-crafted" features

High bias



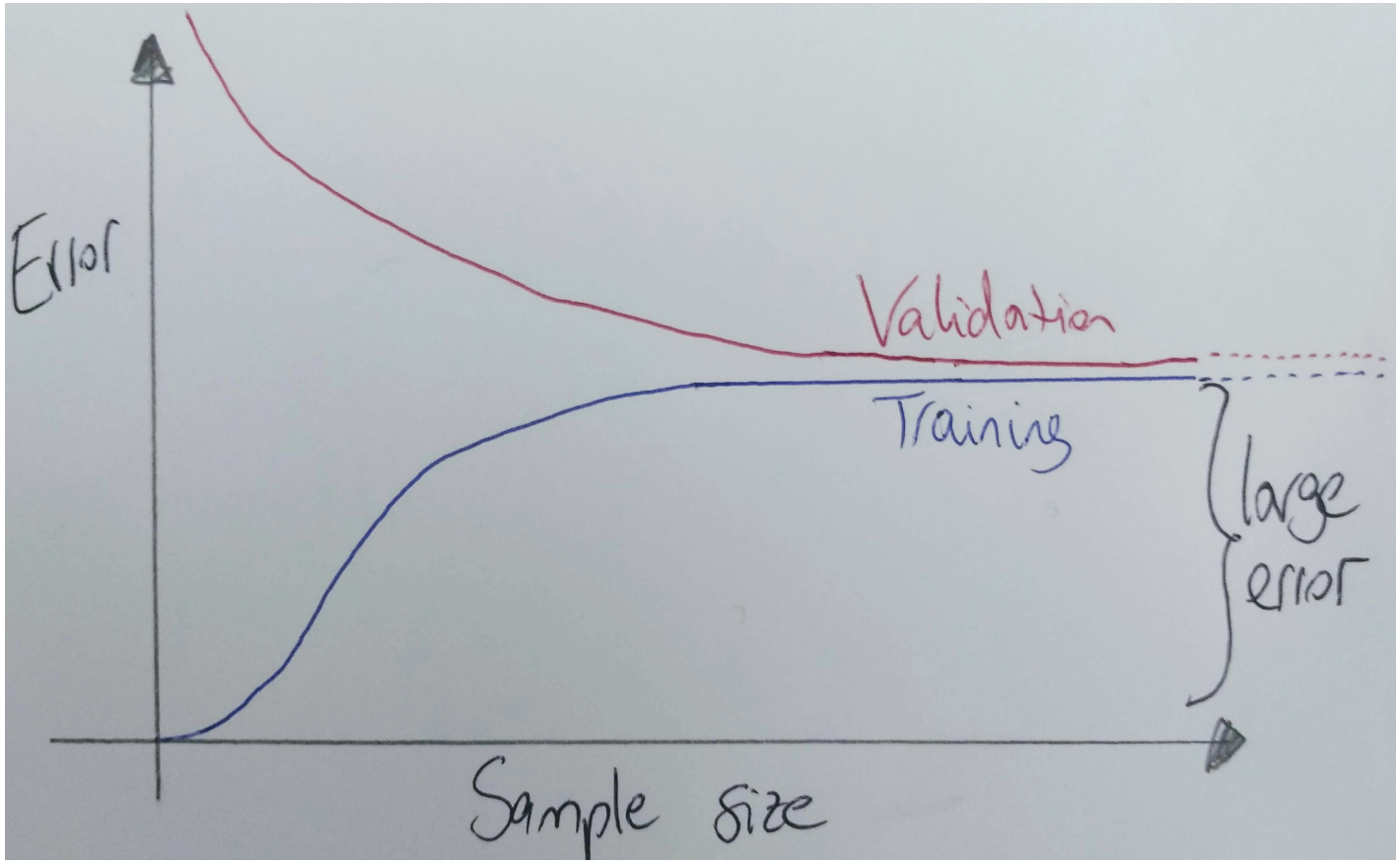
- Try increasing model complexity
- Try additional features
- Add polynomial/splines of features

High bias



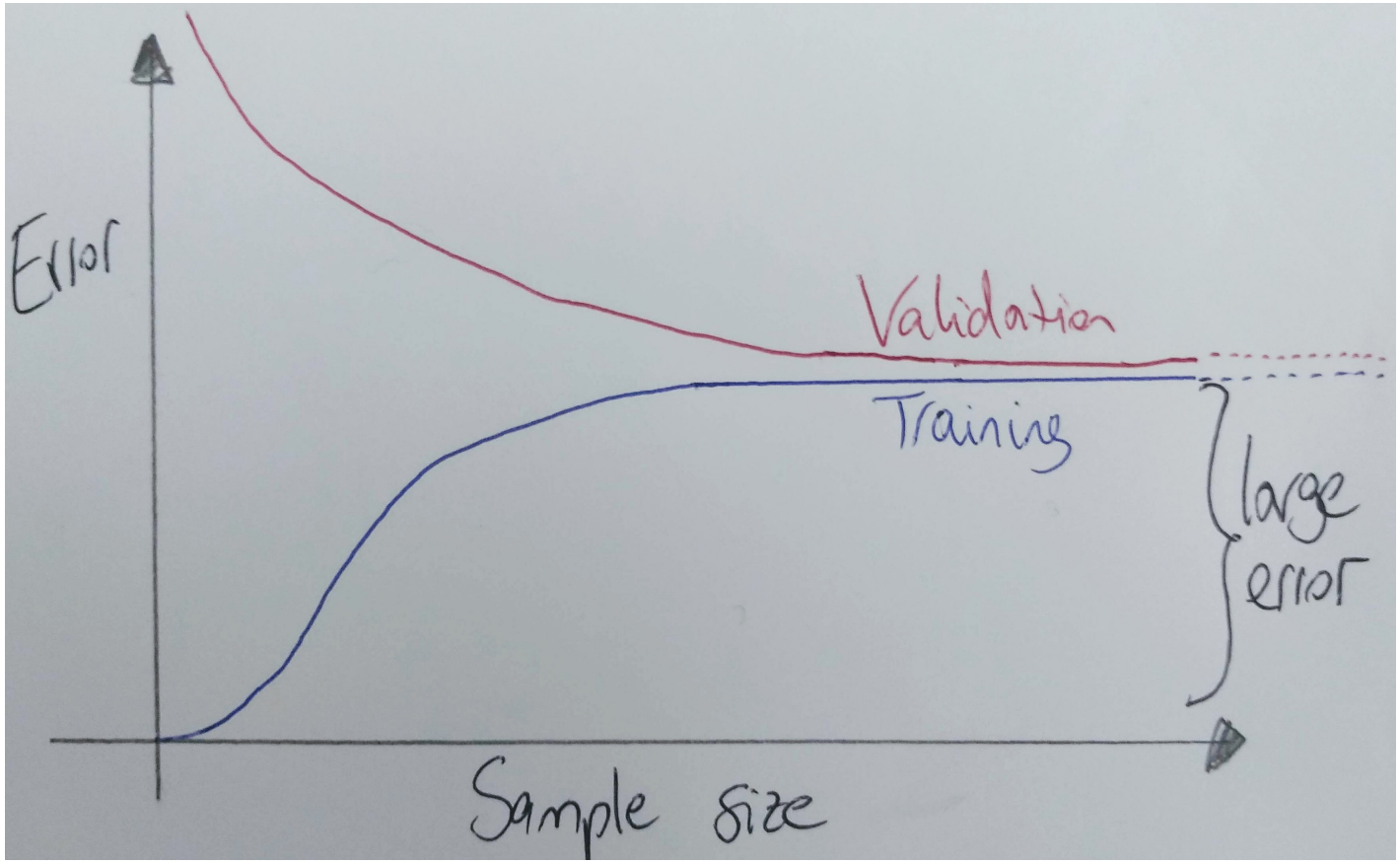
- Try increasing model complexity
- Try additional features
- Add polynomial/splines of features

High bias



- Try increasing model complexity
- Try additional features
- Add polynomial/splines of features

High bias



- Try increasing model complexity
- Try additional features
- Add polynomial/splines of features

Which machine learning algorithm should I use?

- A personal choice/what you're comfortable using rather than some rules set in stone
- Always start with simple models before using more complex ones
- Some methods are more appropriate than others in certain domains:

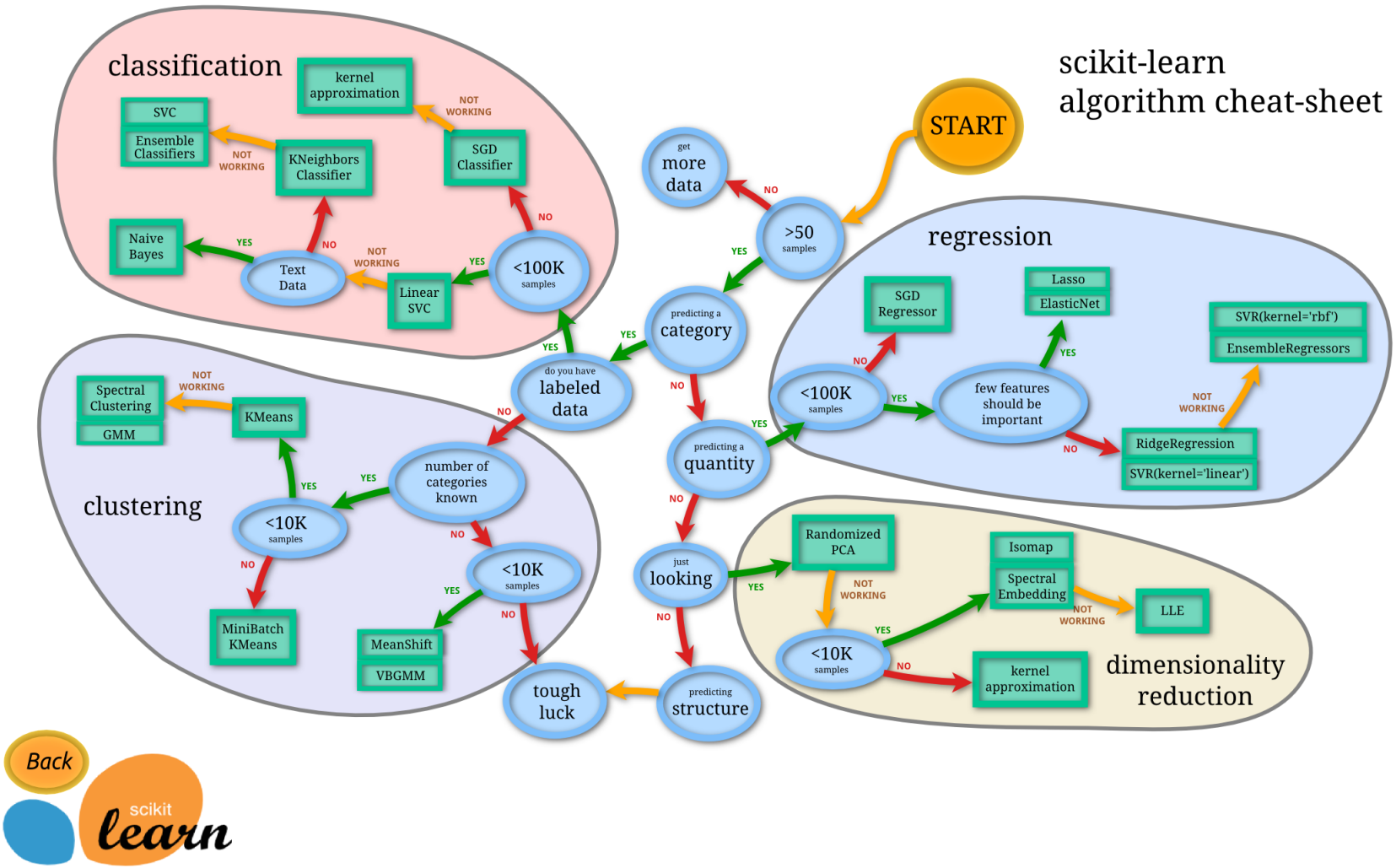
Interpretability: Decision trees or association rule learning

Lots of independent features and data: Naïve bayes

Knowledge of dependencies between features: Bayesian network

Thousands of mixed categorical and continuous variables: Random forests

Which machine learning algorithm should I use?



Source: scikit-learn.org