Reducing Dimensions in Data with scikit-learn

GETTING STARTED WITH FEATURE SELECTION IN SCIKIT-LEARN

Janani Ravi
CO-FOUNDER, LOONYCORN
www.loonycorn.com

Overview

Need for dimensionality reduction in building ML models

Methods for selecting and eliminating features

Feature selection using statistical techniques

Dictionary learning and atom extraction

Prerequisites and Course Outline

Prerequisites

Working with Python and Python libraries

Basic understanding of machine learning algorithms

Prerequisites

Understanding Machine Learning by David Chappell

Building Machine Learning Models in Python with scikit-learn by Janani Ravi

Understanding Machine Learning with Python by Jerry Kurata

Course Outline

Feature selection

- Statistical techniques for feature selection
- Dictionary learning for sparse representations of complex data

Dimensionality reduction in linear data

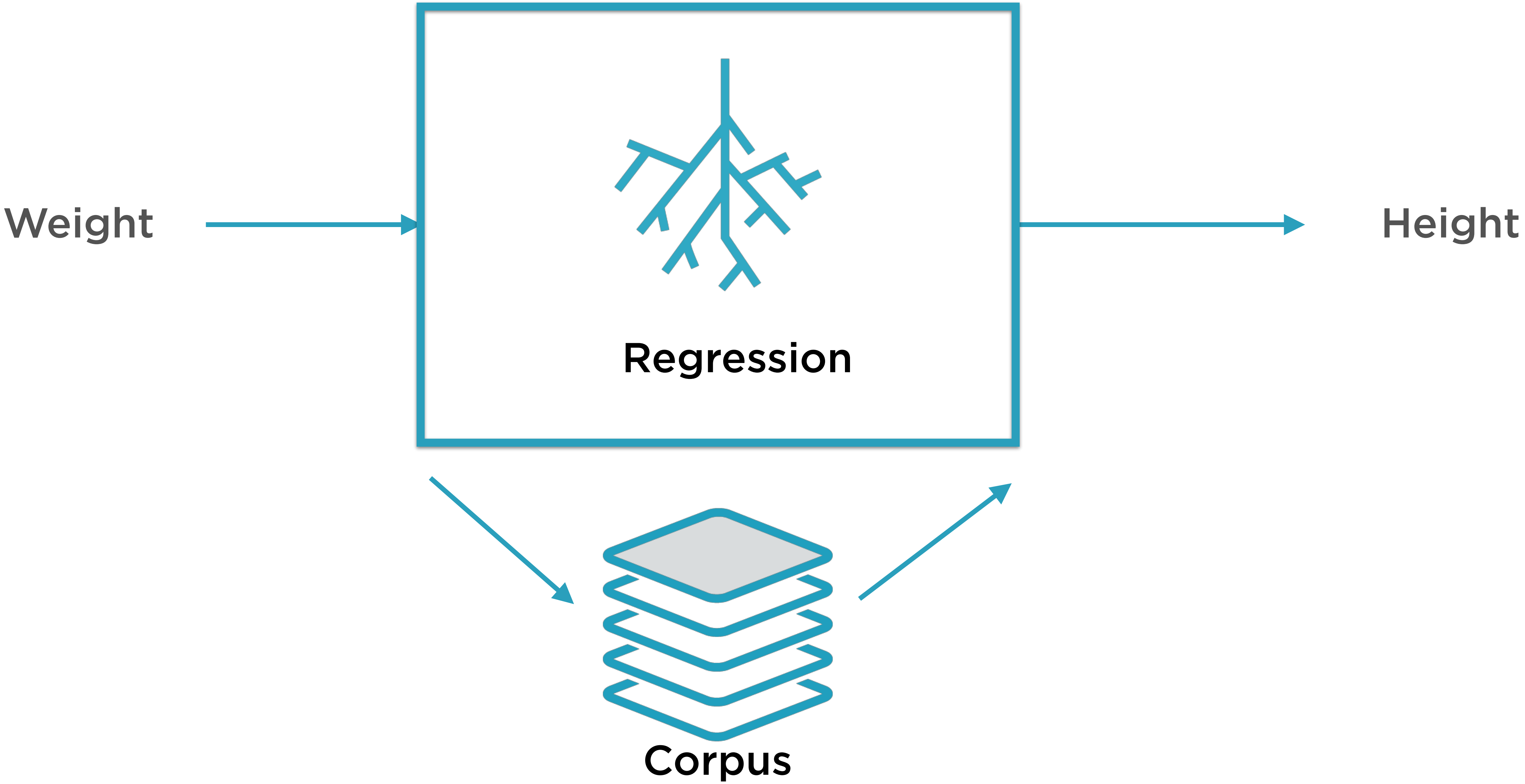
- Principal Components Analysis, Factor Analysis and Linear Discriminant Analysis

Dimensionality reduction in non-linear data

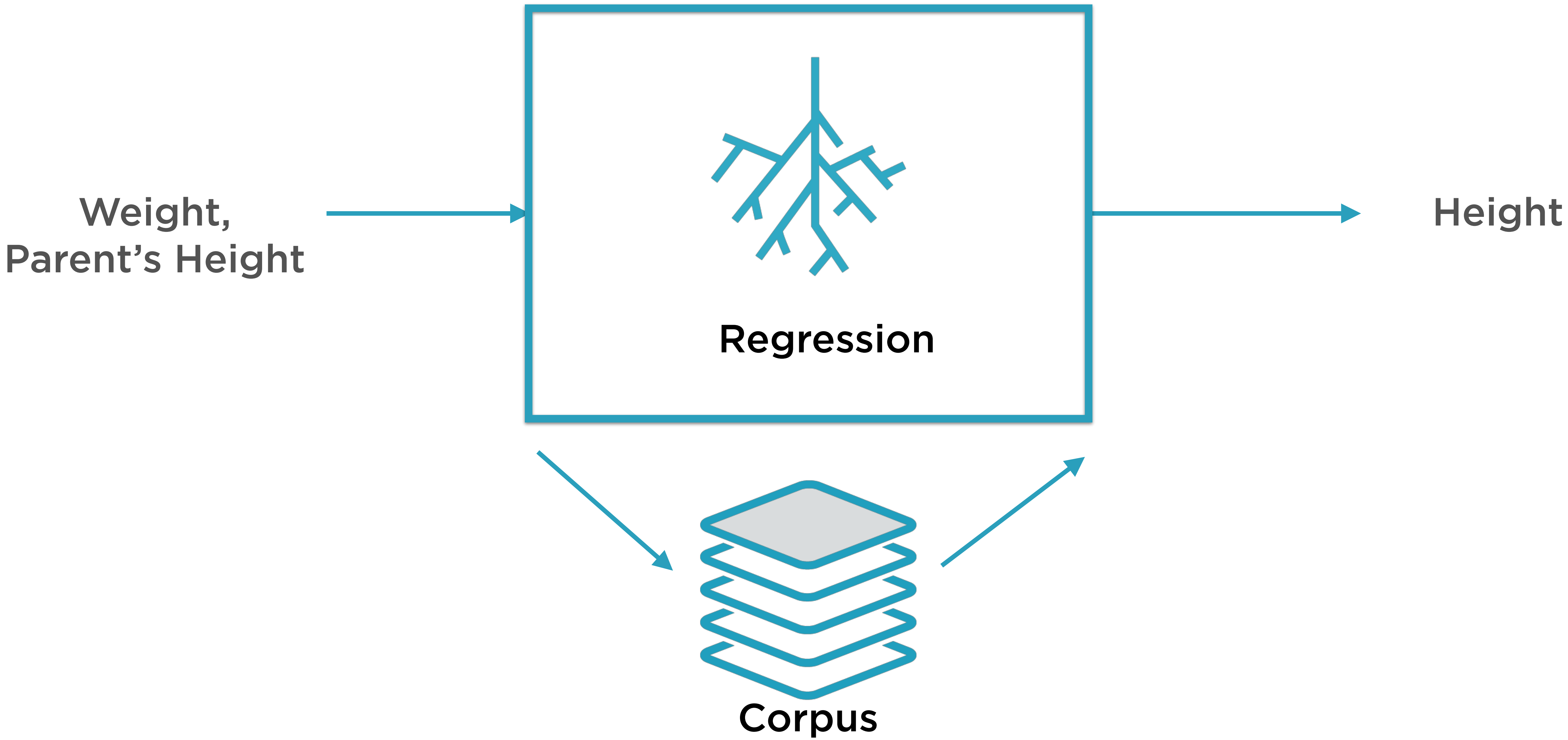
- Manifold learning techniques
- Applying manifold learning to images

The Curse of Dimensionality

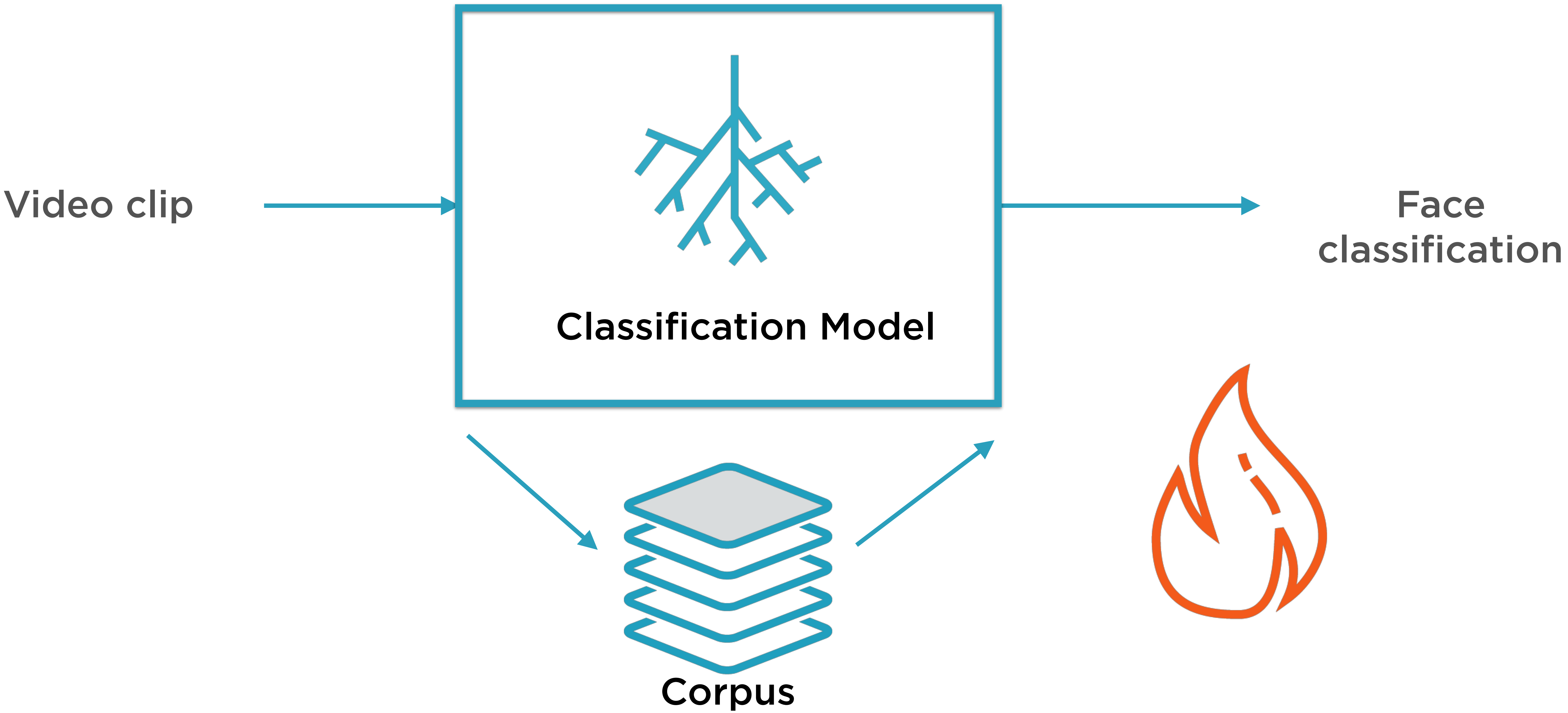
One X Variable



Two X Variables



Dimensionality Explosion



Video clip

Classification Model

Corpus

Face classification

Curse of Dimensionality: As number of **x** variables grows, several problems arise

Curse of Dimensionality

Problems in Visualization

Problems in Training

Problems in Prediction

Curse of Dimensionality

Problems in Visualization

Problems in Training

Problems in Prediction

Problems in Visualization

Exploratory Data Analysis (EDA) is an essential precursor to model building

Essential for

- identifying outliers
- detecting anomalies
- choosing functional form of relationships

Problems in Visualization

Two dimensional visualizations are powerful aids in EDA

Even three-dimensional data is hard to meaningfully visualize

Higher dimensional data is often imperfectly explored prior to ML

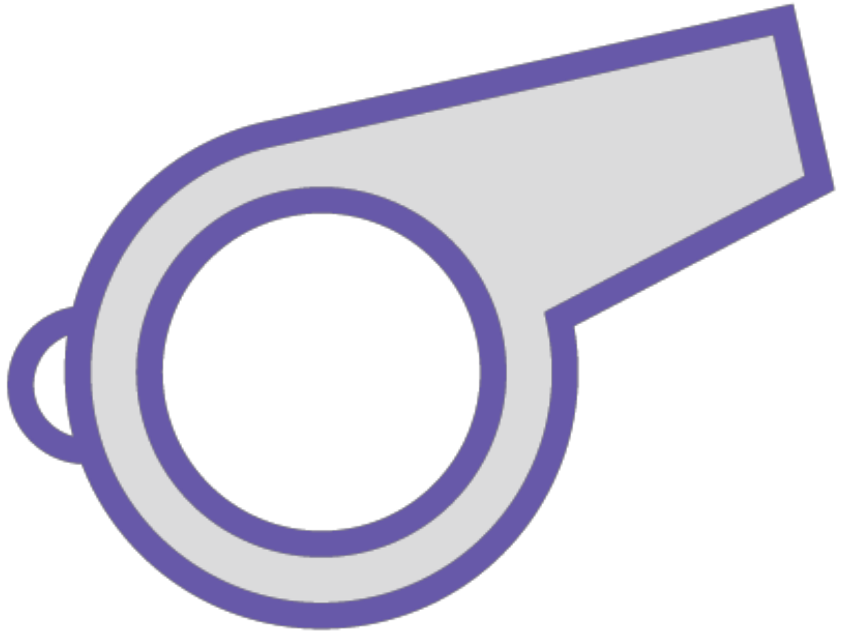
Curse of Dimensionality

Problems in Visualization

Problems in Training

Problems in Prediction

Problems in Training

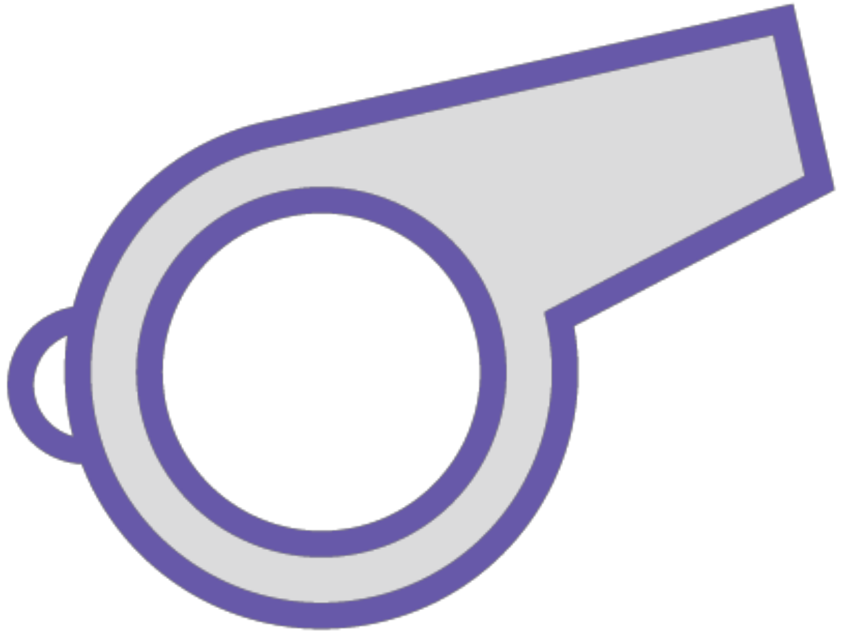


Training is the process of finding best model parameters

Complex models have thousands of parameter values

Training for too little time leads to bad models

Problems in Training



Number of parameters to be found grows rapidly with dimensionality

Extremely time-consuming

For on-cloud training, also extremely expensive

Curse of Dimensionality

Problems in Visualization

Problems in Training

Problems in Prediction

Problems in Prediction

Prediction involves finding training instances similar to test instance

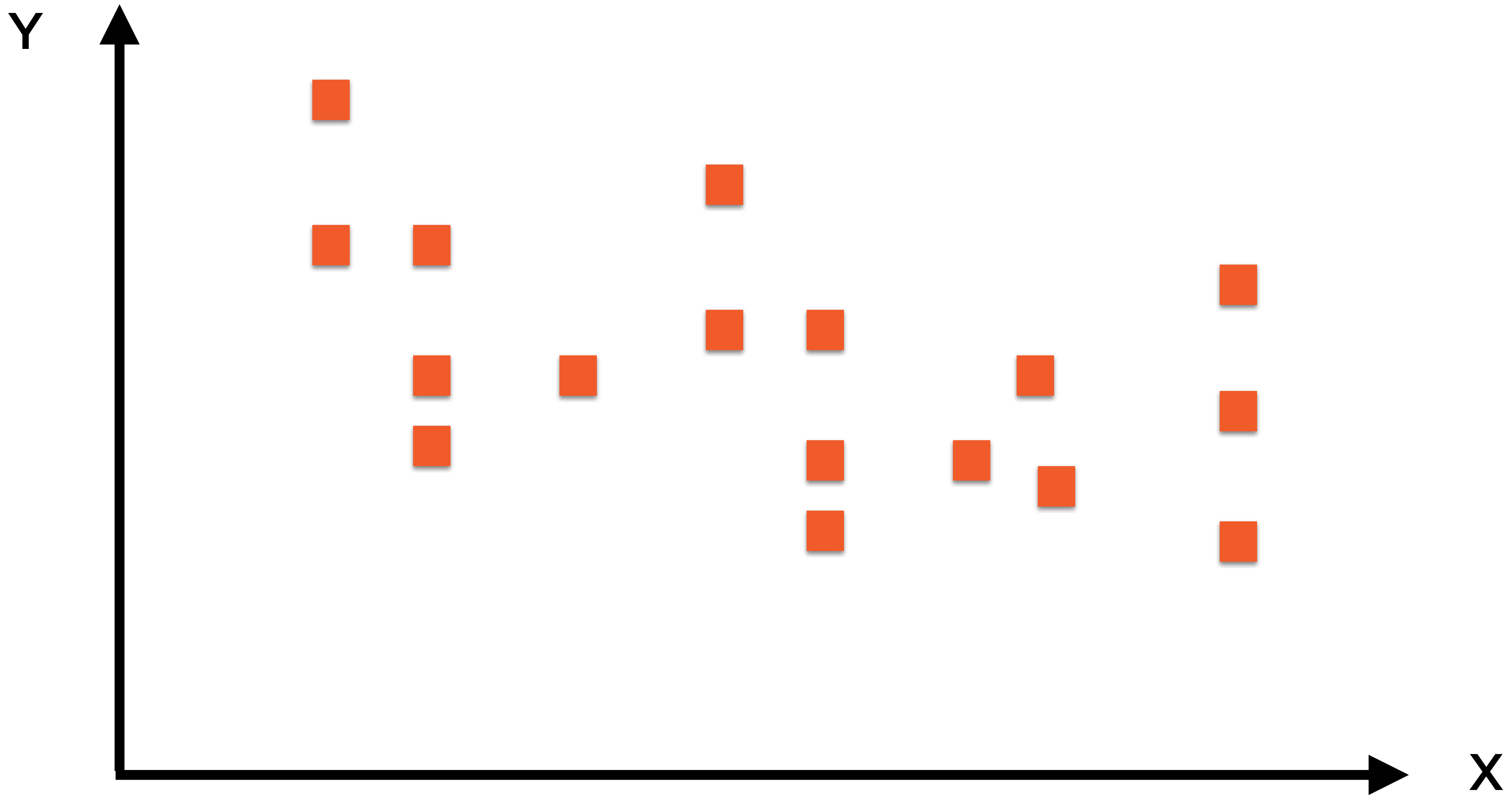
As dimensionality grows, size of search space explodes

Higher the number of X variables, higher the risk of overfitting

Overfitted Models and Data Sparsity

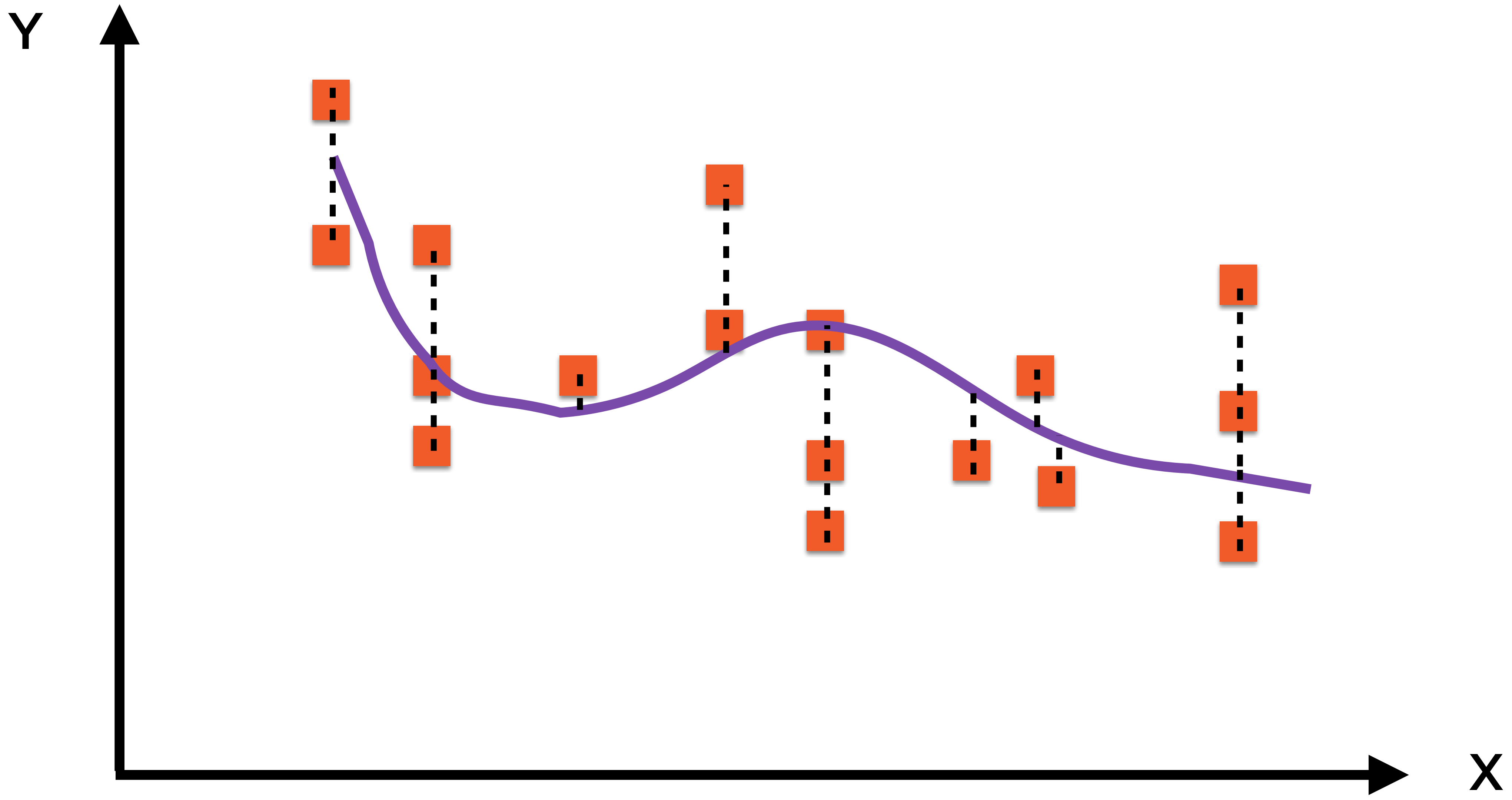
Using a large number of features in training can result in overfitted models

Connecting the Dots



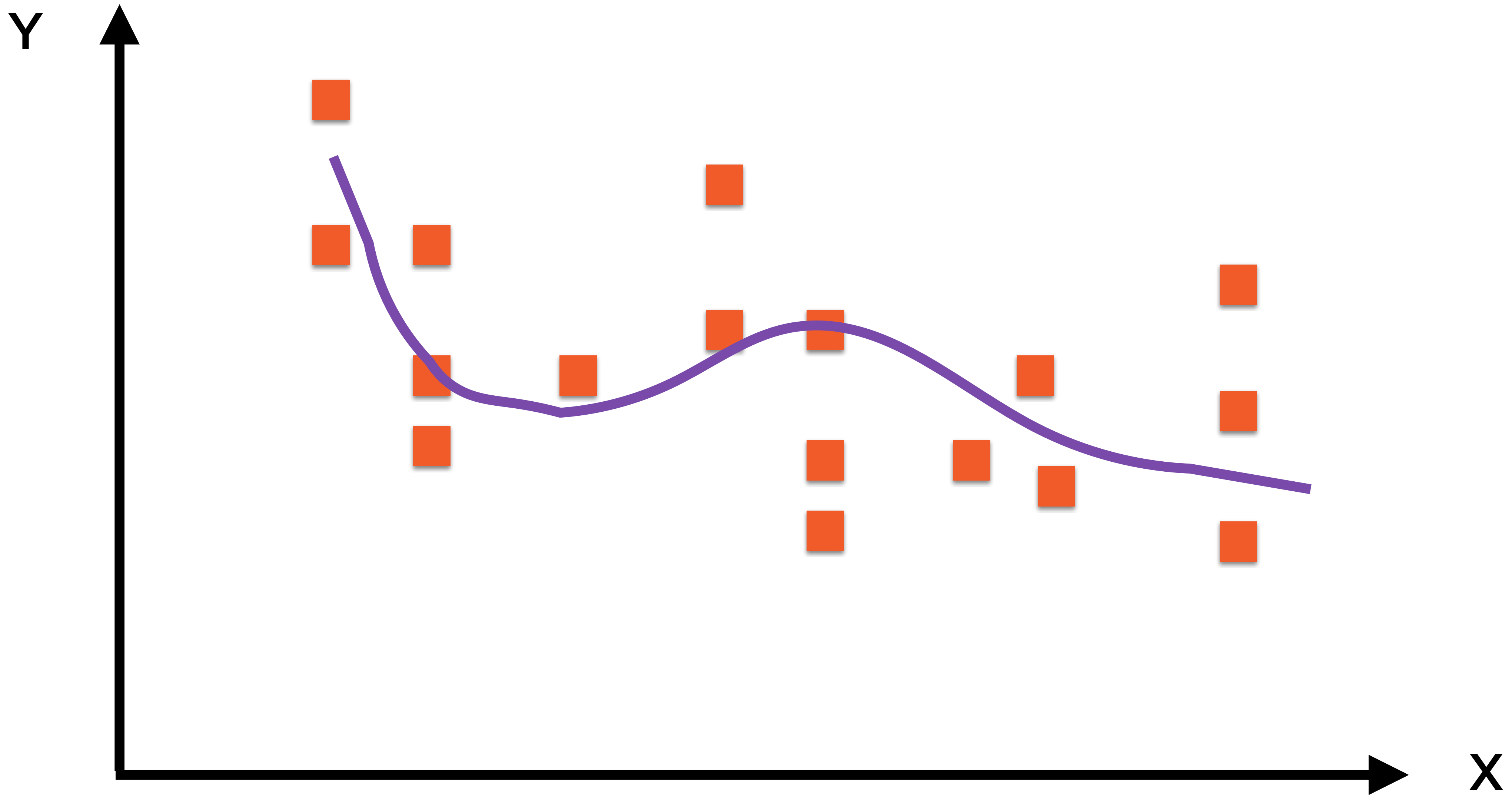
Challenge: Fit the “best” curve through these points

Good Fit?



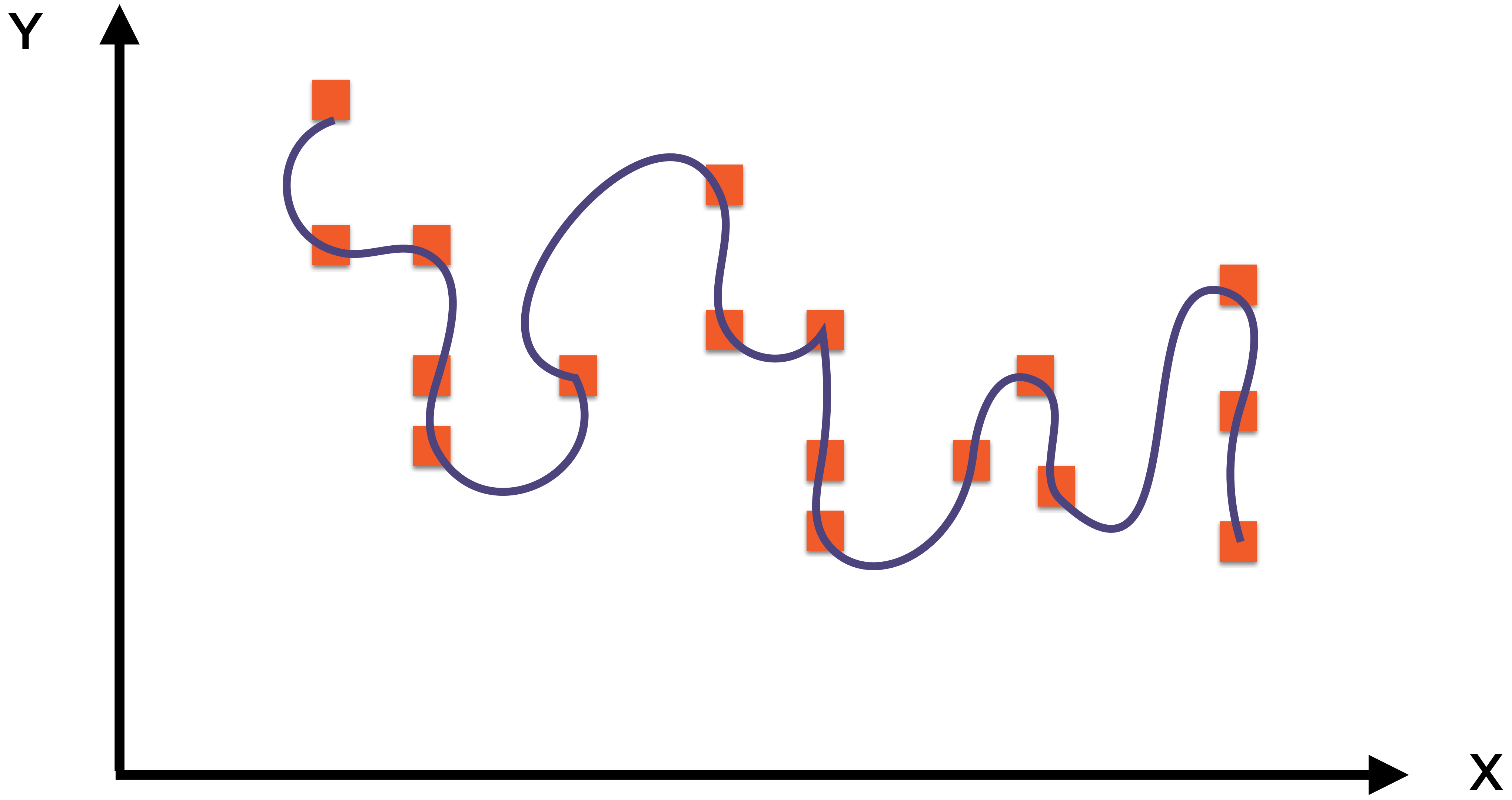
A curve has a “good fit” if the distances of points from the curve are small

Connecting the Dots



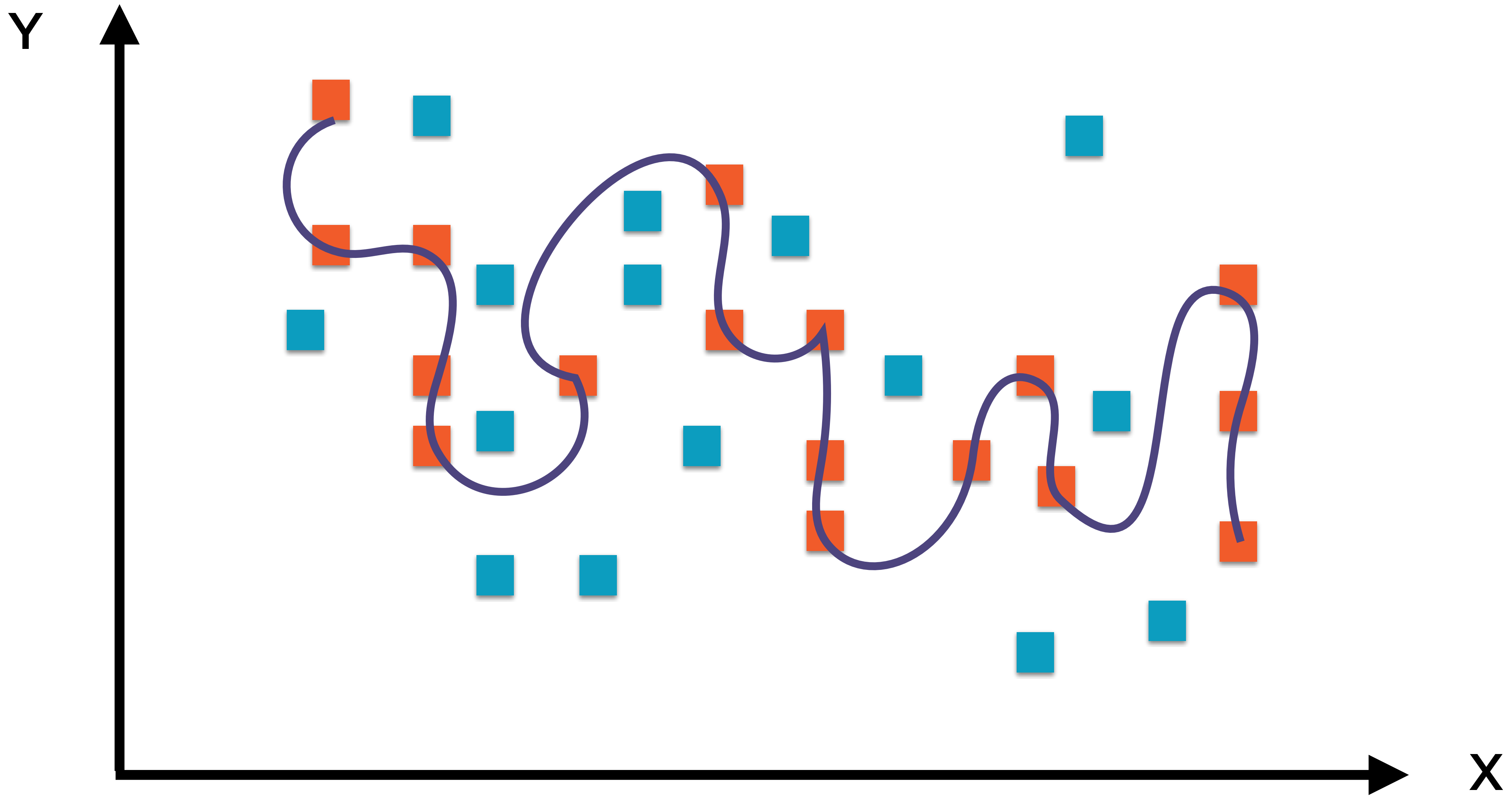
We could draw a pretty complex curve

Connecting the Dots



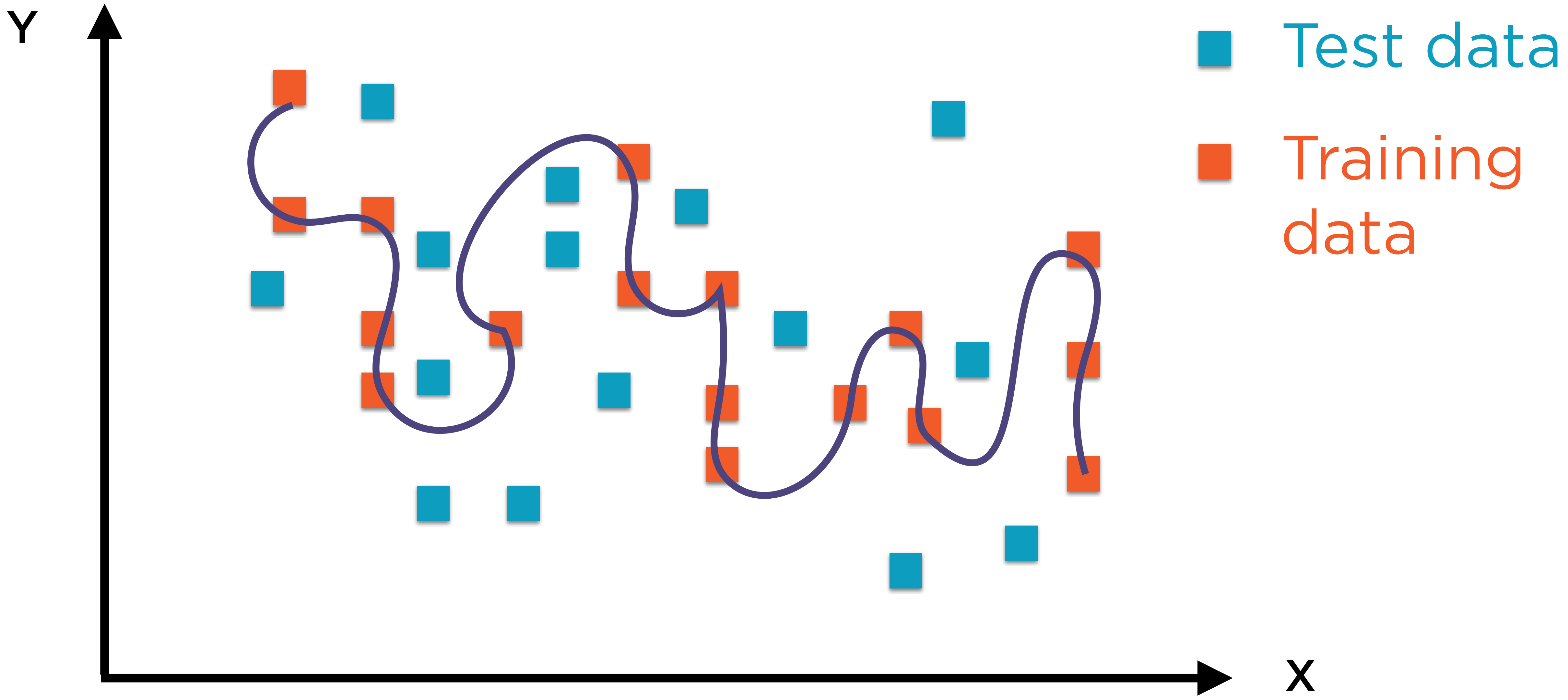
We can even make it pass through every single point

Connecting the Dots



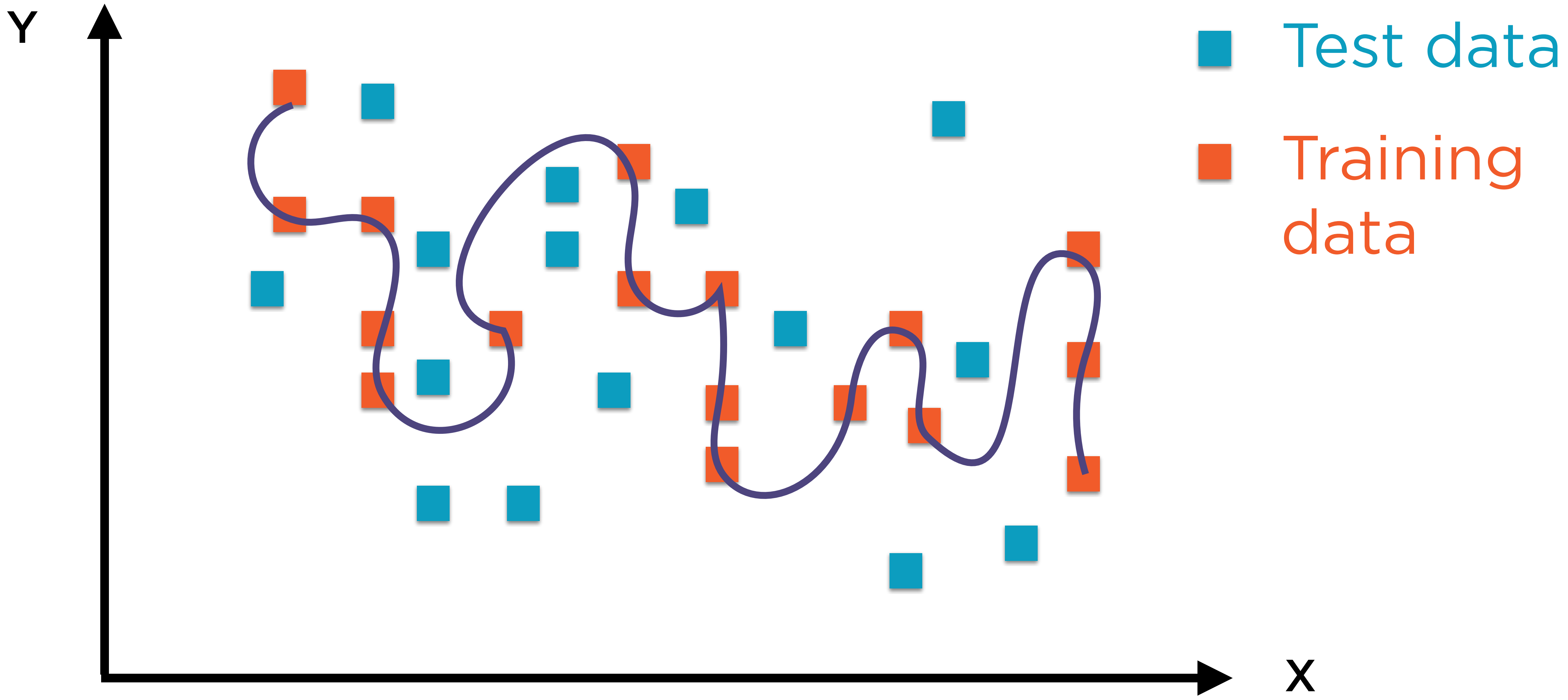
But given a new set of points, this curve might perform quite poorly

Connecting the Dots



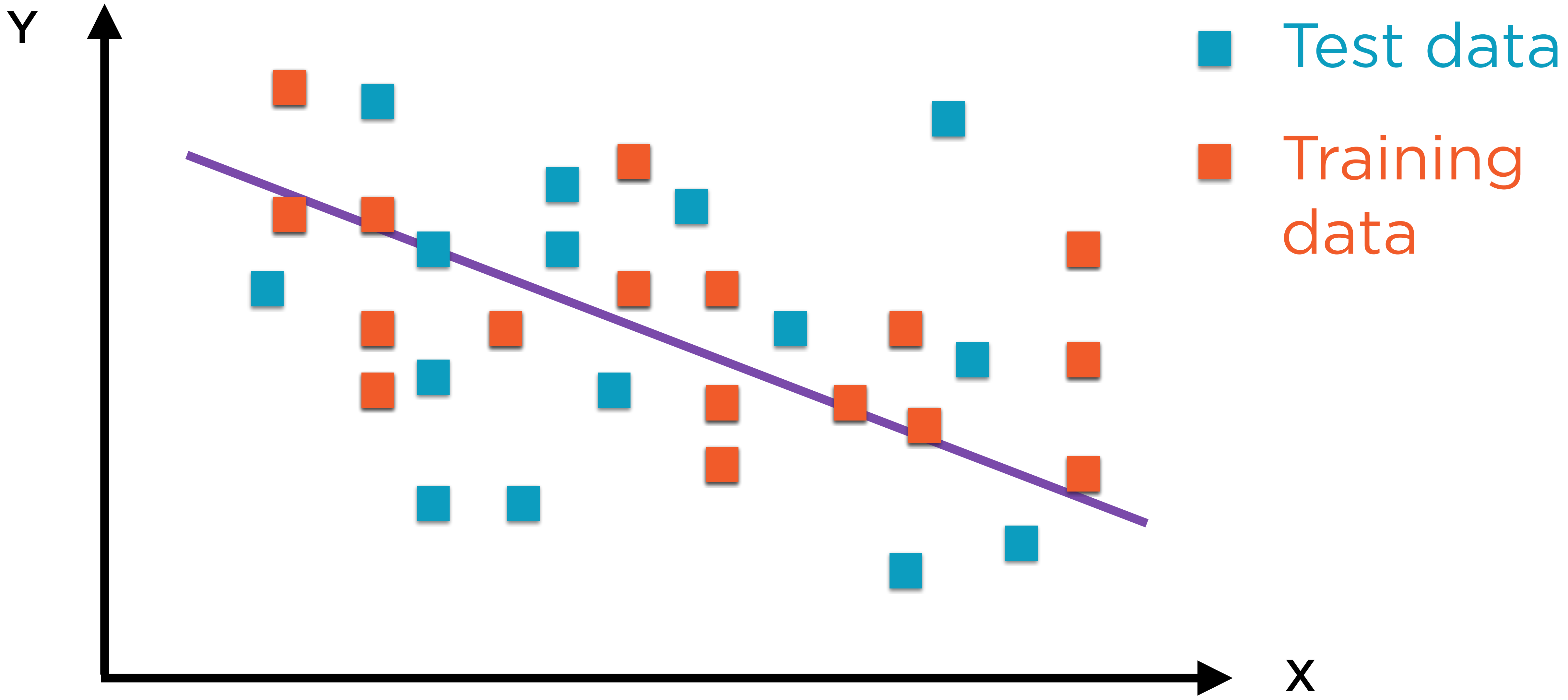
The original points were "training data", the new points are "test data"

Overfitting



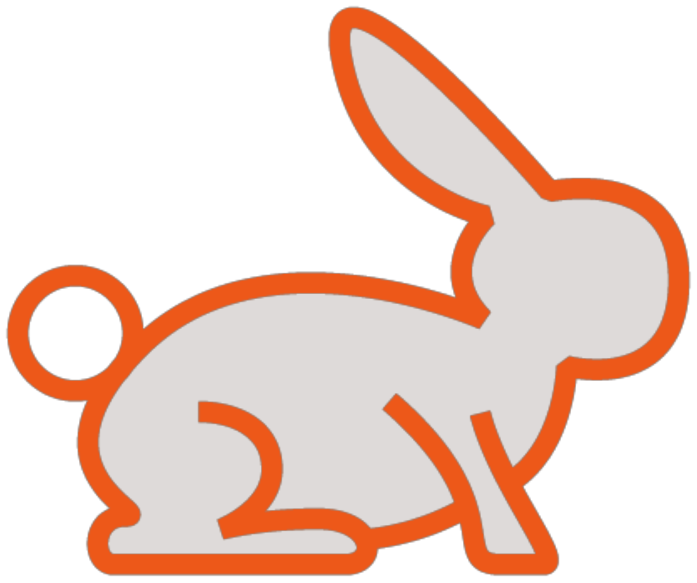
Great performance in training, poor performance in real usage

Connecting the Dots



A simple straight line performs worse in training, but better with test data

Overfitting



Low Training Error

Model does very well in training…

High Test Error

…but poorly with real data

Sparse Datasets

As you add dimensions instances tend to be very far from one another

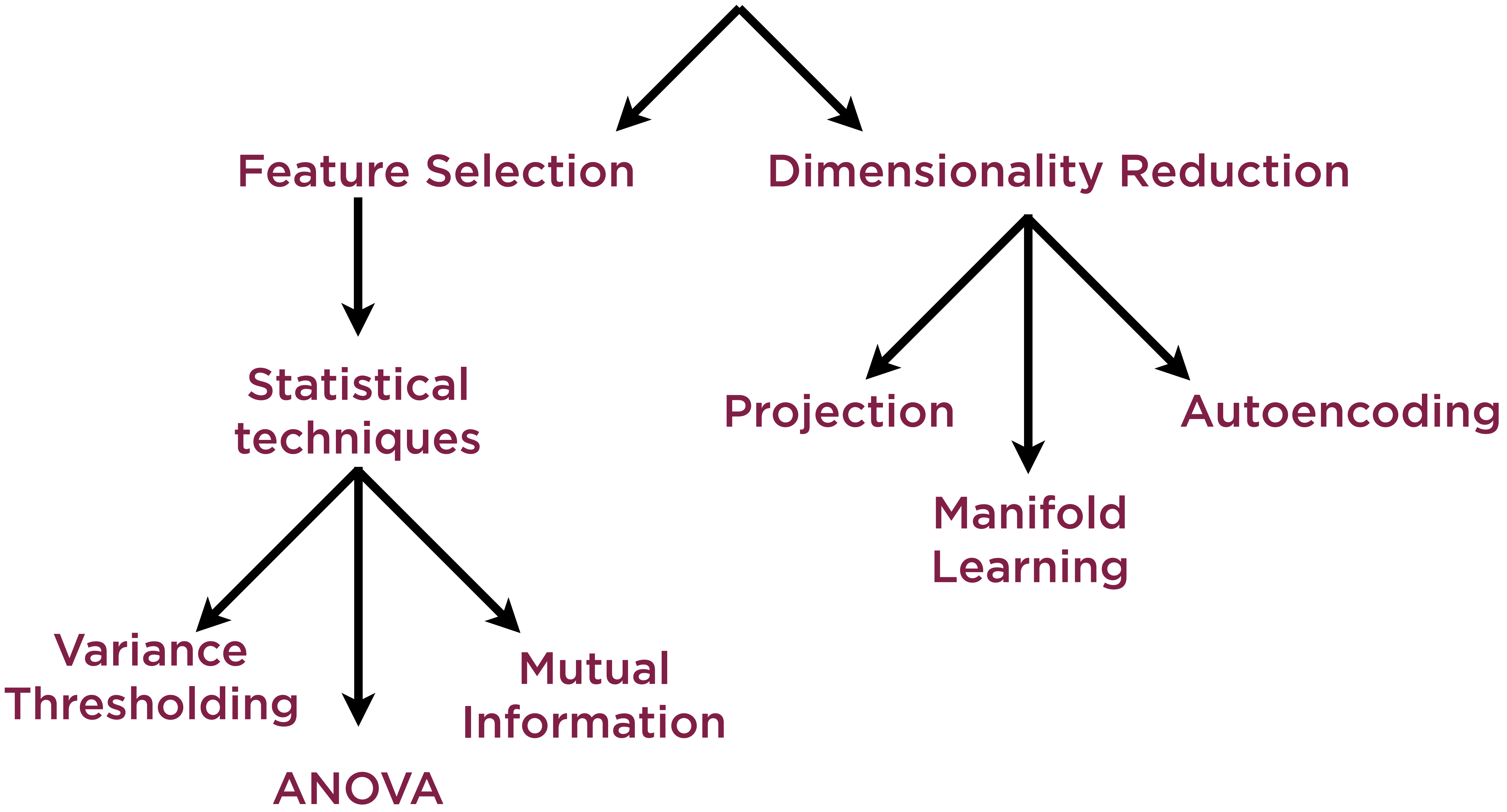
Each prediction instance will be far away from training instances

Not many instances with the same features

Hard to learn patterns

Solutions for Reducing Complexity

Reducing Complexity



Feature Selection

Dimensionality Reduction

Statistical techniques

Projection

Manifold Learning

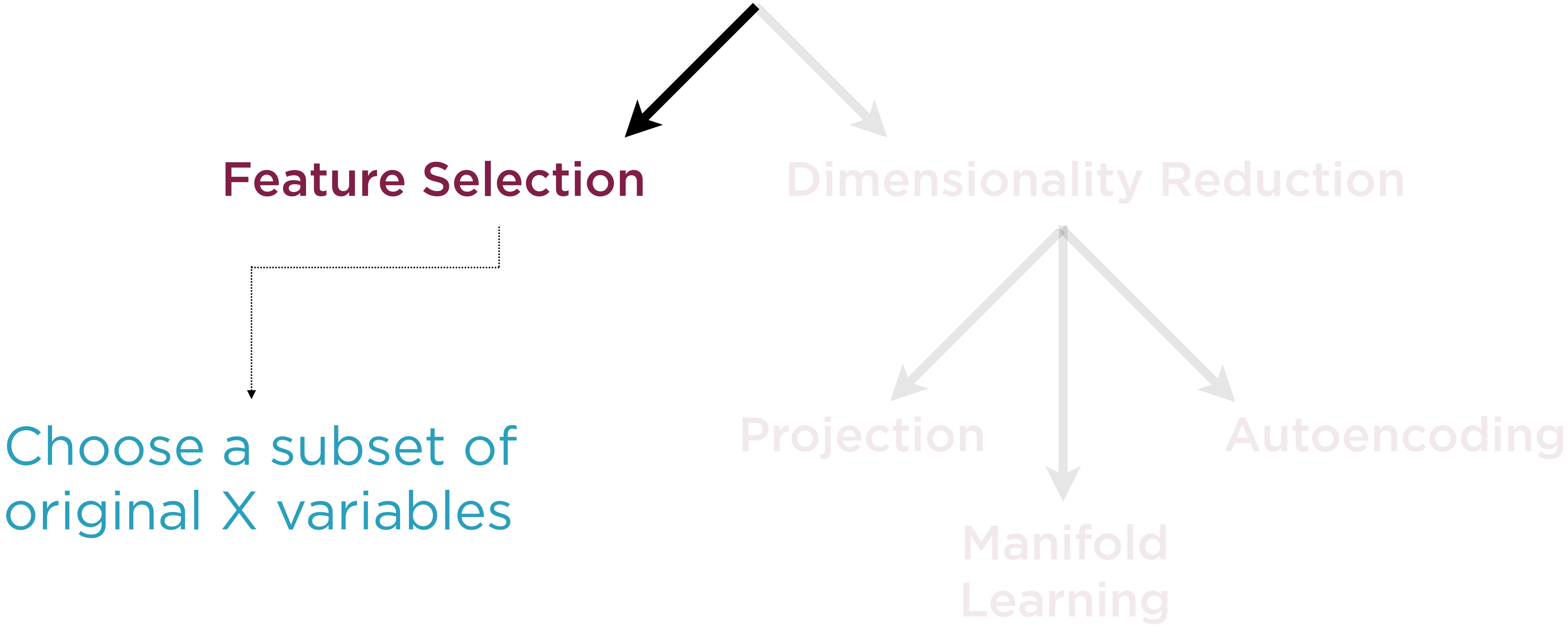
Autoencoding

Variance Thresholding

ANOVA

Mutual Information

Reducing Complexity



Feature Selection

Dimensionality Reduction

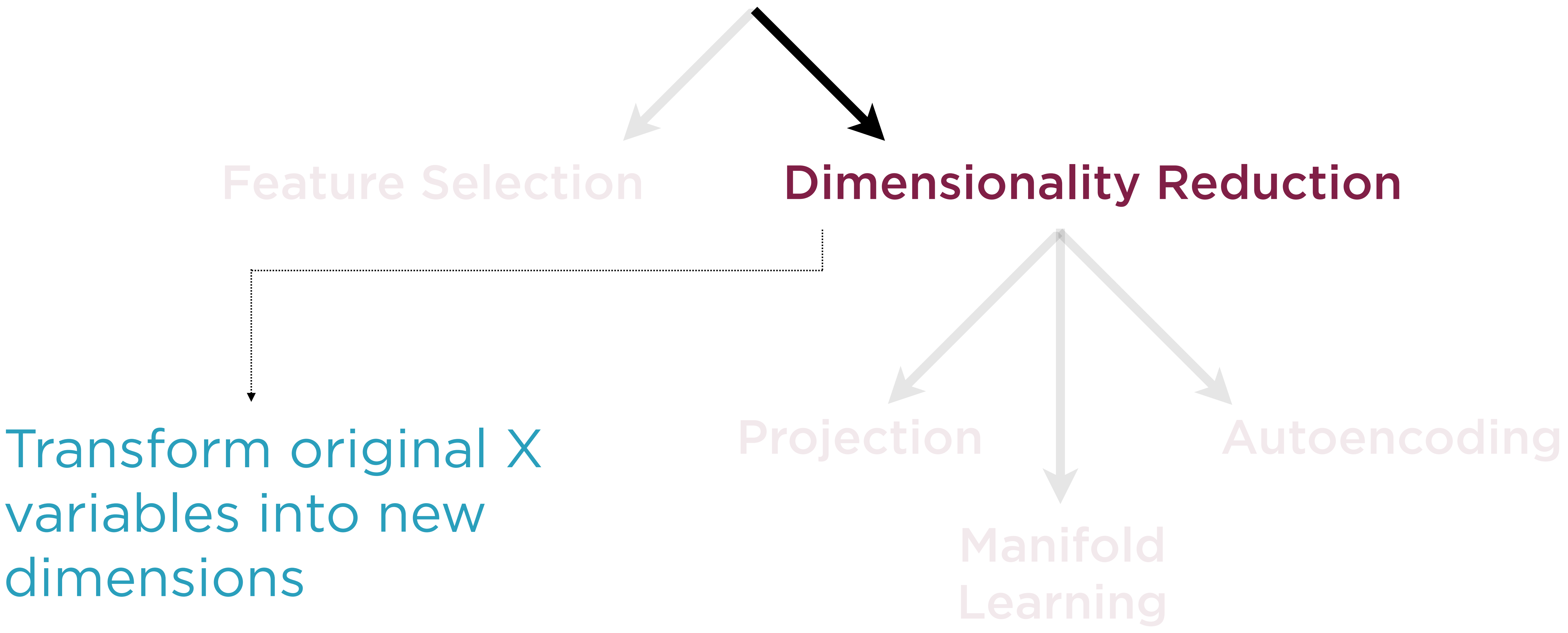
Choose a subset of original X variables

Projection

Manifold Learning

Autoencoding

Reducing Complexity



Feature Selection

Dimensionality Reduction

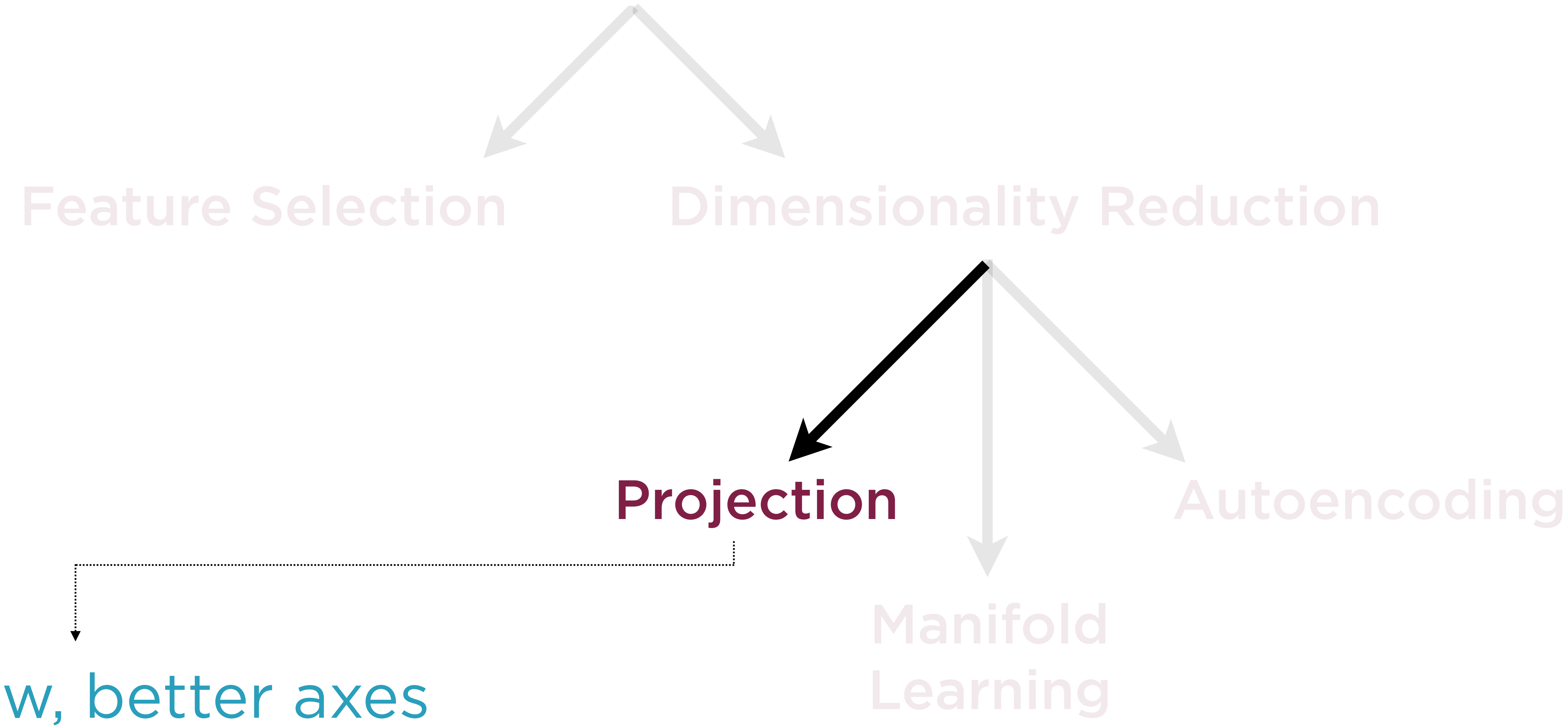
Transform original X variables into new dimensions

Projection

Manifold Learning

Autoencoding

Reducing Complexity



Feature Selection

Dimensionality Reduction

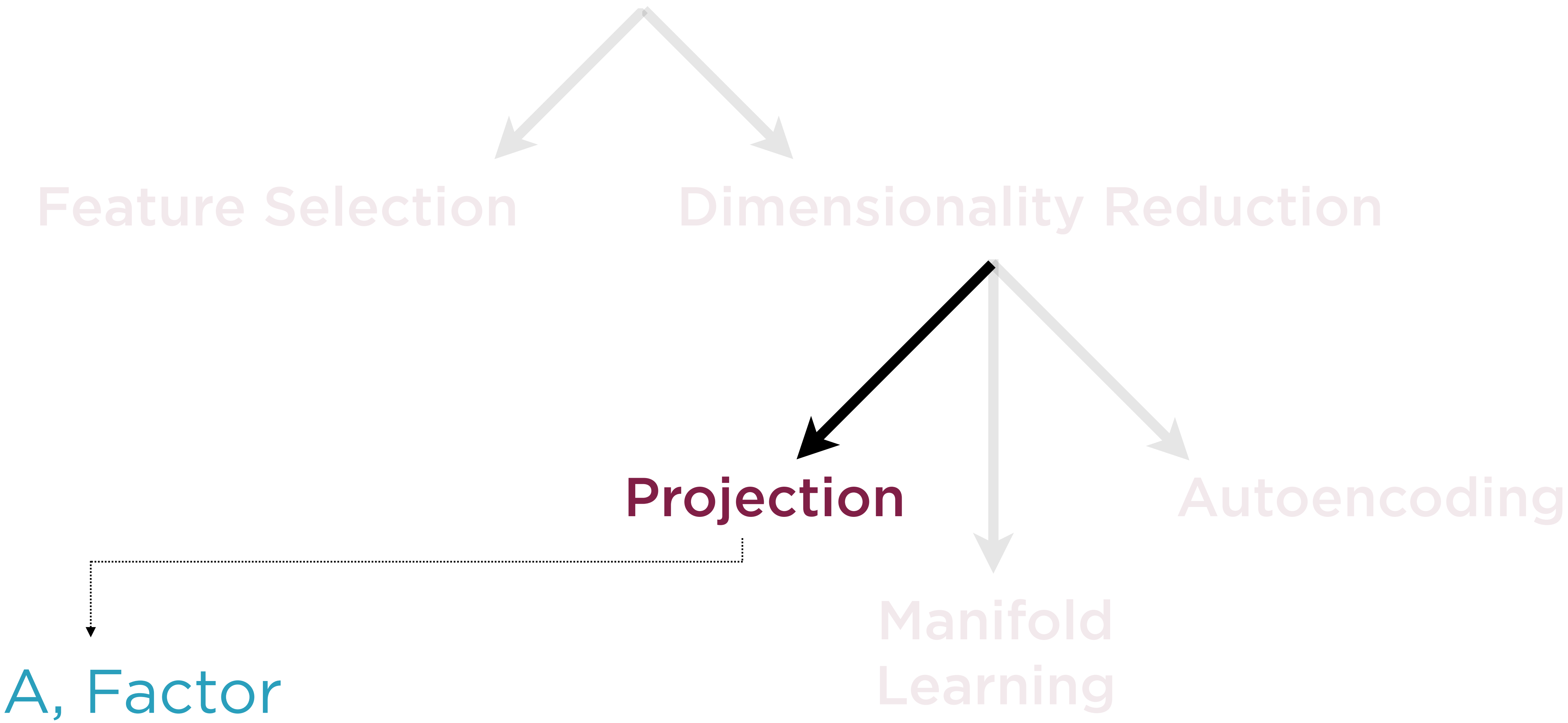
Projection

Manifold Learning

Autoencoding

Find new, better axes
and re-orient data

Reducing Complexity



Feature Selection

Dimensionality Reduction

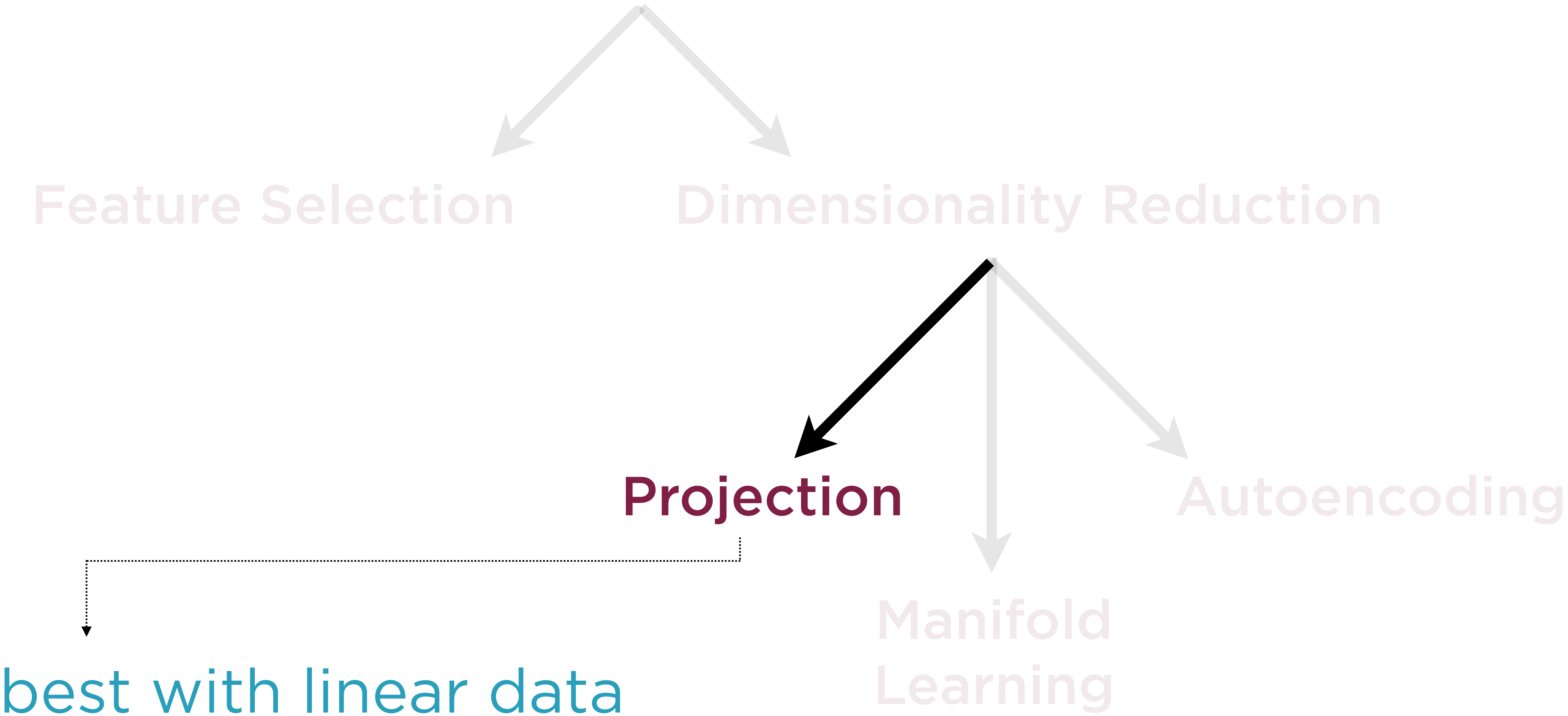
Projection

Manifold Learning

Autoencoding

e.g. PCA, Factor Analysis, LDA, QDA

Reducing Complexity



Feature Selection

Dimensionality Reduction

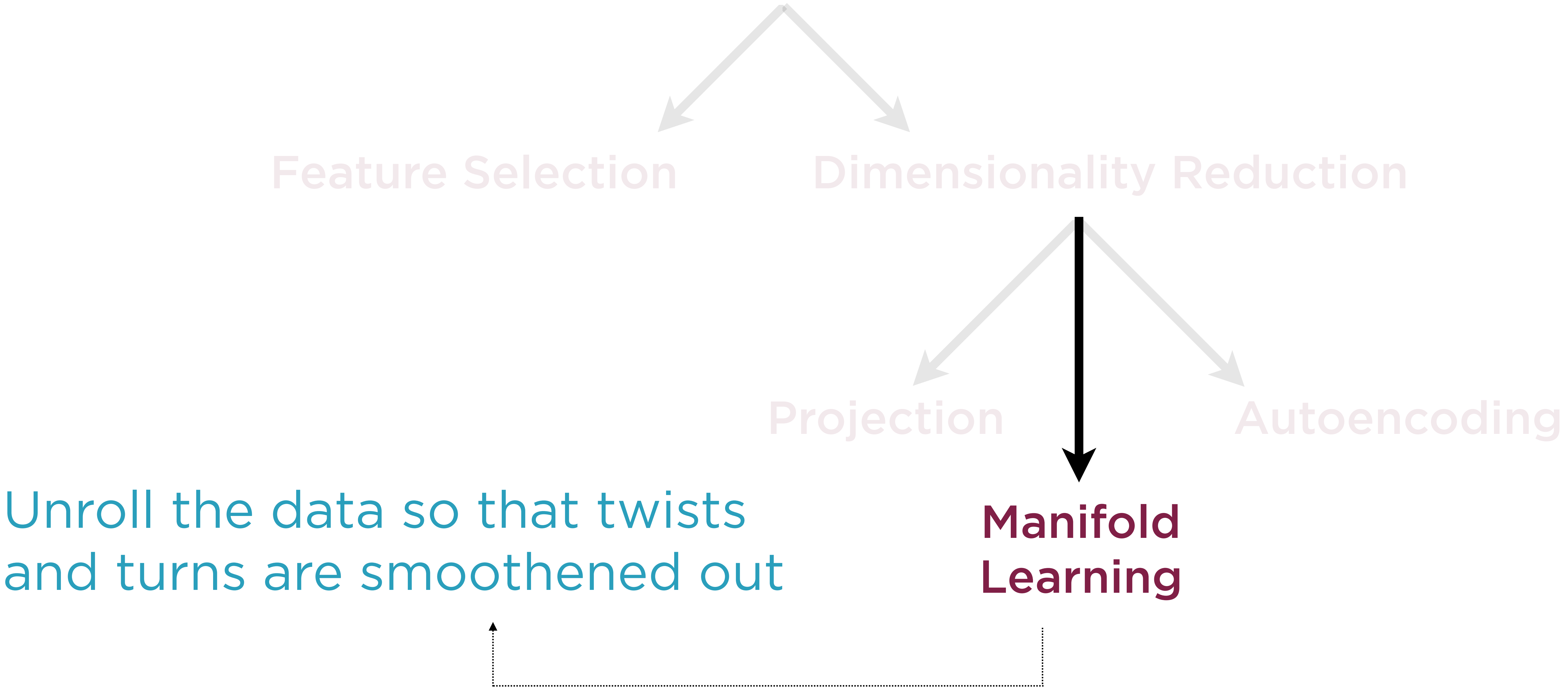
Projection

Manifold Learning

Autoencoding

Works best with linear data (can use kernel trick to extend to non-linear data)

Reducing Complexity



Feature Selection

Dimensionality Reduction

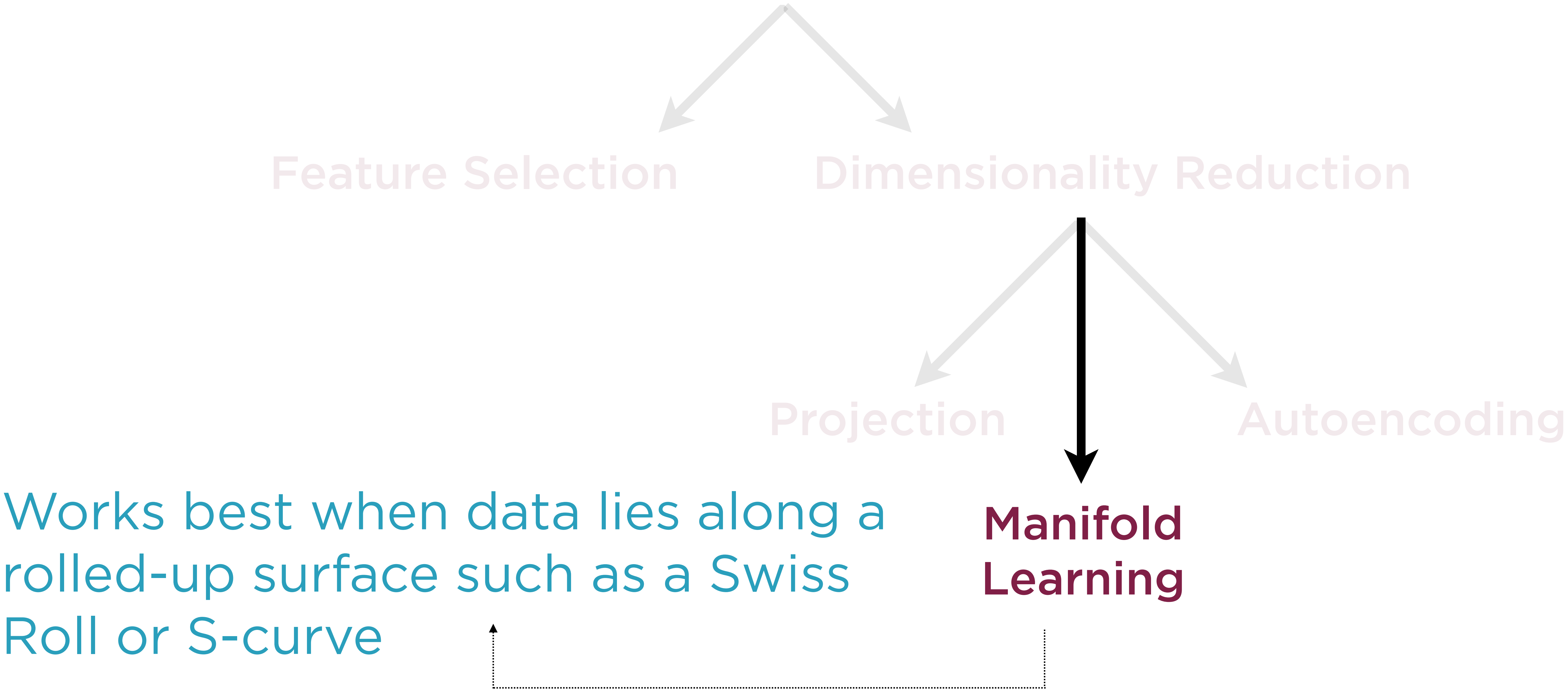
Projection

Autoencoding

Manifold Learning

Unroll the data so that twists and turns are smoothened out

Reducing Complexity



Feature Selection

Dimensionality Reduction

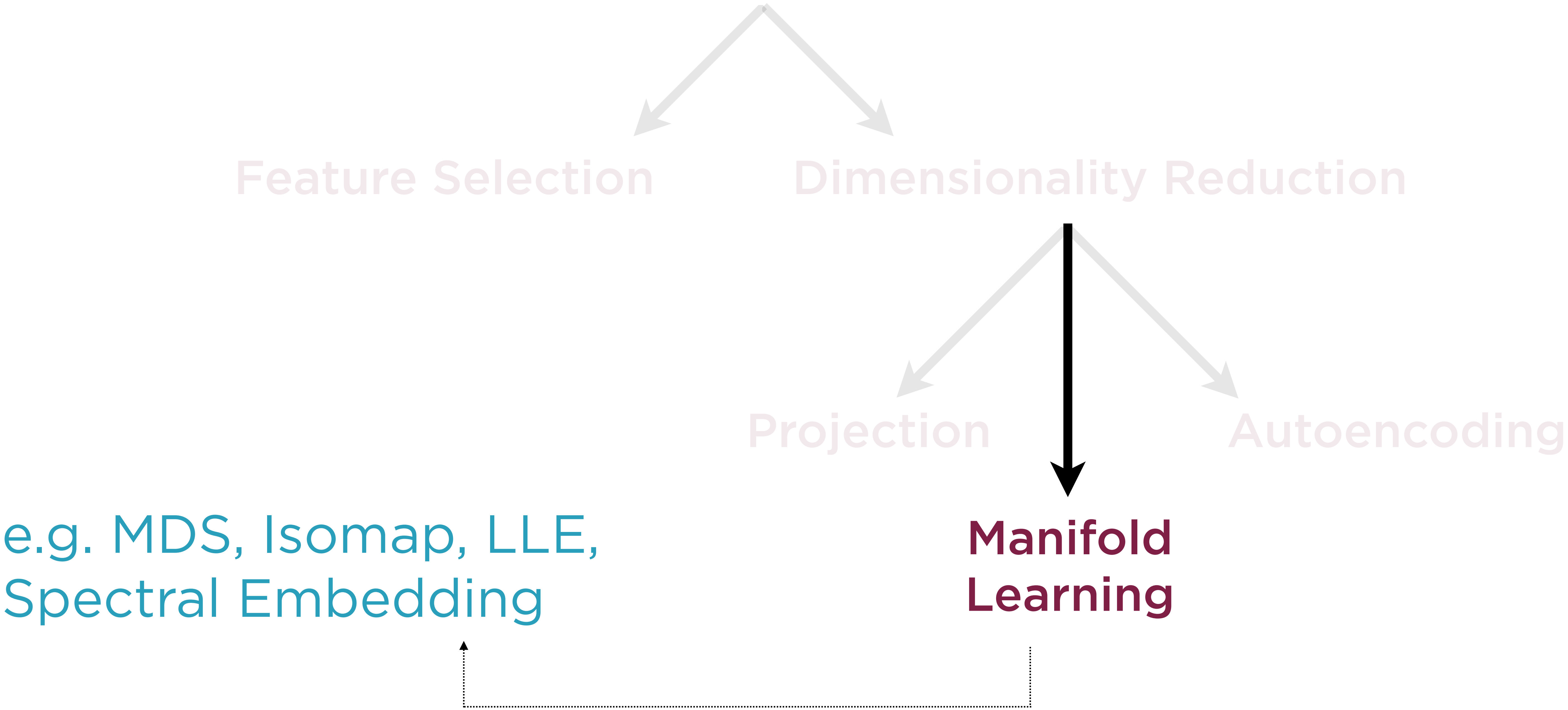
Projection

Autoencoding

Manifold Learning

Works best when data lies along a rolled-up surface such as a Swiss Roll or S-curve

Reducing Complexity



Feature Selection

Dimensionality Reduction

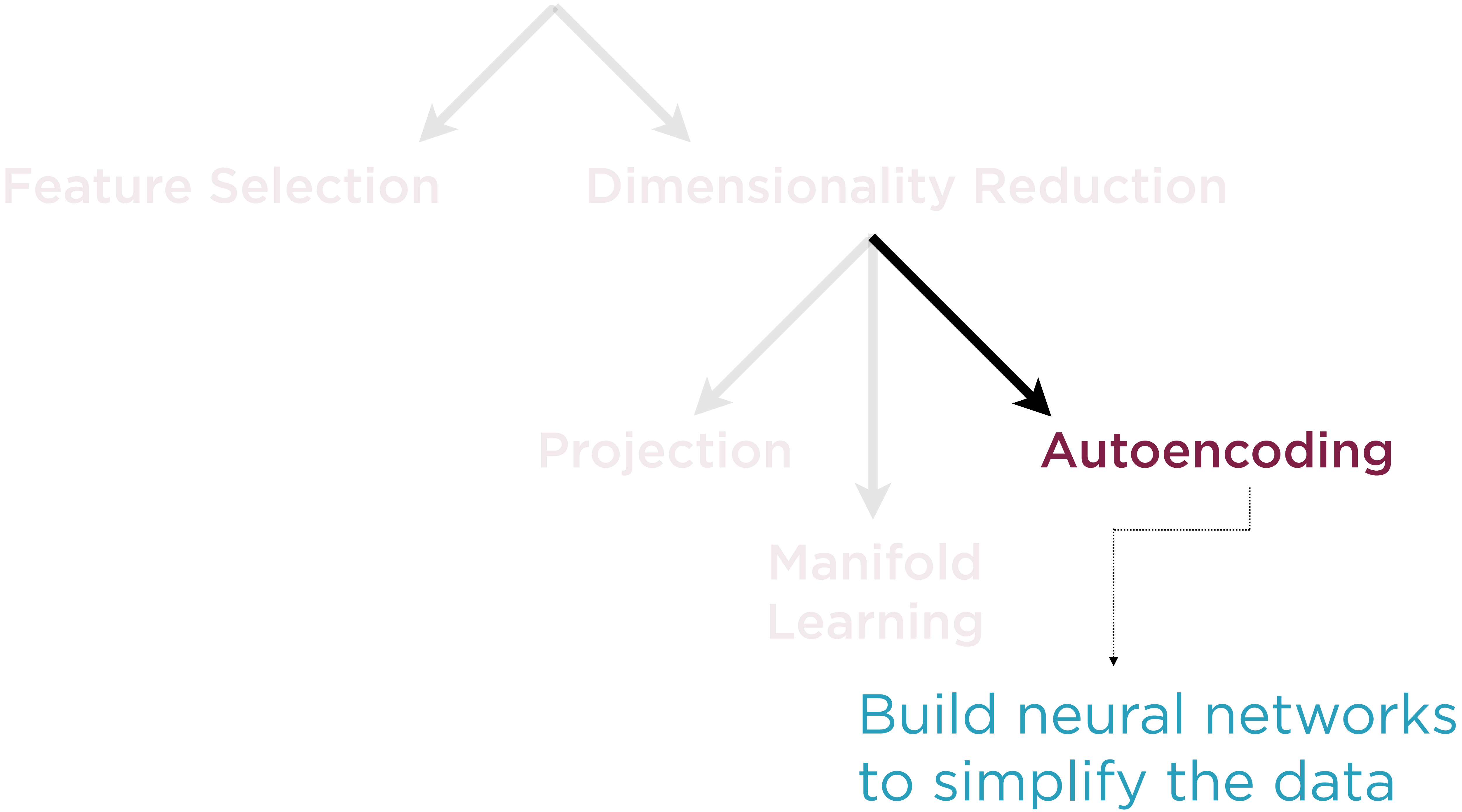
Projection

Manifold Learning

Autoencoding

e.g. MDS, Isomap, LLE, Spectral Embedding

Reducing Complexity



Feature Selection

Dimensionality Reduction

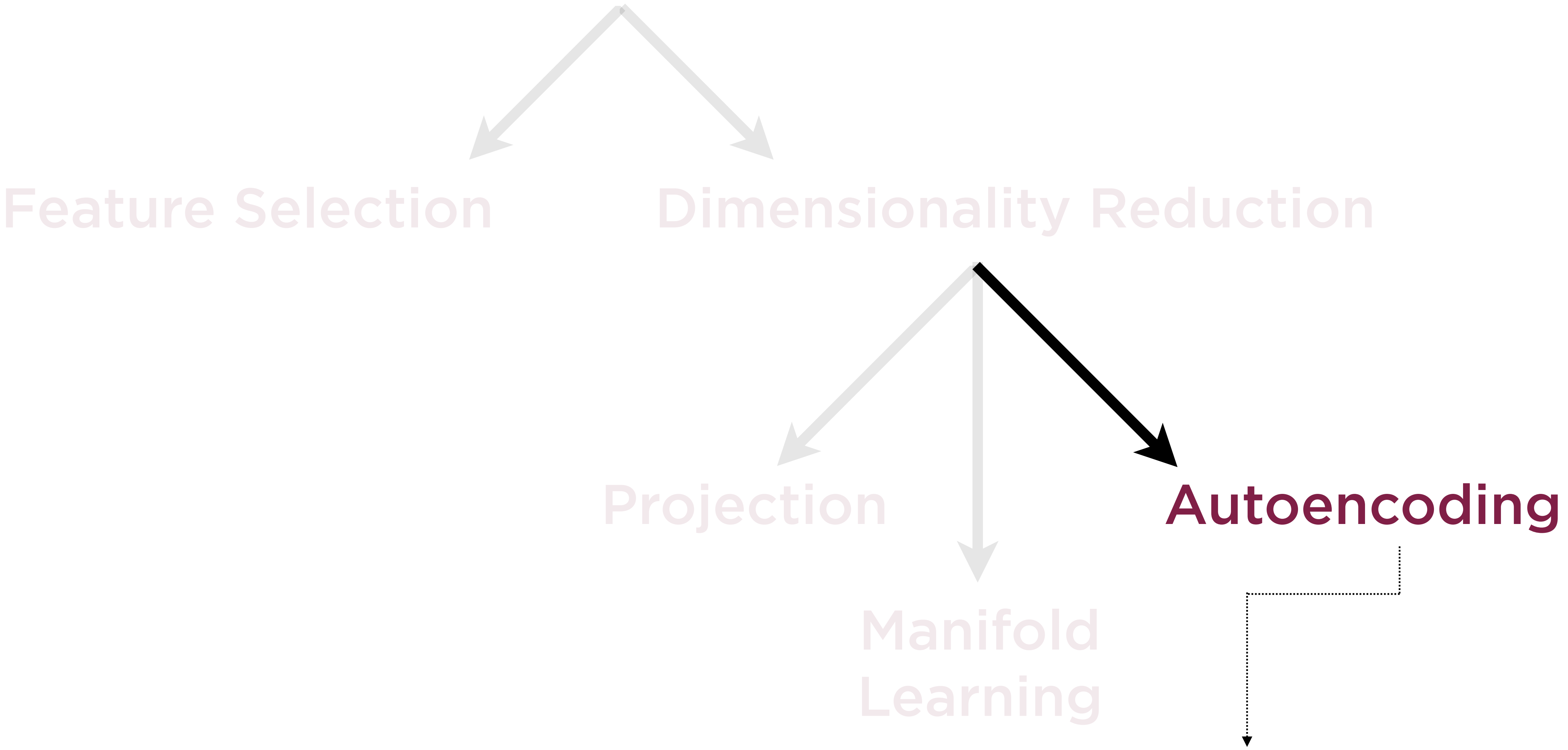
Projection

Manifold Learning

Autoencoding

Build neural networks to simplify the data

Reducing Complexity



Feature Selection

Dimensionality Reduction

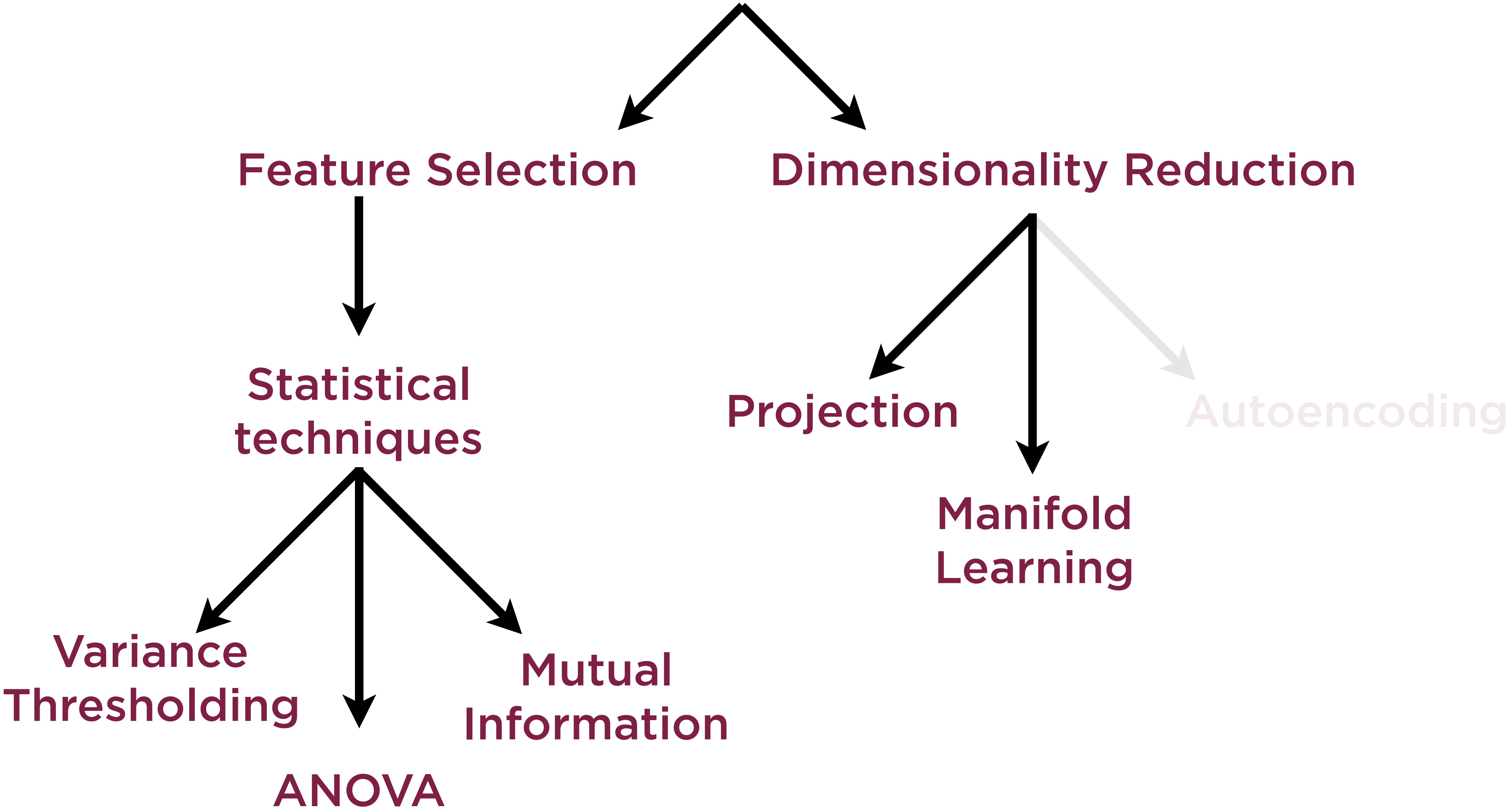
Projection

Manifold Learning

Autoencoding

Extract efficient representations of complex data

Estimators in scikit-learn



Feature Selection

Dimensionality Reduction

Statistical techniques

Projection

Manifold Learning

Autoencoding

Variance Thresholding

ANOVA

Mutual Information

Demo

Exploring the breast cancer dataset for classification

Building a classification model which uses all input features

Demo

Exploring the King County housing prices dataset for regression

Building a kitchen sink regression model which uses all input features

Feature Selection and Dictionary Learning

Choosing Feature Selection

Use Case	Possible Solution
Many X-variables	Feature selection
Most of which contain little information	
Some of which are very meaningful	
Meaningful variables are independent of each other	

Variance Thresholding

If all points have same value for an X-variable, that variable adds no information. Extend this idea and drop columns with variance below a minimum threshold.

Chi-square (χ^2) Feature Selection

For each X-variable, use the Chi-square test to evaluate whether that variable and Y are independent. If yes, drop that feature. Used for categorical X and Y.

Check whether the observed data deviates from expected values in the analysis

The scikit-learn library supports **chi2** tests only for **classification** models

ANOVA

ANalysis ***O***f ***VA***riance

ANOVA Feature Selection

For each X-variable, use the ANOVA F-test to check whether mean of Y category varies for each distinct value of X. If not, drop that X-variable.

ANOVA is considered to be a special case of linear regression

The scikit-learn library has a test which performs **univariate linear regression analysis**

Mutual Information

Measures the amount of information obtained on one random variable by observing another

Mutual Information

Conceptually similar to using ANOVA F-test for feature selection; superior as it also captures non-linear dependencies (unlike ANOVA-based feature selection)

The scikit-learn library has different functions for mutual information tests for classification and regression models

Dictionary Learning

Representation learning method to find a sparse representation of input data

Demo

Using univariate statistics for feature selection

- Univariate linear regression tests
- Mutual information tests

Demo

Using dictionary learning for sparse representations of input data

Summary

Need for dimensionality reduction in building ML models

Overfitting and data sparsity

Feature selection using statistical techniques

Dictionary learning and atom extraction