Machine Learning 101 and H2O

Tips on Predictive Modeling

- Solve the right problem!
- A good answer yesterday beats a great answer tomorrow
- Simplify everything
  - Model
  - Data Size
  - Data Types
- Counting and averages are great analytics!
- Be mindful of “productionizing” your model
  - Input and Output

Outline

- Introduction to Machine Learning
  - Shared concepts
  - Feature creation
- Algorithm Basics
  - Unsupervised Methods
  - Supervised Methods
- Using ML models in practice
  - Scoring code and production
  - Getting Started with ML

What is Machine Learning?

- Set of statistical and optimization tools to best predict outcomes
- Focus on “production analytics”
  - Reducing need for
    - Sampling
    - Periodic revision of models
    - Subjective priors
    - Hand coding models in a production language

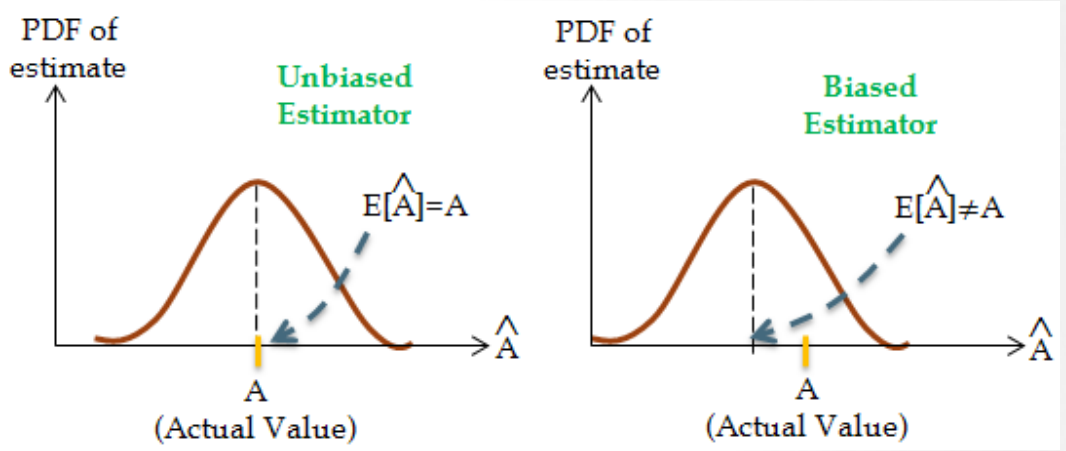
Vocabulary

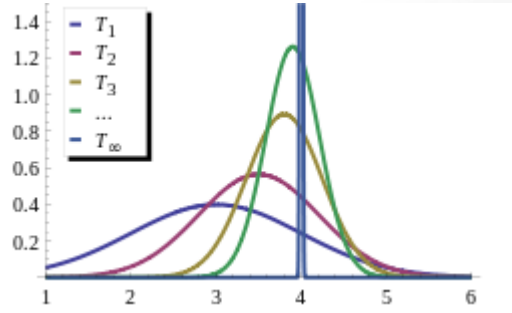
Concept	Statistics\Econometrics	Machine Learning
“Computation”	Fit\Estimate	Train
“Left-hand side”	Dependent variable	Target
“Right-hand side”	Regressor\Predictor\Class	Feature\Factor\Enum
“Goal”	Estimation\Explanation	Prediction

Statistics vs. Machine Learning

Statistics: Good estimators are….

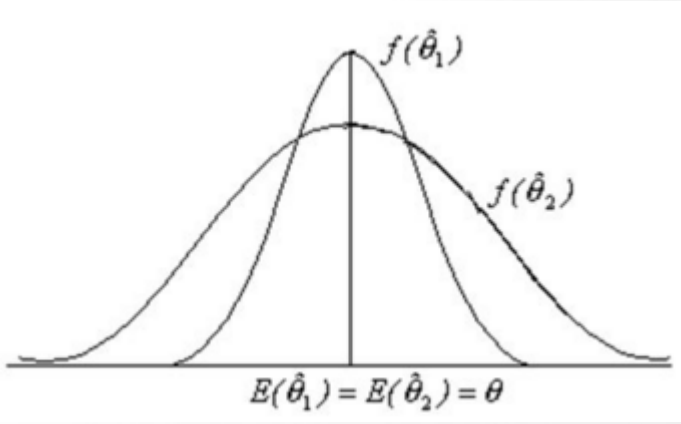
- Unbiased in small samples
- Consistent if not unbiased
- Efficient





Machine Learning: Good models….

- Predict well.



Train vs Test Data Sets

Training Set vs. Test Set

- Partition the original data (randomly or stratified) into a **training** set and a **test** set. (e.g. 70/30)

Training Error vs. Test Error

- It can be useful to evaluate the training error, but you should not look at training error alone.
- Training error is not an estimate of **generalization error** (on a test set or cross-validated), which is what you should care more about.
- Training error vs test error over time is an useful thing to calculate. It can tell you when you start to overfit your model, so it is a useful metric in supervised machine learning.

Train, Test and Validation

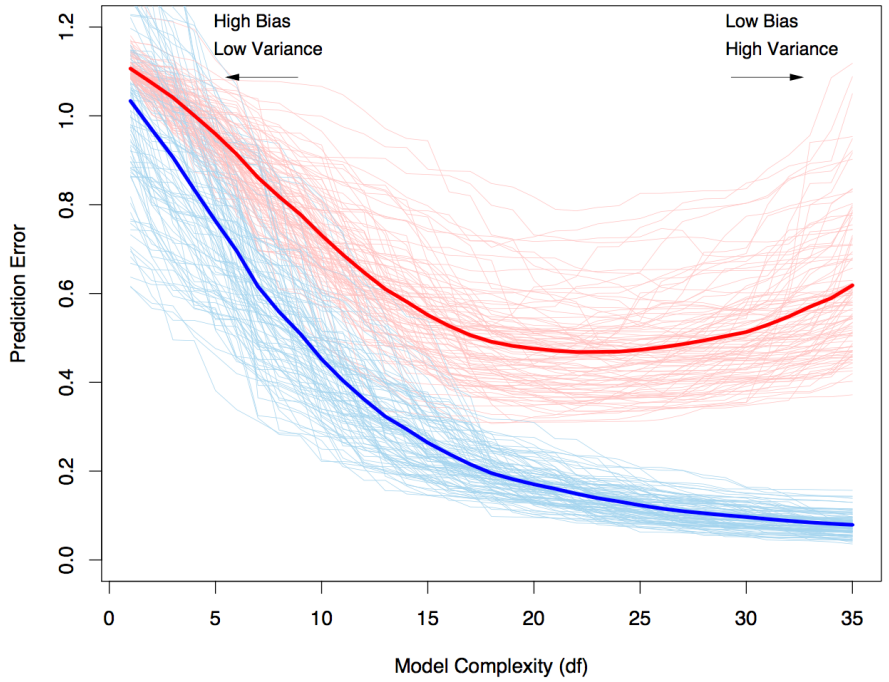


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for* 100 *training sets of size* 50 *each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\text{E}[\overline{\text{err}}]$.

Source: Elements of Statistical Learning

Feature Creation

- Features=Predictors (Factors, Regressors)
- Yes you need domain knowledge

- Example: If I give you a date/time stamp for a past transactions, what would you do with it?

Ensembles

What it is:

- "Ensemble methods use multiple learning algorithms to obtain better predictive performance that could be obtained from any of the constituent learning algorithms." (Wikipedia, 2015)
- Random Forests and Gradient Boosting Machines (GBM) are both ensembles of decision trees.
- Stacking, or Super Learning, is technique for combining various learners into a single, powerful learner using a second-level metalearning algorithm.

What it's not:

Ensembles typically achieve superior model performance over singular methods. However, this comes at a price — computation time.

Machine Learning Methods

Supervised Learning Methods

- Regression (GLM)
  - Lasso
  - Ridge
  - E
- De
- Ra
- Gradient Boosted Models
- Support Vector Machine
- Neural Network
- Deep Learning

Know Y

Unsupervised Learning Methods

- Clustering
  - Kmeans
  - Hierarchical
- Princ
- Auto
- Non-
- Generalized Low Rank Models

Don’t know Y

When do I use what tool?

Supervised Learning:

GENERALIZED LINEAR MODEL

What is the Generalized Linear Model (GLM)?

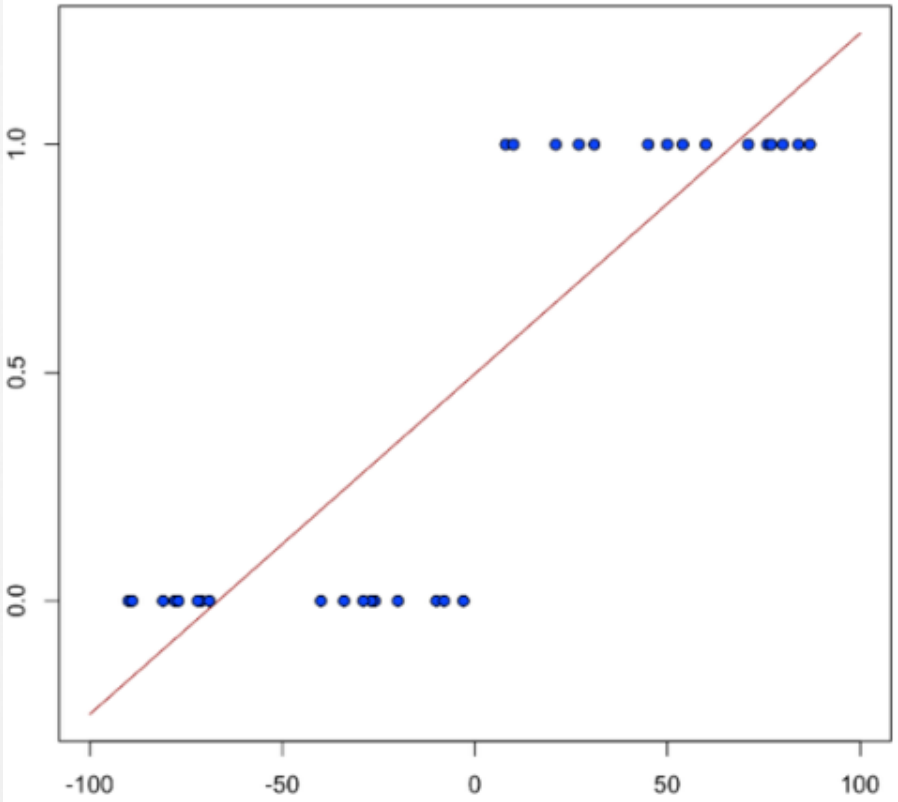
- Class of models that relate X (inputs) to Y (output)

$$\mathrm{E}(\mathbf{Y}) = \boldsymbol{\mu} = g^{-1}(\mathbf{X}\boldsymbol{\beta})$$
$$\mathrm{Var}(\mathbf{Y}) = \mathrm{V}(\boldsymbol{\mu}) = \mathrm{V}(g^{-1}(\mathbf{X}\boldsymbol{\beta})).$$

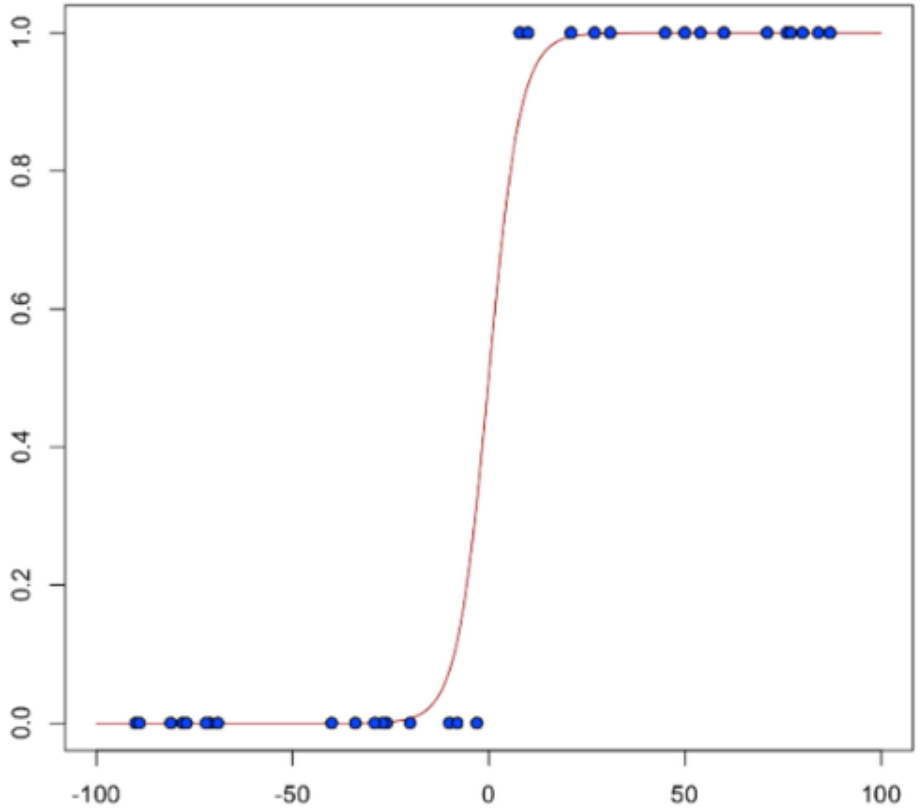
- Allows for a unification of models that have errors of the following form:
  - Normal (Gaussian)
  - Gamma
  - Poisson
  - Binomial (Logistic)
  - Multinomial
- MLE is found by iteratively reweighted least squares

Same predictors, different family and link functions

Linear Regression fit
(family=gaussian,link =identity)



Logistic Regression fit
(family=binomial,link = logit)



Pros and Cons of GLM's

Pros

- Fast to fit
- Easy to interpret
- Well understood statistical properties
- Continuous, categorical, count and classification problems
- Transparent scoring functions

Cons

- Possibility of overfitting
- Difficult to account for non-linearities
- Issues of multicollinearity
  - Non-unique solutions
- Difficult to fully account for interaction effects

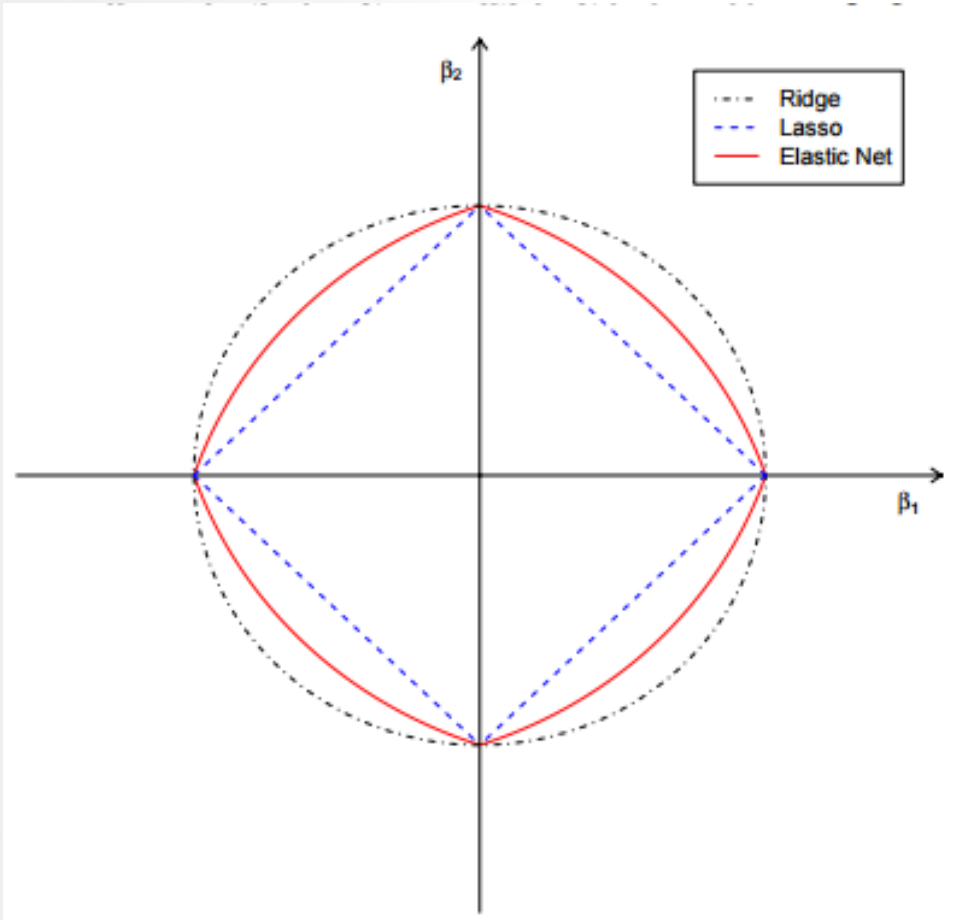
Supervised Learning:

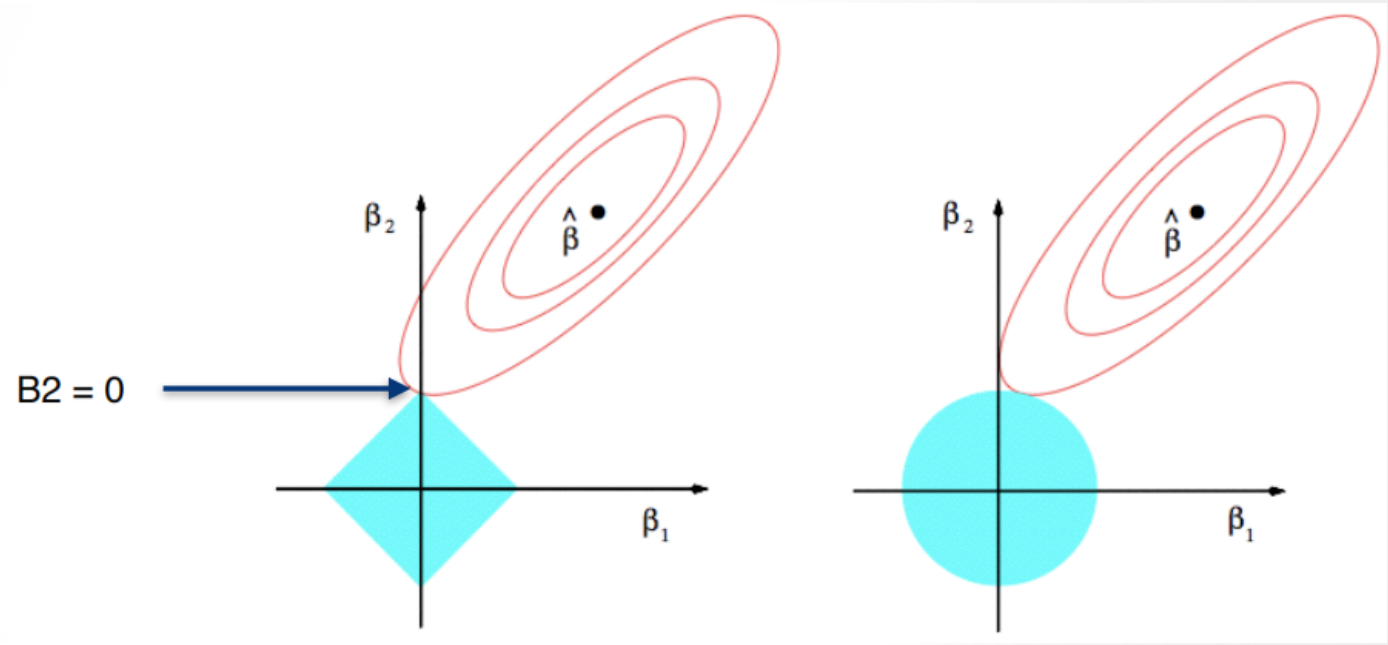
PENALIZED GENERALIZED LINEAR MODEL(LASSO, RIDGE, ELASTIC NET)

Regularized Regression

- Want to build a parsimonious but interpretable model
- Shrink the number of predictors and/or size by imposing penalties on the estimated coefficients (L1, L2)
  - LASSO: $\min_{\beta_0,\beta} \left\{ \frac{1}{N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \beta)^2 \right\}$ subject to $\sum_{j=1}^{p} |\beta_j| \leq t.$
    - picks one correlated variable, others discarded. Sparse.
  - Ridge: minimize $\sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^{\mathsf{T}} \mathbf{z}_i)^2$ s.t. $\sum_{j=1}^{p} \beta_j^2 \leq t$
    - correlated variables coefficients are pushed to the same value
  - Elastic Net: $\arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_2 \|\beta\|^2 + \lambda_1 \|\beta\|_1$
    - sparse solution, correlated variables grouped, enter/ leave the model together

Constraints for L1, L2 and L1&L2





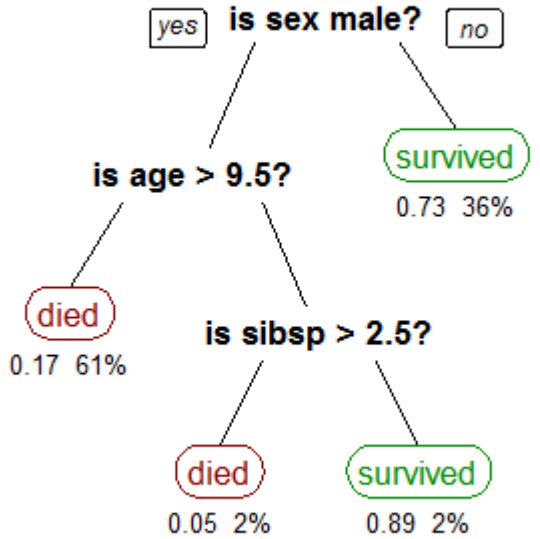
LASSO Regression vs. Ridge Regression

Supervised Learning:

DECISION TREES (RECURSIVE PARTITIONING)

Basics of Decision Trees

- Break sample data into homogenous pieces
  - Sample means
- Categorical and Continuous data
- Splits identify interaction effects and important variables
  - ID3
  - C4.5
  - CHAID



Wikipedia: Titanic Survivors

More info on trees

- Depth
- min rows
- Number of bins in histogram
- Number of categories from a enum (“binning”)
- Stopping criteria

Pros and Cons of Decision Trees

Pros

- Easy to compute
- Easy to interpret
- Robust to correlated features
- Robust to missing values
- Easy scoring functions
- Handles non-linearities in data

Cons

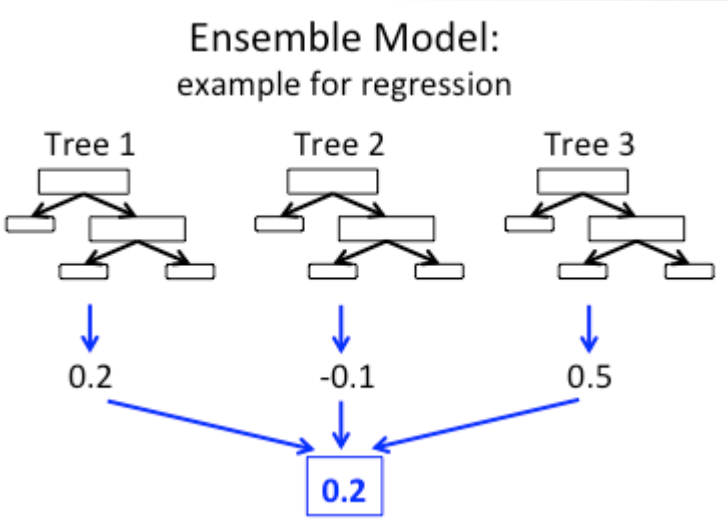
- Overfit data
- Will eliminate collinear variables
- Finds nonlinearities with linear relationships

Supervised Learning:

RANDOM FORESTS

What are Random Forests?

- Average a bunch of decision trees
  - Different trees on different samples and average
    - Different rows, different columns
    - Reduce variance with minimal bias increase
  - Bootstrap aggregation➔ “Bagging”



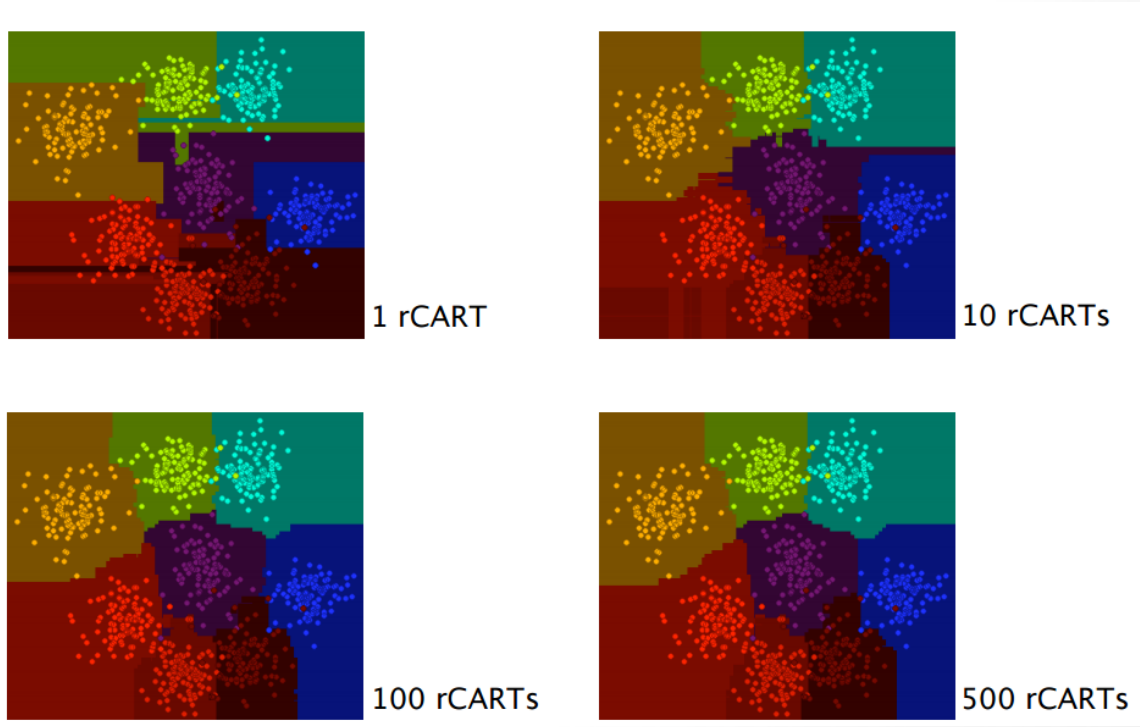
Pros and Cons of Random Forests

Strengths

- Easy to compute
  - Easy to parallelize
- Robust to outliers
- Variable importance

Weaknesses

- Slow to score
- Not transparent

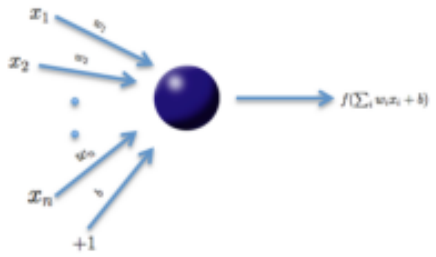


Supervised Learning:

DEEP LEARNING

What is Deep Learning?

What it is:

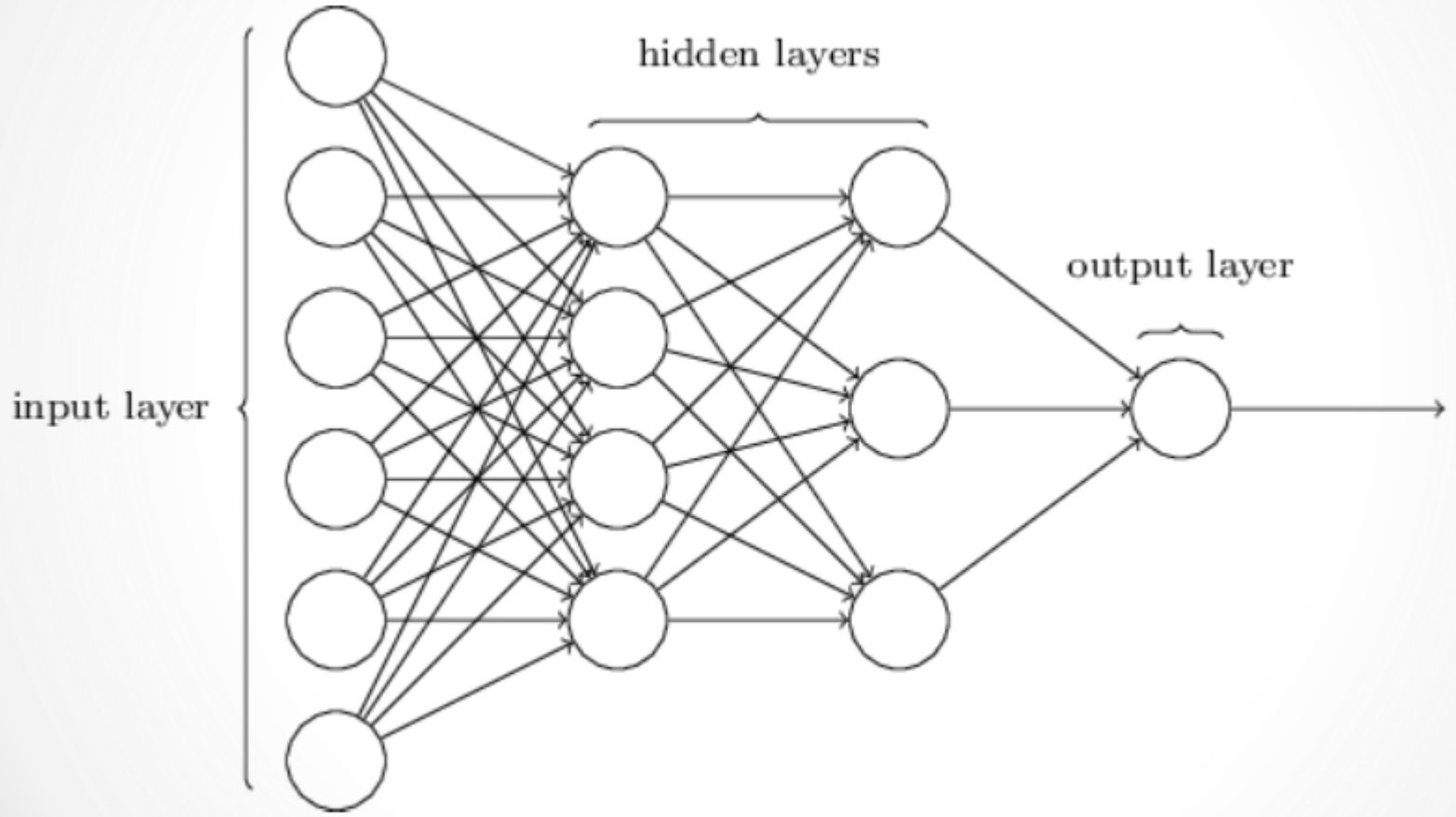


- "A branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using model architectures, composed of multiple non-linear transformations." (Wikipedia, 2015)
- Deep neural networks have more than one hidden layer in their architecture. That's what's "deep."
- Very useful for complex input data such as images, video, audio.

What it's not:

Deep learning architectures, specifically artificial neural networks (ANNs) have been around since 1980, so they are not new. However, there were breakthroughs in training techniques that lead to their recent resurgence (mid 2000's). Combined with modern computing power, they are quite effective.

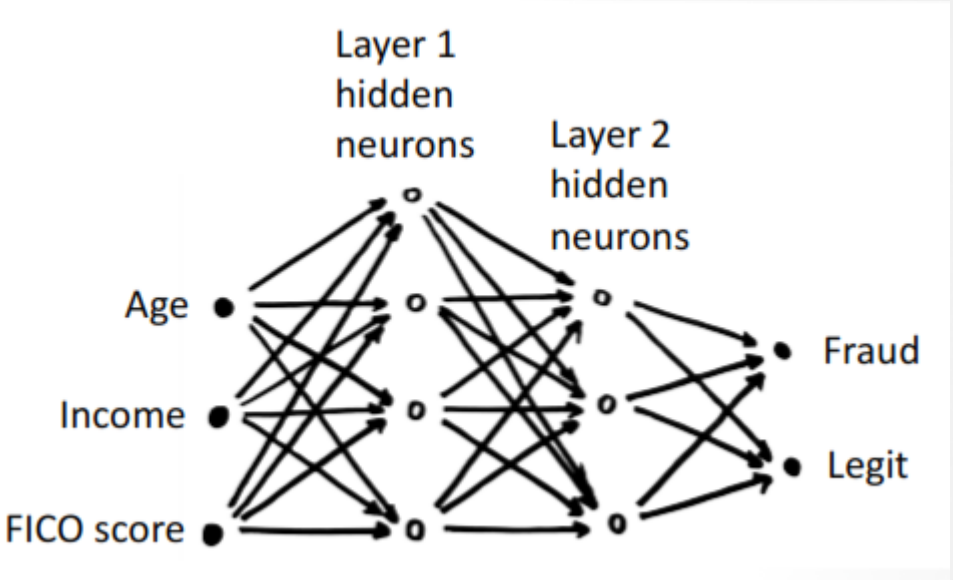
An example “Deep” Neural Network



Example of a deep neural net architecture.

VALUE OF DEEP LEARNING

- Deep Learning learns a **hierarchy of non-linear transformations**
- Neurons transform their input in a non-linear way
- **Black-box, brute-force method, really good at patte rn recognition**
- Deep Learning got a boost in the la st decade due to **faster hardware a nd algorithmic advances**



STRENGTHS AND WEAKNESSES

Strengths

- non linear
- robust to correlated features
- learned features can be extracted
- can stop training at any time
- can be fine-tuned with more data
- great ensemble member
- efficient for multi-class problems
- world-class at pattern recognition

Weaknesses

- slow to train
- slow to score
- not interpretable
- results not fully reproducible
- theory not well understood
- overfits, needs regularization
- many hyper-parameters
- expands categorical variables
- must impute missing values

UNSUPERVISED LEARNING MODELS

Unsupervised Learning Models

- No target!
- Ways to characterize or simplify data

- Uses
  - Clustering (Put rows together)
  - Variable reduction (Put columns together)
  - Anomaly detection

Unsupervised Learning:

K-MEANS CLUSTERING

Clustering

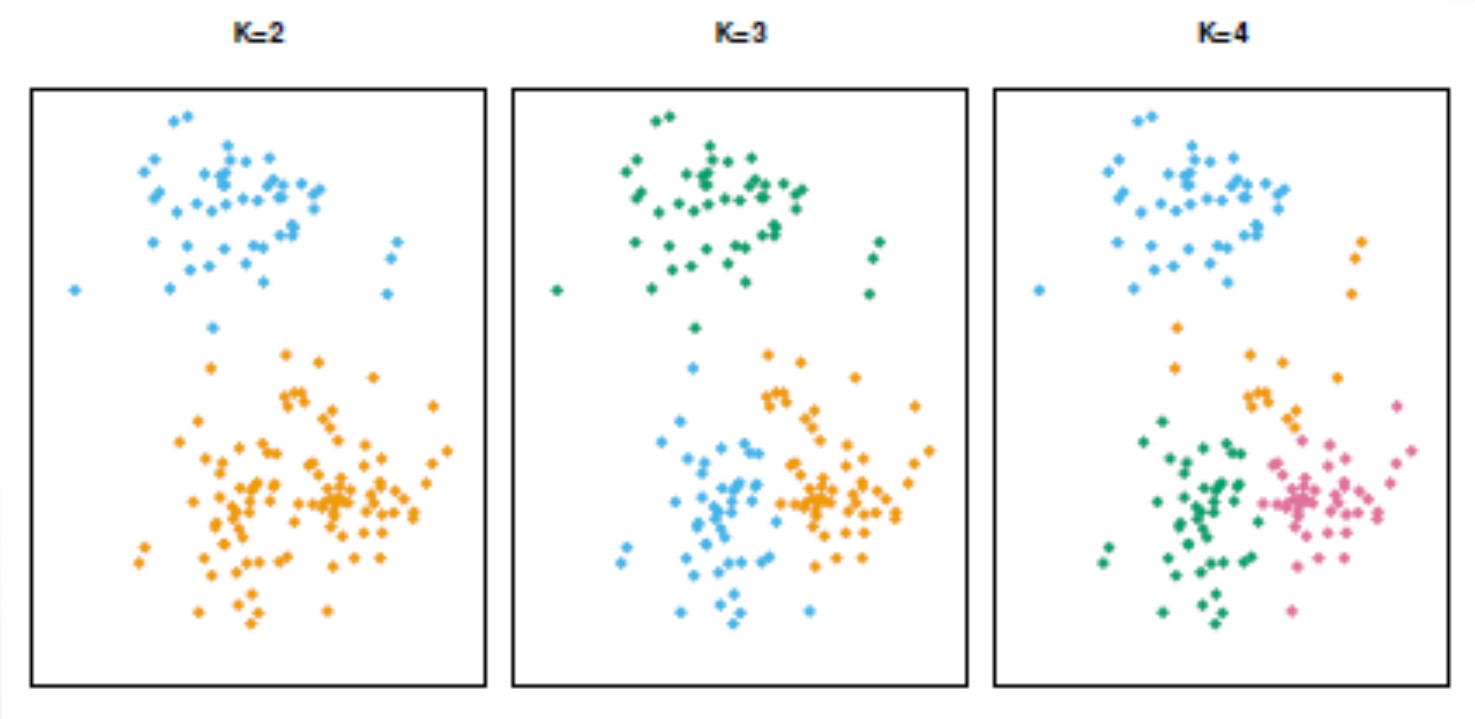
- We are going to try and “bundle” customers/companies/transactions based on “likeness” of certain features.
- These “clusters” will help us simplify and potentially “segment” observations.
  - Segment customers
  - Categorize events

Clustering

Objective: Minimize the within cluster variation for a given k

How to choose k?

- Choose several and evaluate performance
- Often business rules determine feasible k



Unsupervised Learning:

PRINCIPAL COMPONENTS ANALYSIS (PCA)

What is Principal Components Analysis

- Another method of data simplification
- Can I reduce the width (dimensionality) of the dataset by creating one or more new PCA variables that do a “good job” of representing the original signal in the data?
- Uses:
  - Variable reduction=> Fewer predictors
  - Feature extraction=>Anomaly detection

Operationalizing Machine Learning:

SCORING AND SCORING CODE

Score code

- Operationalizing your model means handing off “Score Code”

- Scoring within compute engine
- PMML
- Low level representations
  - Java Object
  - C

Operationalizing Machine Learning:

TIPS ON GETTING STARTED

Balancing Accuracy and Interpretability

- Let's you try different regression techniques
- Train a "black box model" but use it as a benchmark
  - Train explainable model on predictions from the black box
- Use the black box model as backup

- Use variable importance and partial dependency plots for explanation
- Use complicated model for feature creation

https://www.oreilly.com/ideas/predictive-modeling-striking-a-balance-between-accuracy-and-interpretability

H2O.ai

Robust, Tested and Supported Platform for Predictive Analytics

H2O.ai

H2O.ai Overview

- Founded: 2011: Version 3 released in 2015
- Product: H2O open source in-memory prediction engine
- Team: 50+ Core developers and data scientists
- HQ: Mountain View, CA. Sales: U.S., U.K. & Canada

H2O.ai
Machine Intelligence

Scientific Advisory Council

Dr. Trevor Hastie

- John A. Overdeck Professor of Mathematics, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Co-author with John Chambers, *Statistical Models in S*
- Co-author, *Generalized Additive Models*
- 108,404 citations (via Google Scholar)

Dr. Rob Tibshirani

- Professor of Statistics and Health Research and Policy, Stanford University
- PhD in Statistics, Stanford University
- COPPS Presidents' Award recipient
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Author, *Regression Shrinkage and Selection via the Lasso*
- Co-author, *An Introduction to the Bootstrap*

Dr. Stephen Boyd

- Professor of Electrical Engineering and Computer Science, Stanford University
- PhD in Electrical Engineering and Computer Science, UC Berkeley
- Co-author, Convex Optimization
- Co-author, Linear Matrix Inequalities in System and Control Theory
- Co-author, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*

H_2O.ai

What is H2O?

Platform

Open source in-memory prediction engine

- Parallelized and distributed algorithms making the most use out of multithreaded systems and grids
- GLM, Random Forest, GBM, Deep Learning(ANN), GLRM, PCA, K-means etc.

Access

Data, Platform and Client Agnostic

- Runs anywhere
- REST API– drives H2O from R, Python, Web UI, Excel, Tableau
- Score code for all models

Scale

Same code. Different Environments

- Use all of your data without sub setting
- No code changes to go from development to production

Single source of truth for R and Python users

Algorithms on H2O

Supervised Learning

Statistical Analysis	• **Generalized Linear Models (LASSO, Elastic Net, Ridge)** • Binomial, Gaussian, Gamma, Poisson and Tweedie • **Naïve Bayes**
Ensembles	• **Distributed Random Forest**: Classification or regression models • **Gradient Boosting Machine**: Produces an ensemble of decision trees with increasing refined approximations
Deep Neural Networks	• **Deep learning**: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

Algorithms on H2O

Unsupervised Learning

Clustering

- **K-means**: Partitions observations into k clusters/groups of the same spatial size

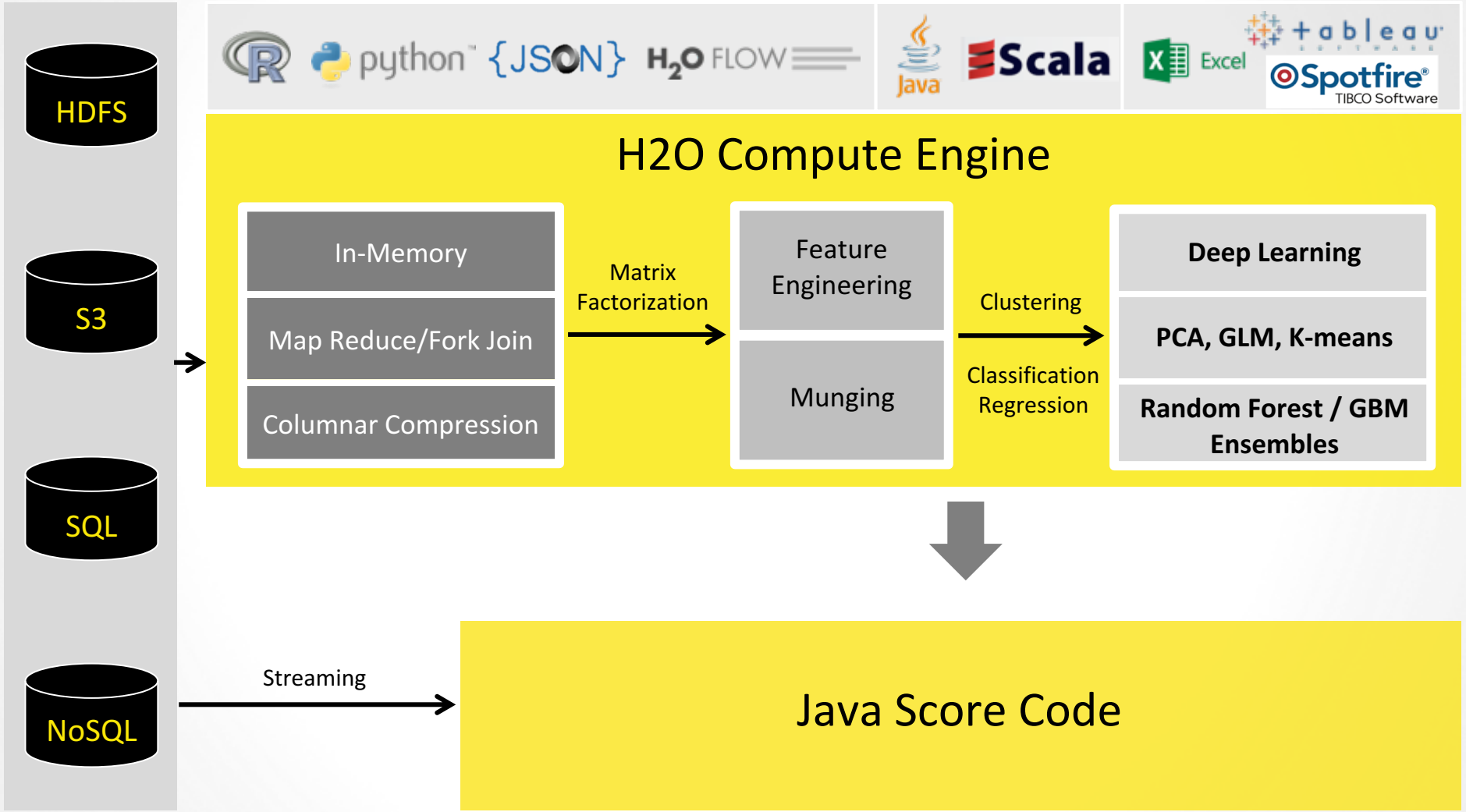
Dimensionality Reduction

- **Principal Component Analysis**: Linearly transforms correlated variables to independent components
- **Generalized Low Rank Models:** extend the idea of PCA to handle arbitrary data consisting of numerical, Boolean, categorical, and missing data

Anomaly Detection

- **Autoencoders**: Find outliers using a nonlinear dimensionality reduction using deep learning

Data and Client Agnostic



R, Python and Flow