PGE 383 Subsurface Machine Learning

Lecture 5c: Feature Transformations

Lecture outline:

- Feature Transformations
- Feature Engineering
- Feature Transformations Hands-on

Motivation for Feature Transformations

There are many reasons that we may need to perform feature transformations.

For example, a method requires the feature to have a specific range or distribution

- principal components analysis requires equivalent variance between features
- artificial neural networks may require all features to range from [-1,1]
- partial correlation coefficients require a bivariate Gaussian distribution, so we apply univariate Gaussian transformation to approximate / improve.
- statistical tests may require a specific distribution, Student's T and χ^2 distributions assumes sampling from Gaussian distributions.
- geostatistical sequential simulation requires an indicator or Gaussian transform
- treat outliers, e.g., Gaussian and rank transforms move outliers inwards

Motivation for Feature Transformations

There are many reasons that we may need to perform feature transformations.

Other practical reasons include,

- provide features consistent for visualization and comparison (violin plots)
- for consistency with statistical assumptions and theory (linearity and homoscedasticity)
- to avoid bias or impose feature weighting for methods (k nearest neighbours regression) that rely on distances calculated in predictor feature space
- deal with outliers and noisy data, and match theoretical outcomes

PGE 383 Subsurface Machine Learning

Lecture 5c: Feature Transformations

Lecture outline:

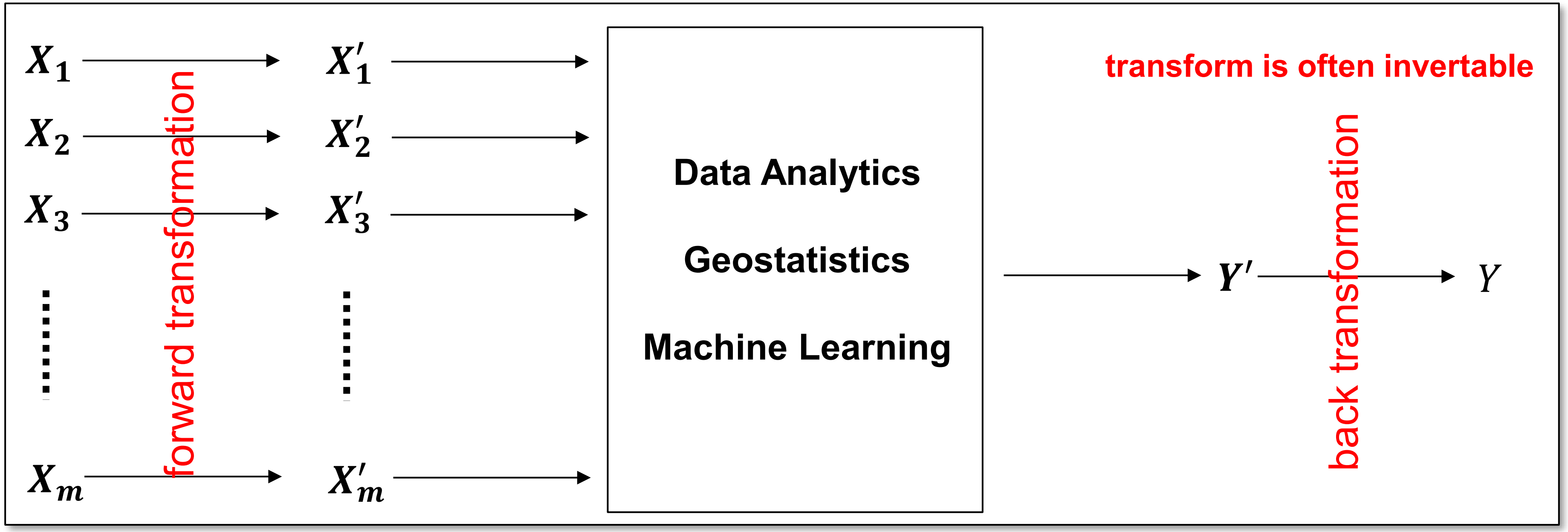
- **Feature Transformations**

Feature Transformations

The application of a transformation applied to the feature

$$x'_\alpha = f(x_\alpha)$$ **deterministic mathematical function**

- May be applied to a predictor feature prior to input into a predictive model
- May be applied to a response feature output from a predictive model
- May be applied to any feature to improve an inferential or predictive workflow
- Could be just applied to improve data visualization



An illustration of feature transformations to support data analytics, geostatistics and machine learning.

Feature Transformations

Working in transformed space:

Forward Transform

$$x'_\alpha = f(x_\alpha)$$

transform: f; transformed: x'_α; original: x_α

Back Transform

$$x_\alpha = f^{-1}(x'_\alpha)$$

inverse transform: f^{-1}; original: x_α; transformed: x'_α

Original Feature Space $\xrightarrow{f(x_\alpha)}$ **Transformed Features Space** $\xrightarrow{f^{-1}(x'_\alpha)}$ **Original Feature Space**

Data — Modeling — Inference Prediction

Schematic of the common workflow, transform from original feature space to transformed space, conduct modeling steps and back transform to original space.

Feature Transformations

Feature Engineering

Using domain expertise to extract improved predictor or response features from raw data,

- improve the performance, accuracy and convergency, of inferential or predictive machine learning
- improve model interpretability (or may worsen interpretability if our engineered features are in unfamiliar units)
- mitigate outliers & bias, consistency with assumptions such as Gaussianity, linearization, dimensional expansion

Feature transformation and feature selection are two forms of feature engineering.

Feature Transformations

We start with very simple transformation and move to more complicated ones

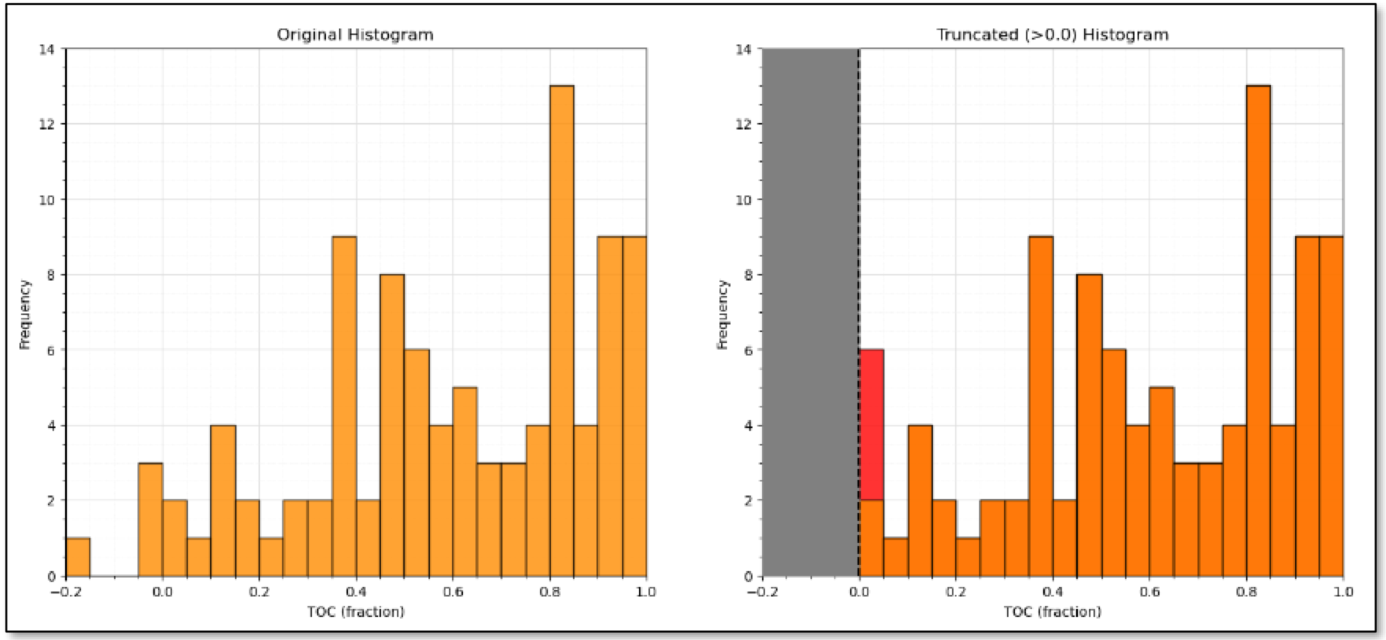
- In general, this topic is not complicated and may not be super very interesting!
- But, feature transformations are common in many data analytics and machine learning workflows
- You'll learn what they are and how to do them in Python with standard packages

Feature Truncation

Feature Truncation

Due to measurement error or imprecision of methods and workflows, it is possible to have feature values that are implausible

- e.g. negative porosity, percentages outside of [0%, 100%] etc.
- compositional data like mineral grades that are positive and sum to 100%
- We may also have outliers that exceed the range of the majority of the data set
- Truncation is the following operations:



Feature truncation at 0.0 for TOC feature, from MachineLearning_feature_transformations chapter of e-book

$$x'_\alpha = \min(x_\alpha, x_t)$$

set the sample value to a threshold if less than the threshold

$$x'_\alpha = max(x_\alpha, x_t)$$

set the sample value to a threshold if greater than the threshold

Affine Correction

The affine correction is the transform of the feature distribution to a new mean and variance.

- this is a shift and stretch / squeeze of the original property distribution
- assumes no shape change, rank preserving

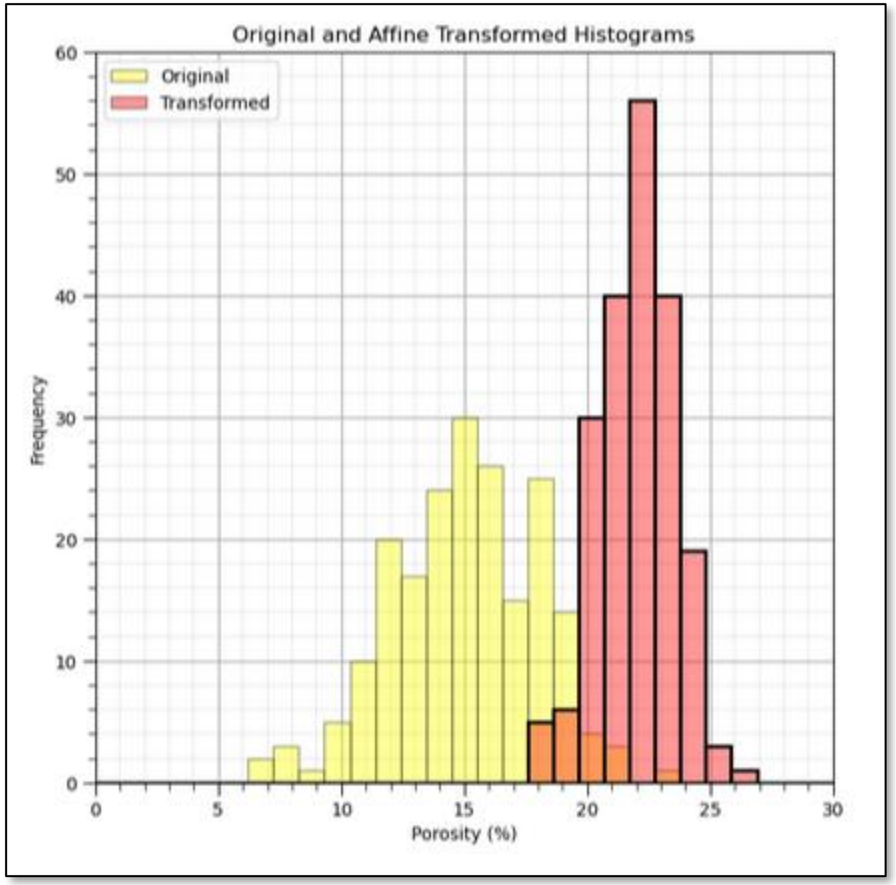
$$x'_\alpha = \frac{\sigma_{x'_\alpha}}{\sigma_{x_\alpha}} \cdot (x_\alpha - \overline{x_\alpha}) + \overline{x'_\alpha}$$

where,

- σ_{x_α} and $\sigma_{x'_\alpha}$ are the original and target standard deviation
- $\overline{x_\alpha}$ and $\overline{x'_\alpha}$ are the original and target mean

Examples,

- apply debiased feature distributions
- apply bootstrap scenarios, low, mid high distributions



Affine correction, from MachineLearning_feature_transformations chapter of e-book

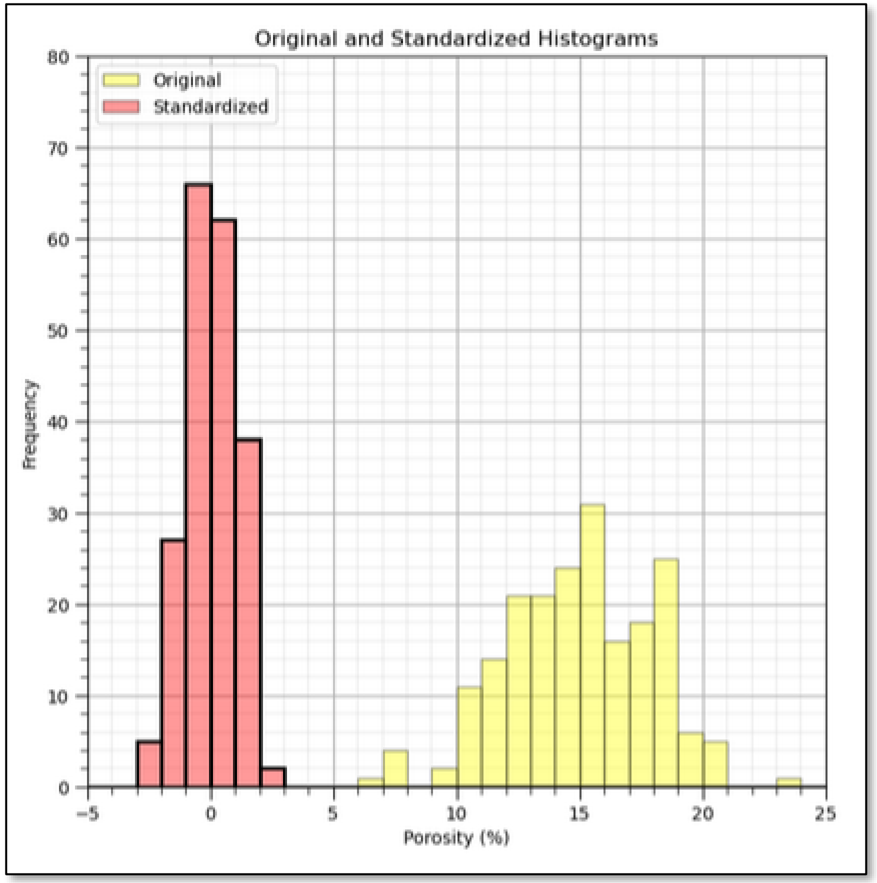
Standardization

Standardization is the transform of the feature distribution to a mean of 0 and variance of 1.

- this is a shift and stretch / squeeze of the original property distribution
- assumes no shape change, rank preserving
- specific case of the affine correction

$$x'_\alpha = \frac{\cancelto{1}{\sigma_{x'_\alpha}}}{\sigma_{x_\alpha}} \cdot (x_\alpha - \overline{x_\alpha}) + \cancelto{0}{\overline{x'_\alpha}}$$

$$x'_\alpha = \frac{1}{\sigma_{x_\alpha}} \cdot (x_\alpha - \overline{x_\alpha})$$



Standardization, from MachineLearning_feature_transformations chapter of e-book

Normalization
Min / Max Transform

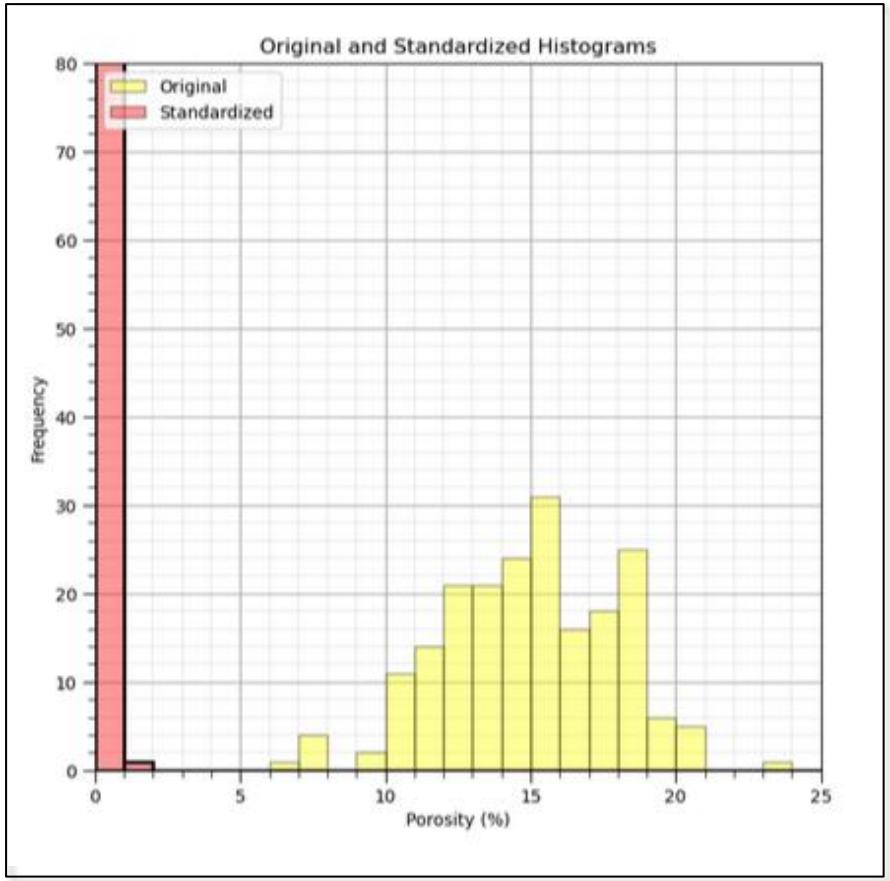
Normalization is the transform of the feature distribution to a min of 0 and max of 1 (sometimes -1 to +1), also known as min-max normalization.

- this is a shift and stretch / squeeze of the original property distribution
- assumes no shape change, rank preserving

$$x'_\alpha = \frac{x_\alpha - \min(x_\alpha)}{max(x_\alpha) - \min(x_\alpha)}$$

Methods that require standardization and min/max normalization:

- k-means clustering, k-nearest neighbour regression
- β coefficient's for feature ranking
- standardized variograms
- artificial neural networks forward transform of predictor features and back transform of response features to improve activiation function sensitivity



Normalization, from MachineLearning_feature_transformations chapter of e-book

Example of normalization of a feature distribution.

L1/L2 Normalizer

L1 / L2 Normalizer is performed across features over individual samples to constrain the sum

- The L1 Norm has the following constraint across samples

$$\sum_{\alpha=1}^{m} x'_{i,\alpha} = 1.0, \qquad i = 1, \ldots, n$$

- The L1 normalizer transform:

$$x'_{i,\alpha} = \frac{x_{i,\alpha}}{\sum_{\alpha=1}^{m} x_{i,\alpha}}$$

- The L2 Norm has the following constraint across samples

$$\sum_{\alpha=1}^{m} \left(x'_{i,\alpha}\right)^2 = 1.0, \qquad i = 1, \ldots, n$$

- The L2 normalizer transform:

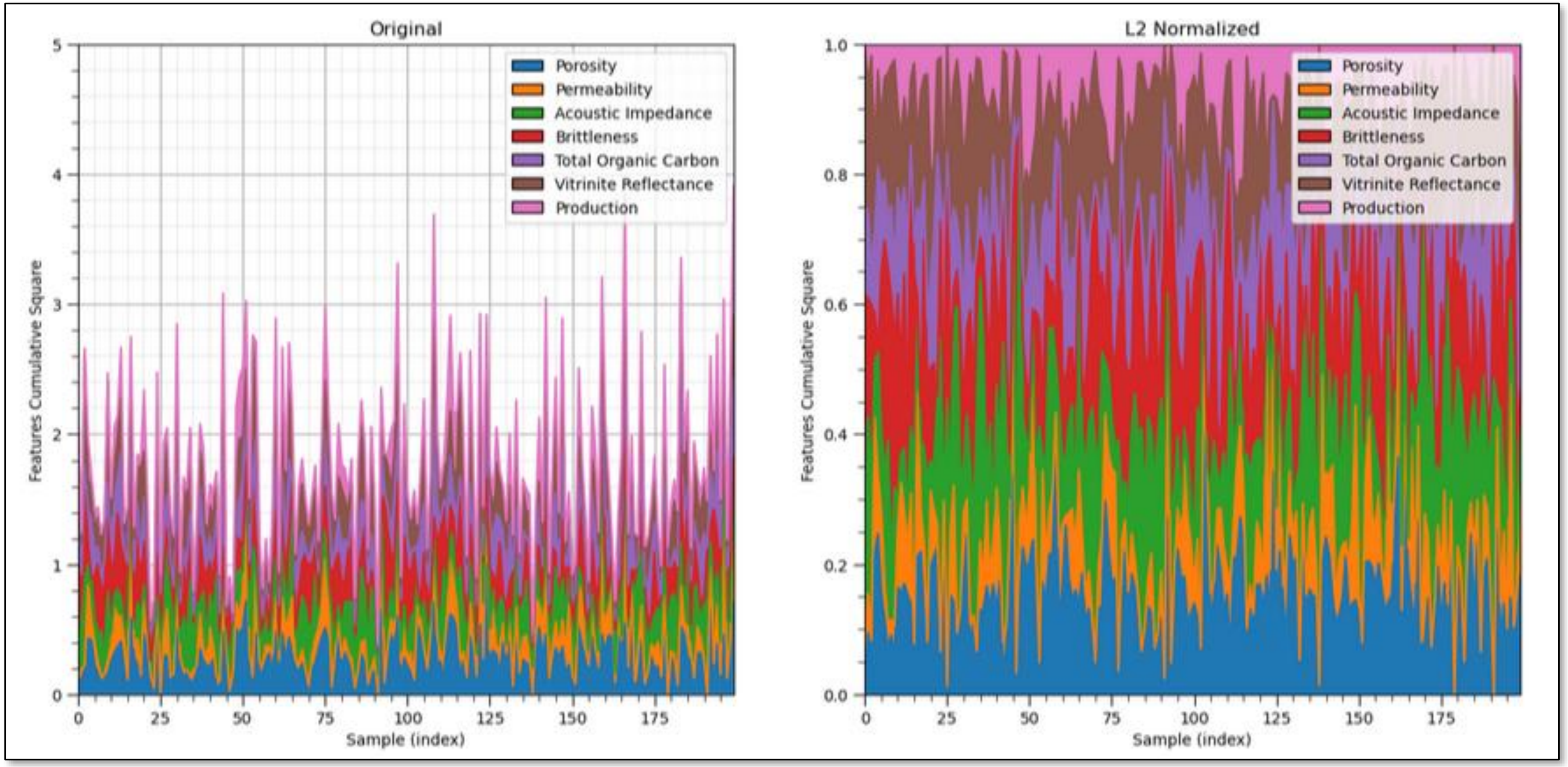
$$x'_{i,\alpha} = \sqrt{\frac{\left(x_{i,\alpha}\right)^2}{\sum_{\alpha=1}^{m} \left(x_{i,\alpha}\right)^2}}$$

Example, applied in text classification and clustering, and L1 for compositional data (sum 1.0 constraint)

L1/L2 Normalizer

L1 / L2 Normalizer is performed across features over individual samples to constrain the sum

- Example of L2 normalizer



L2 normalizer, from MachineLearning_feature_transformations chapter of e-book.

Binary / Indicator Transform

Indicator coding is transforming a feature to a probability relative to a category or a threshold.

If $I\{\mathbf{u}: z_k\}$ is an indicator for a categorical variable,

- What is the probability of a realization equal to a category?

$$I(\mathbf{u};\boldsymbol{z_k}) = \begin{cases} 1, & \text{if } Z(\mathbf{u}) \;=\boldsymbol{z_k} \\ 0, & \text{otherwise} \end{cases}$$

- e.g. given threshold, $z_2 = 2$, and data at $\mathbf{u}_1, z(\mathbf{u}_1) = 2$, then $I\{\mathbf{u}_1; z_2\} = 1$
- e.g. given threshold, $z_1 = 1$, and a RV away from data, $\mathrm{Z}(\mathbf{u}_2)$ then $I\{\mathbf{u}_2; z_1\} = 0.25$

If $I\{\mathbf{u}: z_k\}$ is an indicator for a continuous variable,

- What is the probability of a realization less than or equal to a threshold?

$$I(\mathbf{u};\boldsymbol{z_k}) = \begin{cases} 1, & \text{if } Z(\mathbf{u}) \;\leq \boldsymbol{z_k} \\ 0, & \text{otherwise} \end{cases}$$

- e.g. given threshold, $z_1 = 6\%$, and data at $\mathbf{u}_1, z(\mathbf{u}_1) = 8\%$, then $I\{\mathbf{u}_1; z_1\} = 0$
- e.g. given threshold, $z_4 = 18\%$, and a RV,$\mathrm{Z}(\mathbf{u}_2) = N[16\%, 3\%]$ then $I\{\mathbf{u}_1; z_k\} = 0.75$

Binary / Indicator Transform

Example of indicator transforms for a categorical variable

Original Data	$I\{\mathbf{u}_\alpha; z_1 = 1\}$	$I\{\mathbf{u}_\alpha; z_2 = 2\}$	$I\{\mathbf{u}_\alpha; z_3 = 3\}$
$z(\mathbf{u}_1) = 3$	0	0	1
$z(\mathbf{u}_2) = 1$	1	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$z(\mathbf{u}_n) = 2$	0	1	0

Example of indicator transform of a categorical feature.

Our $\mathrm{z}(\mathbf{u}_\alpha)$, $\alpha = 1, \ldots, n$, data become $k = 1, \ldots, K$ sets of n data, a new variable that indicates the probability of being each category.

- This indicator transform of a categorical features is also known as **one-hot encoding**

Binary / Indicator Transform

Example of indicator transforms for a continuous variable.

Original Data	$I\{\mathbf{u}_\alpha; z_1 \leq 5\%\}$	$I\{\mathbf{u}_\alpha; z_2 \leq 10\%\}$	$I\{\mathbf{u}_\alpha; z_3 \leq 15\%\}$
$z(\mathbf{u}_1) = 12\%$	0	0	1
$z(\mathbf{u}_2) = 4\%$	1	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$z(\mathbf{u}_n) = 17\%$	0	0	0

Example of indicator transform of a continuous feature.

Our $\mathrm{z}(\mathbf{u}_\alpha)$, $\alpha = 1, \ldots, n$, data become $k = 1, \ldots, K$ sets of n data, a new variable that indicates the probability of being less than or equal to each threshold.

- Encoding is based on assigned thresholds, 5%, 10%, and 15%.

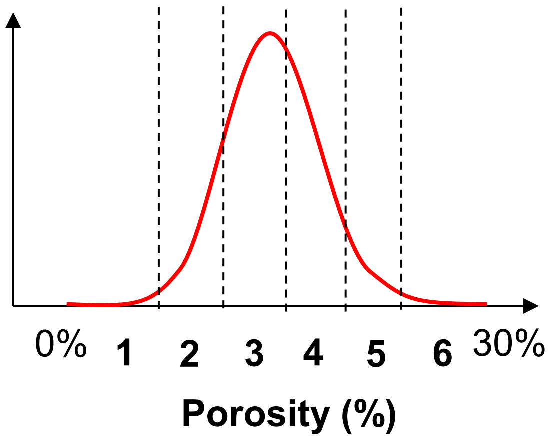
Binary / Indicator Transform

Methods that require binary / indicator transform,

- indicator variograms, indicator kriging and indicator simulation
- indicator maps
- environmental and economics thresholds and modeling probabilities of occurance
- artificial neural networks with categorical predictor or response features

K Bins Discretization

- Bin the range of the feature into K bins
- Then for each sample assignment of a value of 1 if the sample is within a bin and 0 if outsize the bin
  - strategies include uniform width bins (uniform) and uniform number of data in each bin (quantile)

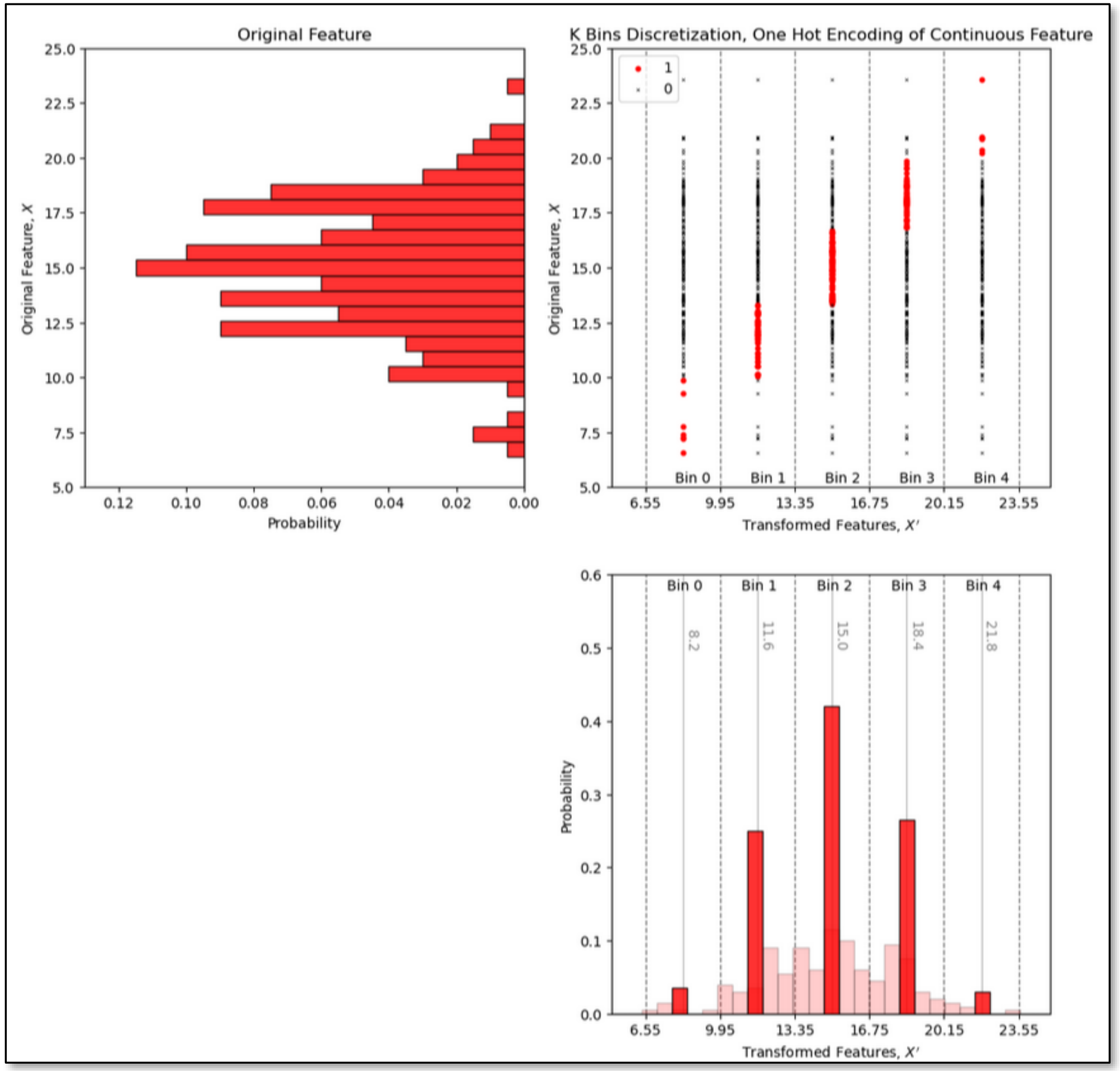


x_α	x_α^1	x_α^2	x_α^3	x_α^4	x_α^5	x_α^6
2%	1	0	0	0	0	0
16%	0	0	0	1	0	0
26%	0	0	0	0	0	1
8%	0	1	0	0	0	0

Simple example of K bins discretization

K Bins Discretization

Example of K bins discretization, one hot encoding

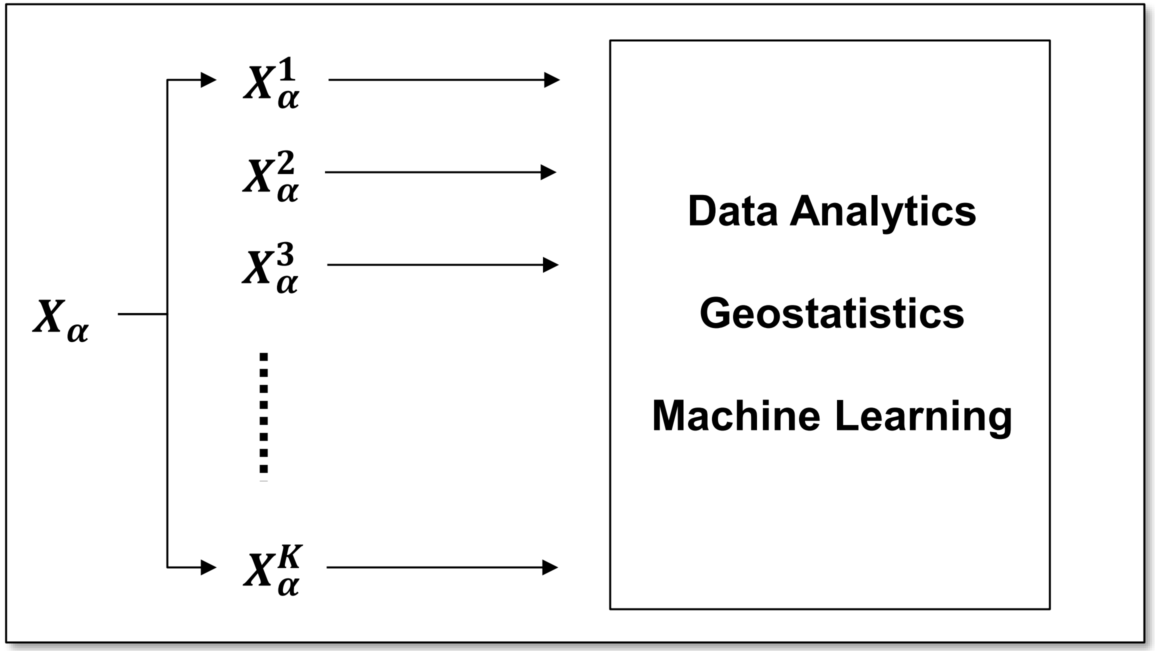


K bins discretization, from MachineLearning_feature_transformations chapter of e-book

K Bins Discretization

K bins discretation for Soft Data Encoding

- A probability coding, probability of the sample existing in each bin, could integrate sample uncertainty, soft data
- A form of basis expansion (more during support vector machines)



Probability coding / basis expansion

K Bins Discretization

Methods that require K bins discretization:

- basis expansion to work in a higher dimensional space
- discretiation of continuous features to categorical features for categorical methods such as naïve Bayes classifier
- histogram construction and Chi-square test for difference in distributions
- mutual information binning

Gaussian Anamorphosis

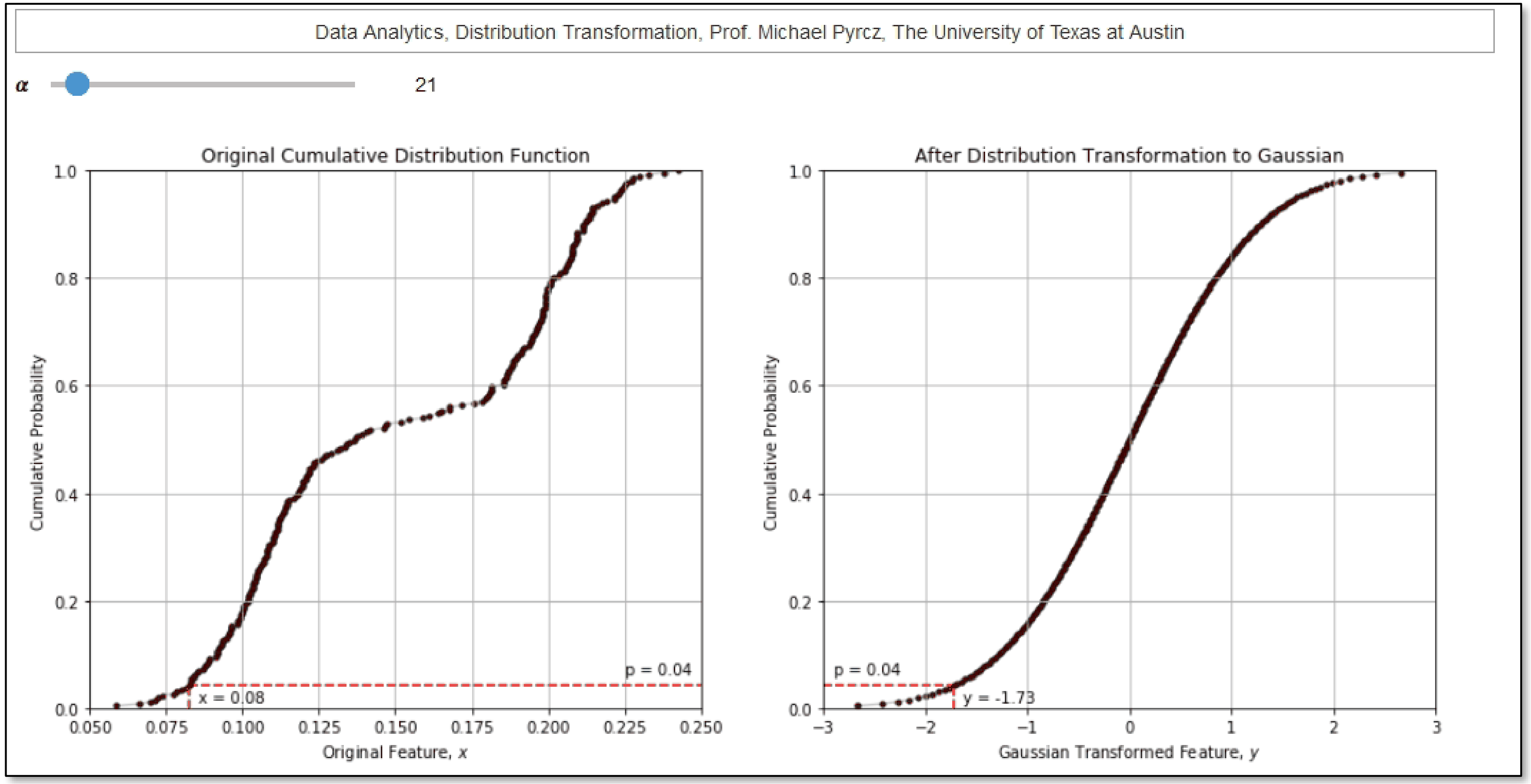
Quantile transformation to a Gaussian distribution.

- Mapping feature values through their cumulative probabilities.

$$y = G_y^{-1}\left(F_x(x)\right)$$

- where F_x is the original feature cumulative distribution function (CDF) and G_y is the Gaussian CDF
- Gaussian probability density function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$



Gaussian anamorphosis is a specific case of distribution quantile transformation, from Interactive_Distribution_Transformation.ipynb.

- Gaussian CDF

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} exp\left[-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right] dy$$

Unbounded $-\infty < x < +\infty$

Gaussian Anamorphosis

More on the Gaussian distribution

- shorthand for a normal distribution is

$$N(\mu, \sigma^2), \text{ for example } N[0,1] \text{ is standard normal}$$

- much of “natural variation” / measurement error is Gaussian

- parameterized fully by mean, variance and correlation coefficient (if multivariate)

- distribution is unbounded, no min nor max, extremes are very unlikely, some type of truncation is often applied

- Warning, many workflows apply univariate Gaussian anamorphosis and then assume bivariate or multivariate Gaussian, this is not correct, but it is generally too difficult to transform our data to multivariance Gaussian.

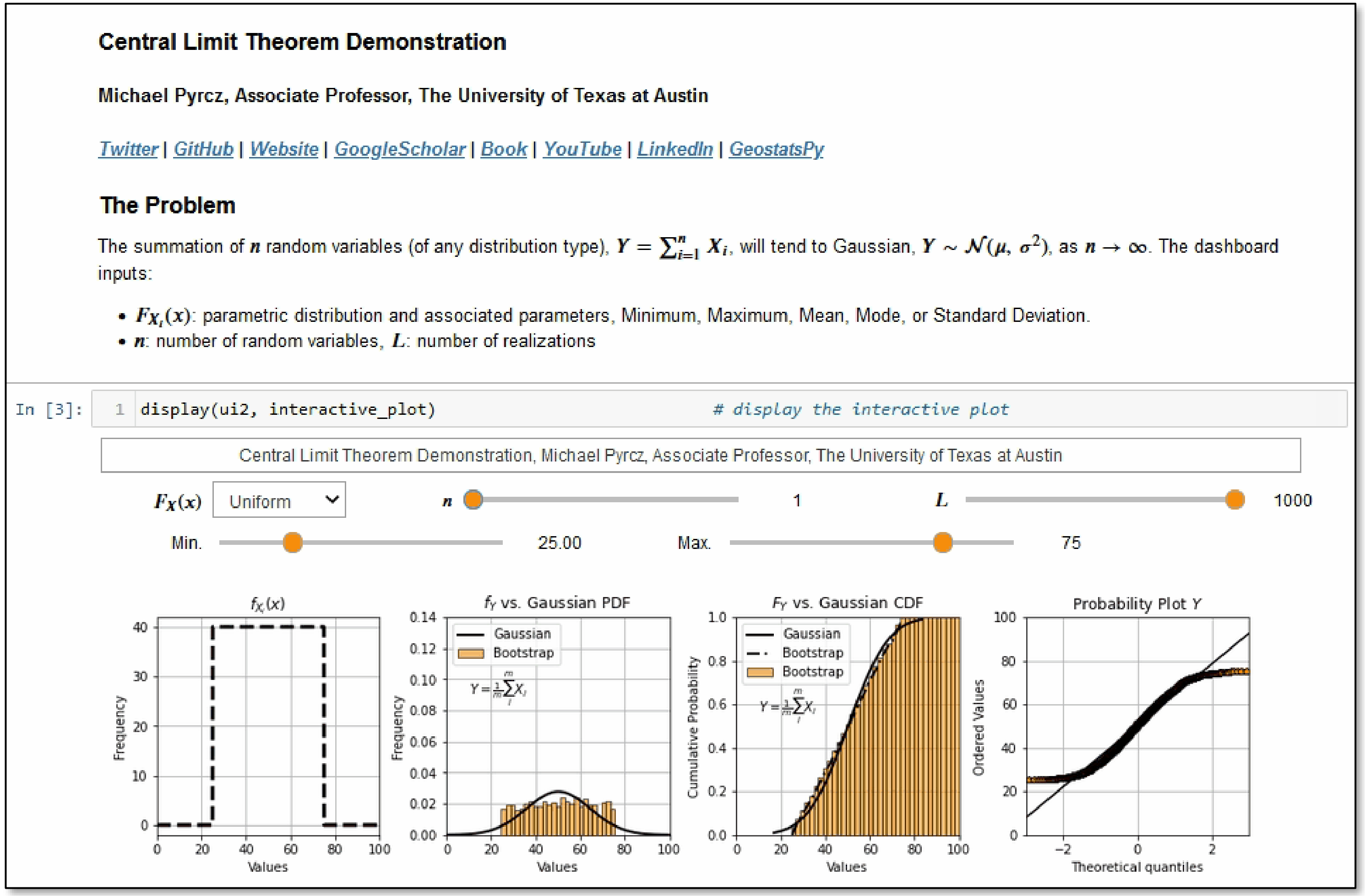
Gaussian Anamorphosis

Central Limit Theorem

The summation / average of multiple random variables tends towards a Gaussian distributed.

- this occurs quickly with 3-4 independent variables

- some reservoir properties may be Gaussian distributed (e.g. porosity is the average of pore space vs. grains over smaller volumes.



Interactive central limit theorem demonstration, from Interactive_Central_Limit_Theorem.ipynb.

Gaussian Anamorphosis

The multivariate Gaussian distribution,

$$f_X(x_1, \ldots, x_m) = \frac{exp\left(-{}^{1}/_{2}\,(x-\mu)^T \Sigma^{-1}(x-\mu)\right)}{\sqrt{(2\pi)^m |\Sigma|}}$$

where $\boldsymbol{\mu}$ is the m vector of means and $\boldsymbol{\Sigma}$ is the $m \times m$ matrix of all pairwise covariances.

$$\mu = [\mu_1, \ldots, \mu_1] \qquad \Sigma = \begin{bmatrix} \sigma_1^2 & \cdots & C_{1,m} \\ \vdots & \ddots & \vdots \\ C_{m,1} & \cdots & \sigma_m^2 \end{bmatrix}$$

A very compact parameterization:

$$m + \frac{m(m+1)}{2}$$

including m means, m variances and $\frac{m(m-1)}{2}$ unique covariances (covariance matrix, $\boldsymbol{\Sigma}$, is symmetric)

Gaussian Anamorphosis

The multivariate Gaussian distribution

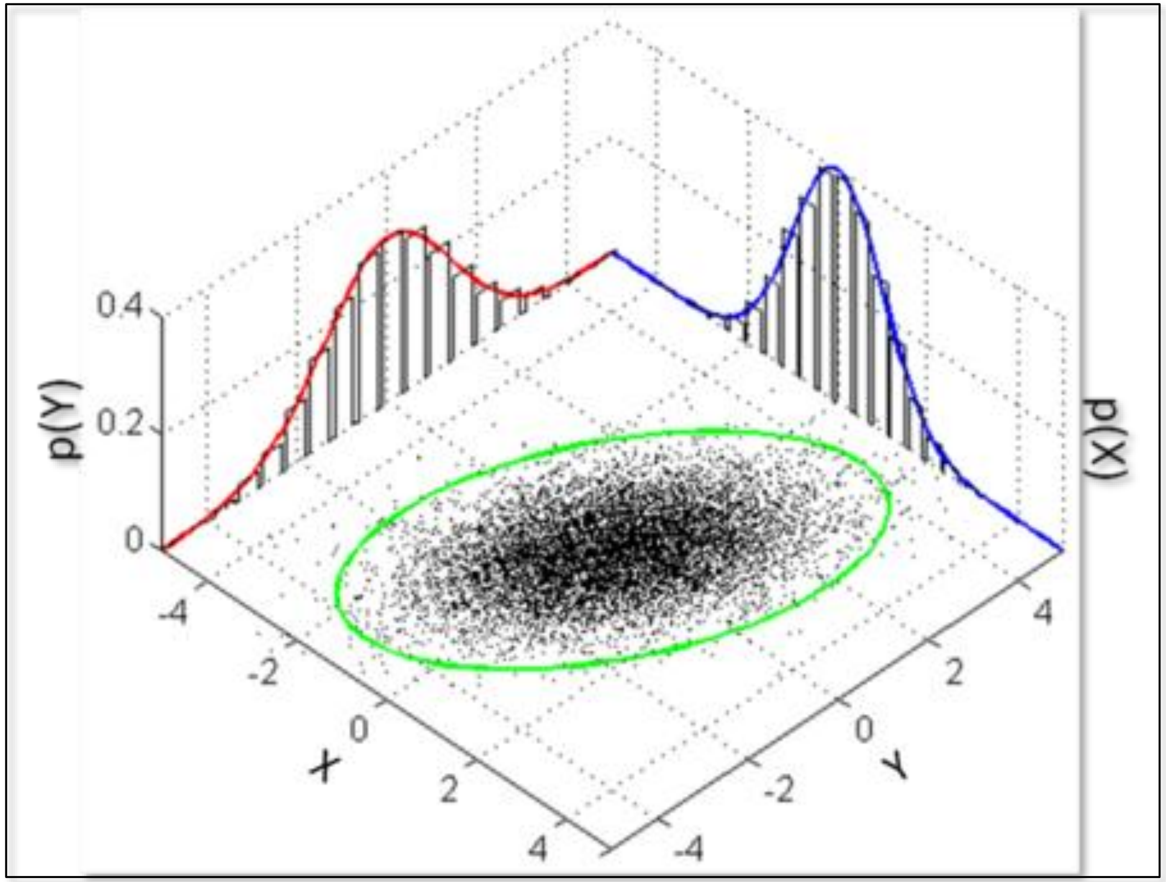
All marginal distributions are Gaussian,

$$f_{X_1}(x_1) \sim N(\mu_1, \sigma_1^2)$$

$$f_{X_2}(x_2) \sim N(\mu_2, \sigma_2^2)$$

$$\vdots$$

$$f_{X_m}(x_m) \sim N(\mu_m, \sigma_m^2)$$



Gaussian joint and marginal distributions, image from https://en.wikipedia.org/wiki/Multivariate_normal_distribution#/media/File:MultivariateNormal.png.

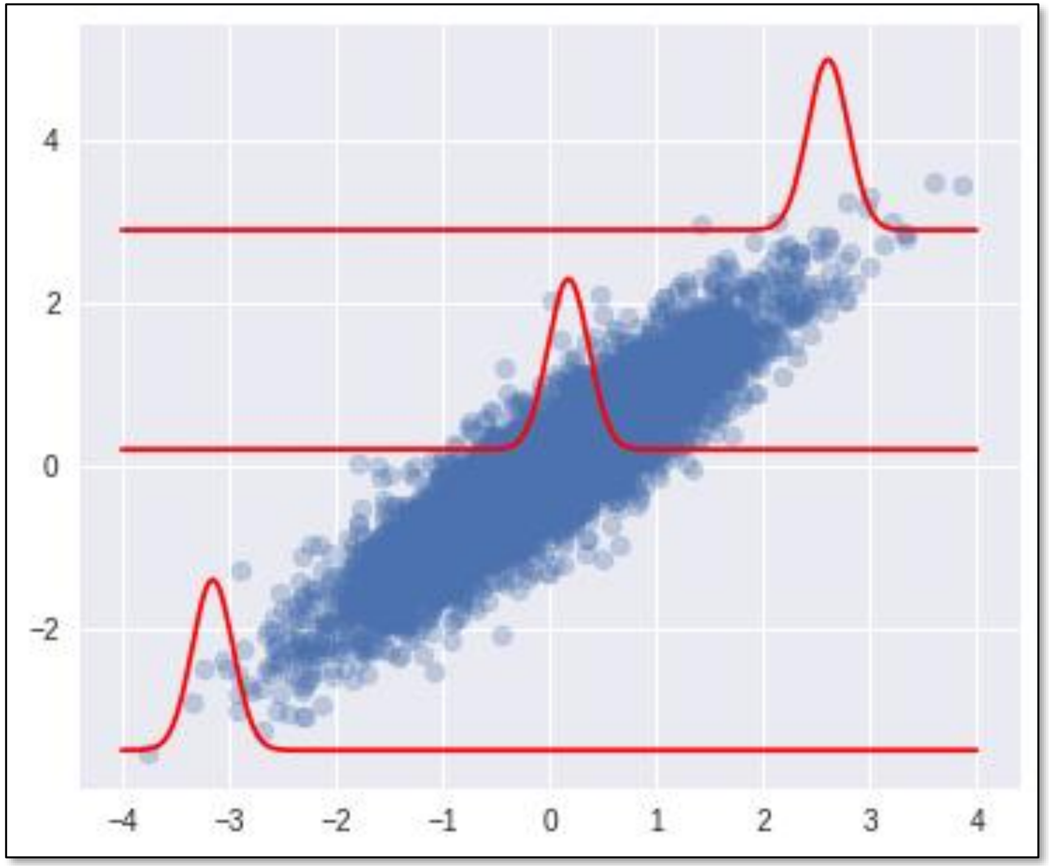
Gaussian Anamorphosis

The multivariate Gaussian distribution

All conditional distributions are Gaussian, here's the bivariate case,

$$f_{X_1|X_2}(x_1 \,|X_2 = x_2) \sim N\left(\mu_1 + \frac{\sigma_1}{\sigma_2}\rho(x_2 - \mu_2), (1-\rho^2)\sigma_1^2\right)$$

- the conditional variance is homoscedastic, does not depend on the mean!



Bivariate Gaussian conditional distributions, image from https://stats.stackexchange.com/questions/385823/variance-of-conditional-multivariate-gaussian.

Gaussian Anamorphosis

Methods that require a Gaussian distribution,

- Pearson product-moment correlation coefficients completely characterize multivariate relationships when data are multivariate Gaussian
- partial correlations require bivariate Gaussian
- sequential simulation (geostatistics) assumes Gaussian to reproduce the global distribution
- Student's t test for difference in means
- Chi-square distributions is derived from sum of squares of Gaussian distributed random variables
- Gaussian naïve Bayes classification assumes Gaussian conditionals

Uniform Distribution Transform

Quantile transformation to a uniform distribution (e.g cumulative probabilies)

- Mapping feature values through their cumulative probabilities.

$$y = F_y^{-1}\left(F_x(x)\right)$$

- where F_x is the original feature cumulative distribution function (CDF) and F_y is the uniform CDF

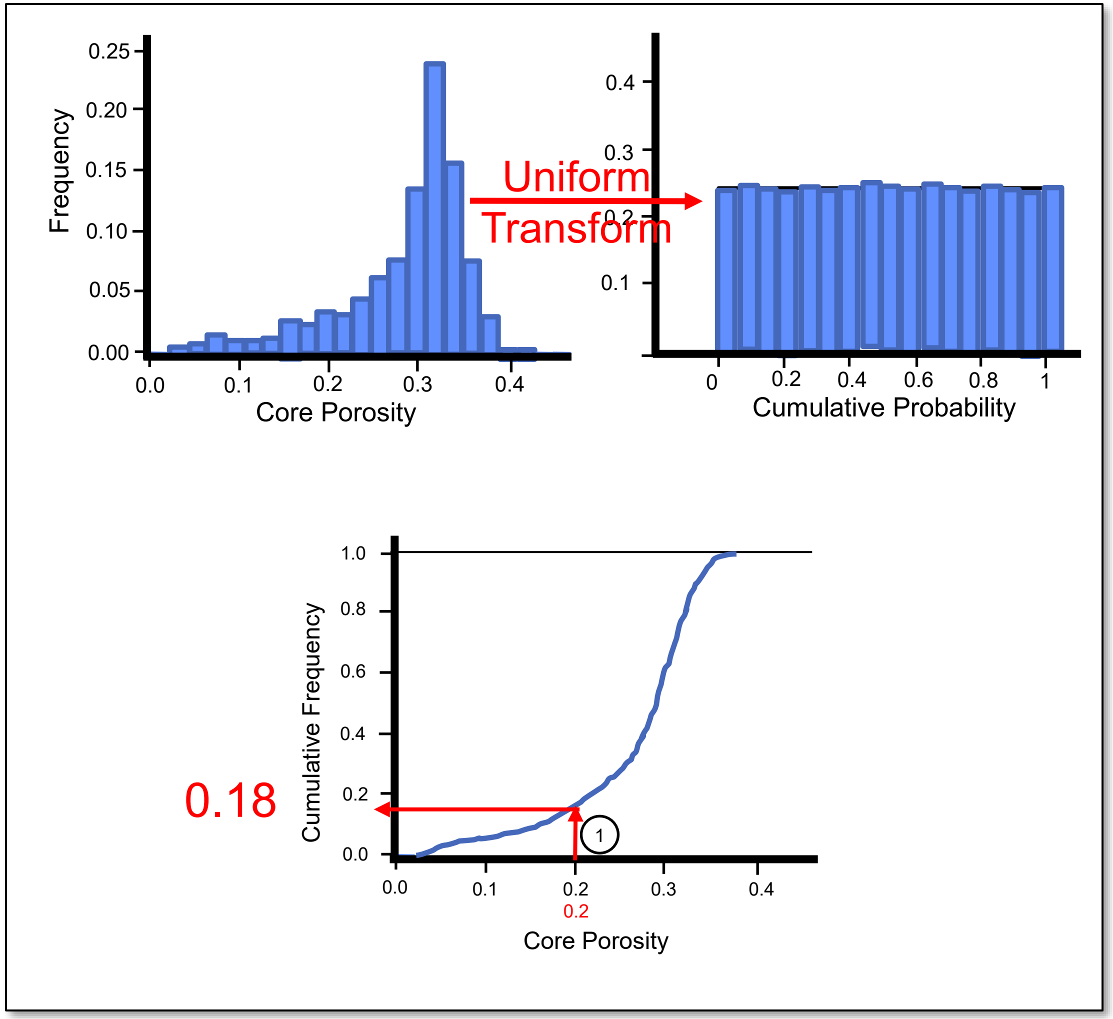
In a specific case we stop short with at the cumulative probabilities

$$p = F_x(x)$$

Uniform probability density function

$$f_x(x) = \frac{1}{N} = constant$$

and uniform CDF $F_x(x) = \frac{1}{N} x$ bounded $x_{min} < x < x_{min}$



Uniform distribution transformation, specifically to cumulative probability.

Lecture outline:

- Feature Engineering

Other Feature Engineering

Once again, methods and workflows that take raw data and transform it into predictor features that can be used in machine learning.

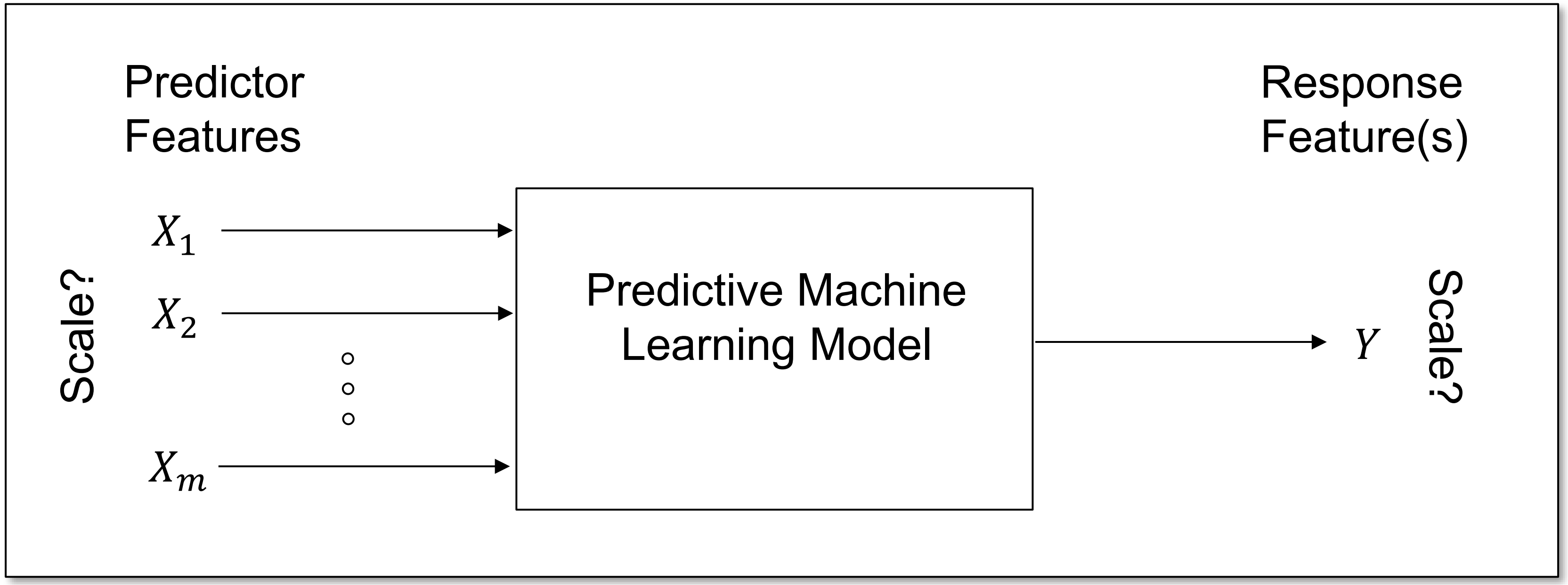
- data debiasing, uncertainty modeling, feature imputation, feature transformation, feature selection and feature projection (dimensionality reduction) are all forms of feature engineering

Let’s now provide a brief overview of other forms of feature engineering that focus on integrating domain expertise.

- scale
- new features / proxy models

Feature Engineering
Scale

All spatial data, the features associated with the machine learning model may have different scale.



Schematic of predictive machine learning.

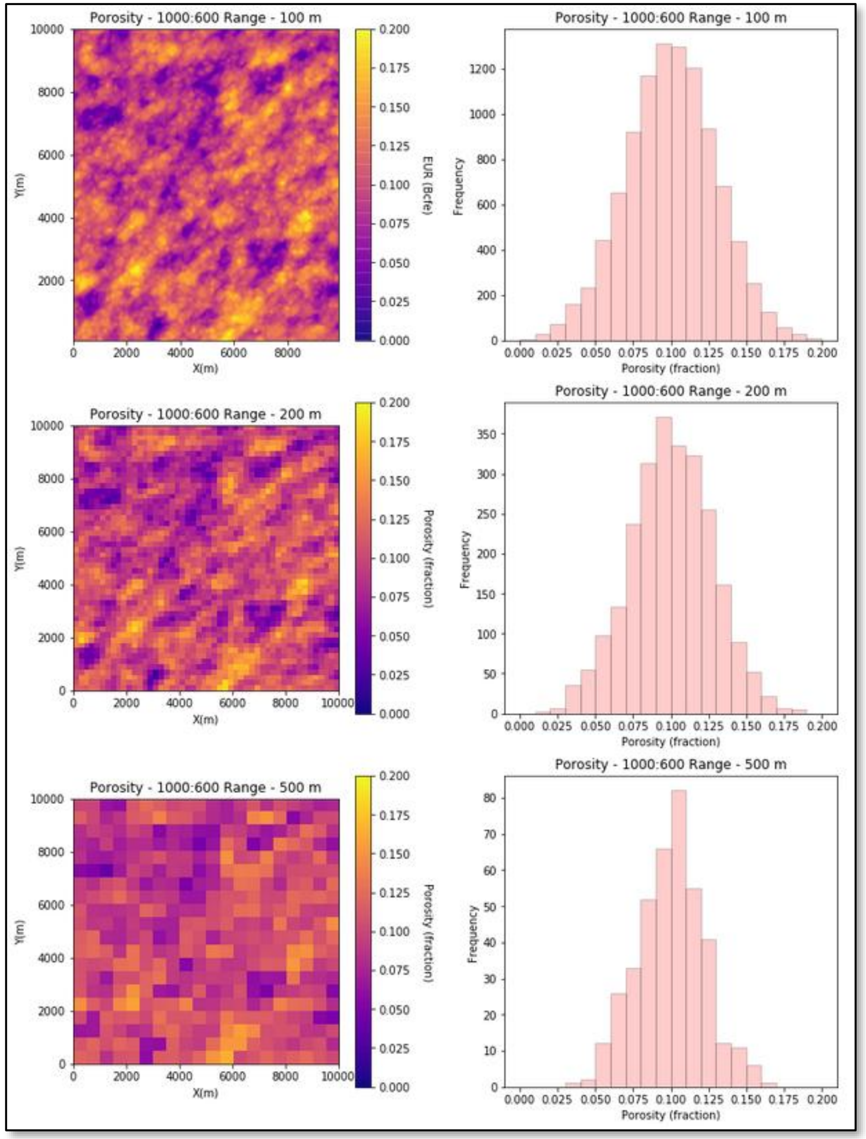
Feature Engineering
Scale

What is the impact of scale?

- all statistics have an implicit scale choice
- therefore so do all statistical learning and machine learning models

Scale impacts the statistics and models.

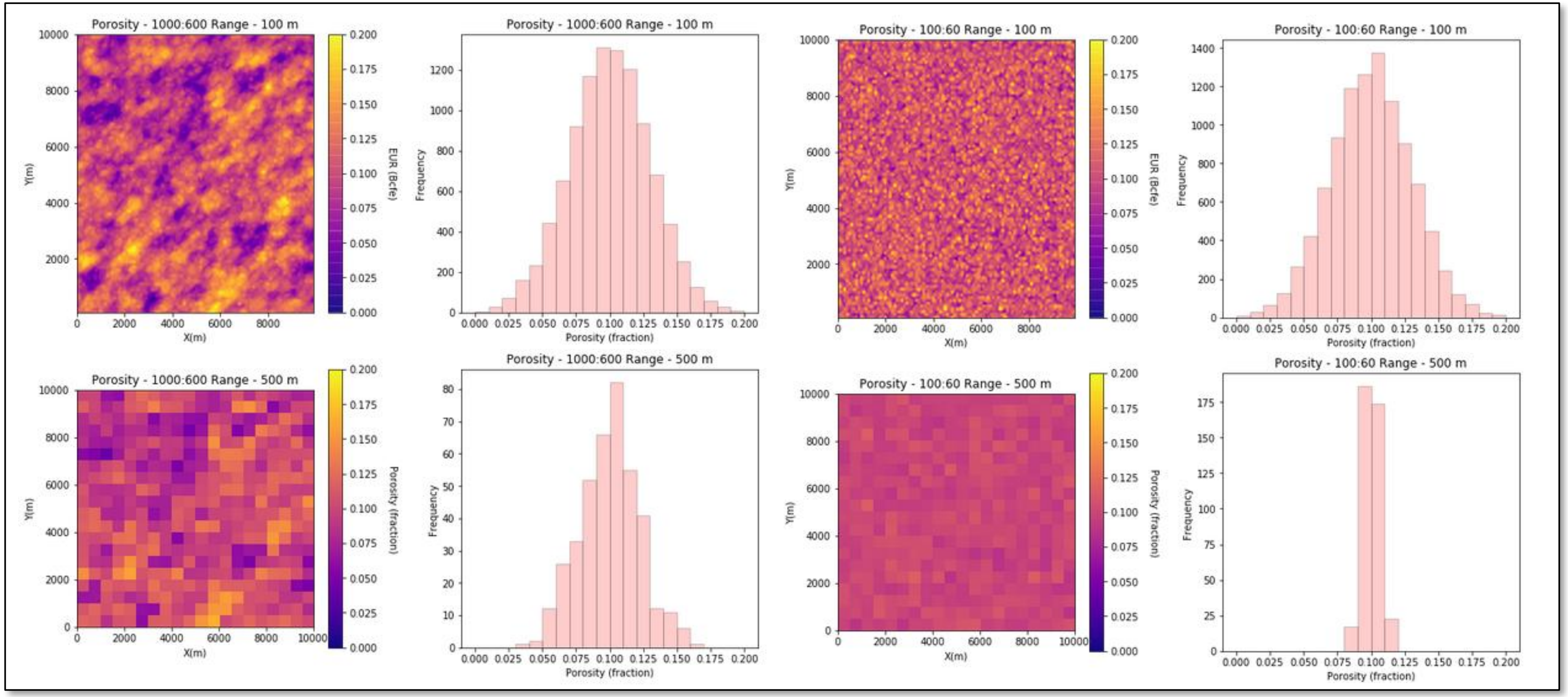
- see the change in variance as we change the scale of this 2D map of porosity.



The impact of scale on the histogram, see change in variance, from Applied Geostatistics in Python e-book, Volume Variance chapter.

Feature Engineering
Scale

The impact of scale on statistics, statistical / machine learning depends on spatial continuity.

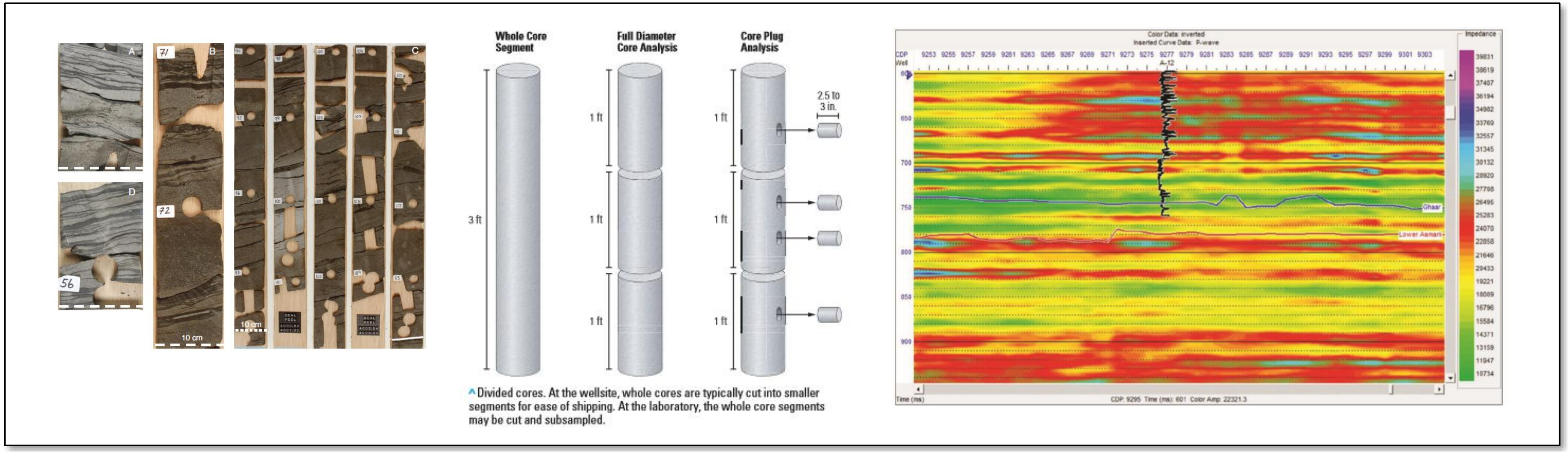


Anisotropic spatial continuity (left) and no spatial continuity (right), observe the difference in impact of change in scale on the histogram, from Applied Geostatistics in Python e-book, Volume Variance chapter.

Feature Engineering
Scale

The data we work with has a wide variety of scales. In big data terms this is part of the ‘data variety’ challenge.

- feature scale is the volume over which a sample is representative.



Well core-based features are at foot scale, while seismic-based features have a scale of 10’s of feet vertically and 100’s of feet horizontally.

Rigorous integration of scale is an ‘unsolved problem’. Issues such as missing scale remain because of data sparcity.

Feature Engineering
Scale

At the very least we need to ensure the dimensionality between the features is consistent.

- well logs are 1D data
- maps are 2D data
- seismic attributes are 3D data

In this case, we need to build our models from features with consistent dimensionality.

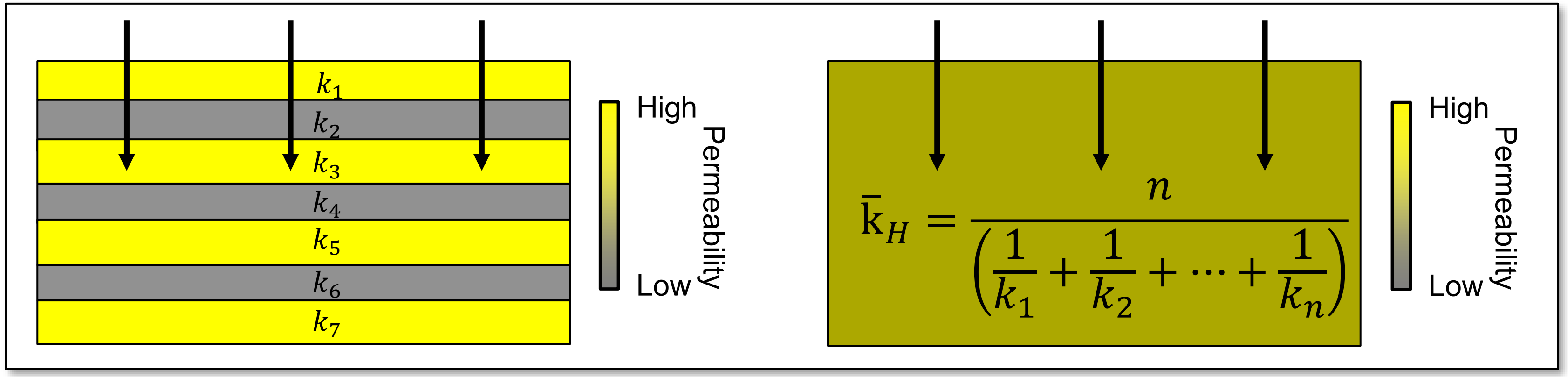
- well logs (1D) → project into a model (3D)
- well logs (1D) → scale up, average over well and post on a map (2D)
- seismic (3D) → scale up, average over columns and post on a map (2D)

Feature Engineering
Scale

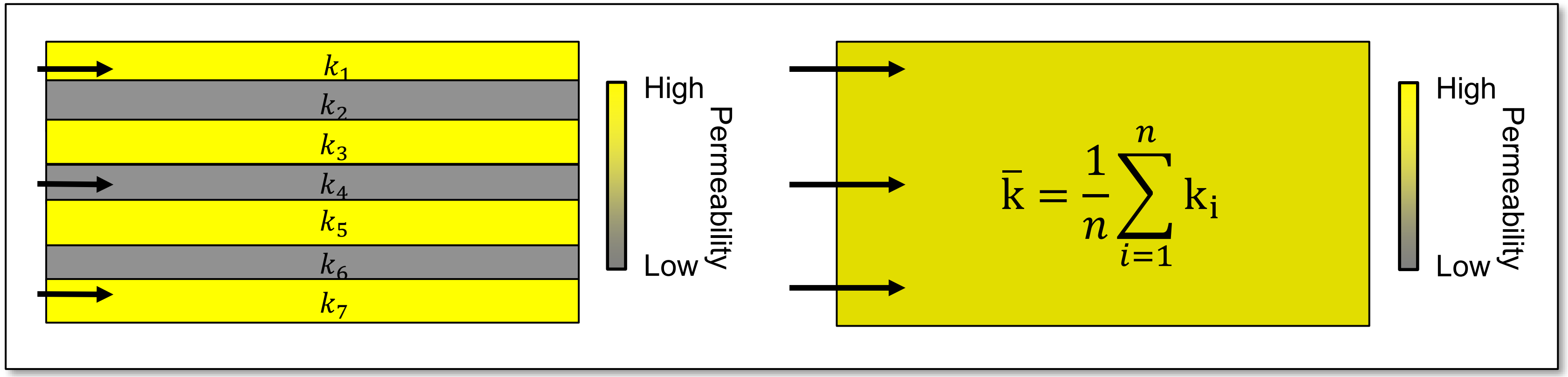
Another Interpretation of Central Tendency is Effective Property

Could I replace all the permeabilities of these layers with a single effective permeability?

- when we apply flow simulation to both models they flow the same!



Harmonic mean is applied to calculate effective permeability for flow across layers, smallest permeabilities have the greatest impact.



Arithmetic mean is applied to calculate effective permeability for flow along layers, extreme permeabilities have the greatest impact.

Feature Engineering
Scale

Another Interpretation of Central Tendency is Effective Property

A more general form is **power law averaging**

$$\overline{x}_P = \left(\frac{1}{n} \sum_{i=1}^{n} x_i^p \right)^{\frac{1}{p}}$$

Power Law Averaging

$$K_{eff} = \left[\frac{1}{v} \int_v k(u)^p du \right]^{\frac{1}{p}}$$

Example of continuous permeability power law upscaling.

Power Law Averaging for Volumetric Scale Up of Permeability

- useful to calculate effective permeability where flow is not parallel nor perpendicular to distinct permeability layers
- flow simulation may be applied to calibrate (calculate the appropriate power for power law averaging)

Power = -1 Harmonic Average

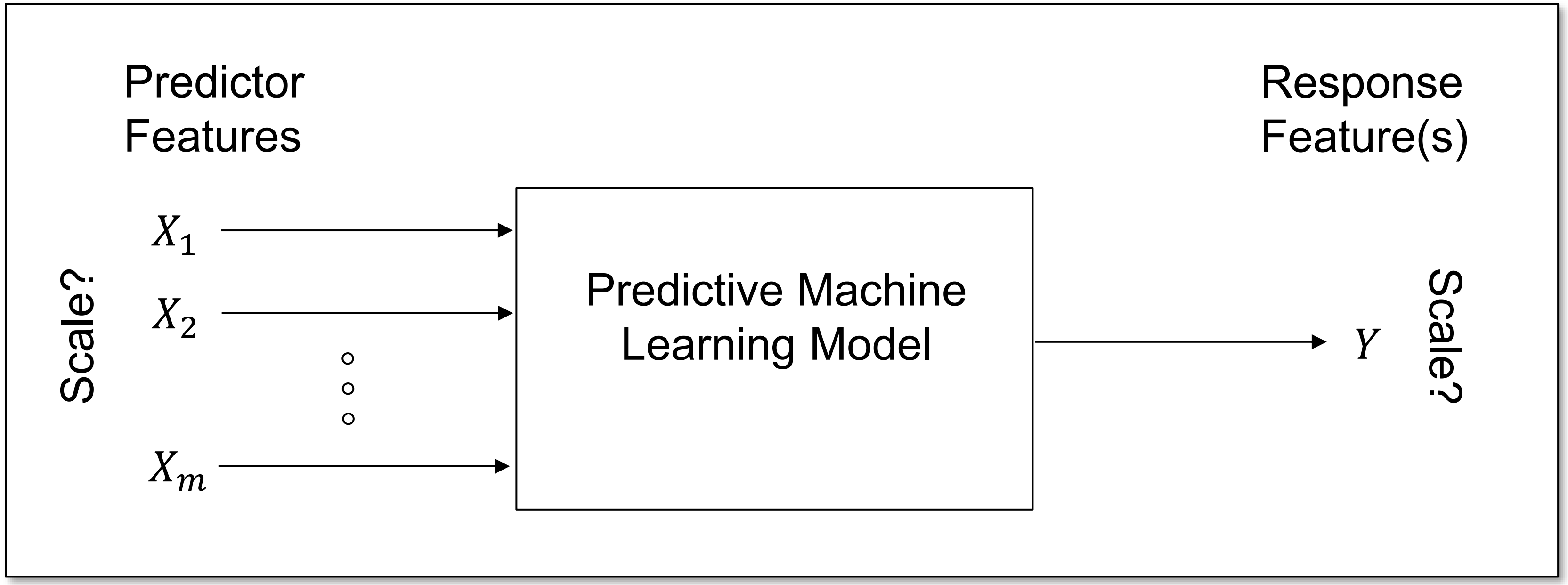
Power = 0 Geometric Average*

Power = 1 Arithmetic Average

* Proof in limit as p→0, see Zanon (2002) on Canvas.

Feature Engineering New Features, Proxy Models

It may be useful to develop our own features based on the raw data features.



Schematic of predictive machine learning.

Examples,

- rock quality index is commonly applied, $^{K_i}/_{\varphi_i}$
- spatial, distance and volumetric calculations, OIP, effective porosity
- truncations such as setting low permeability rock to 0 porosity

PGE 383 Subsurface Machine Learning

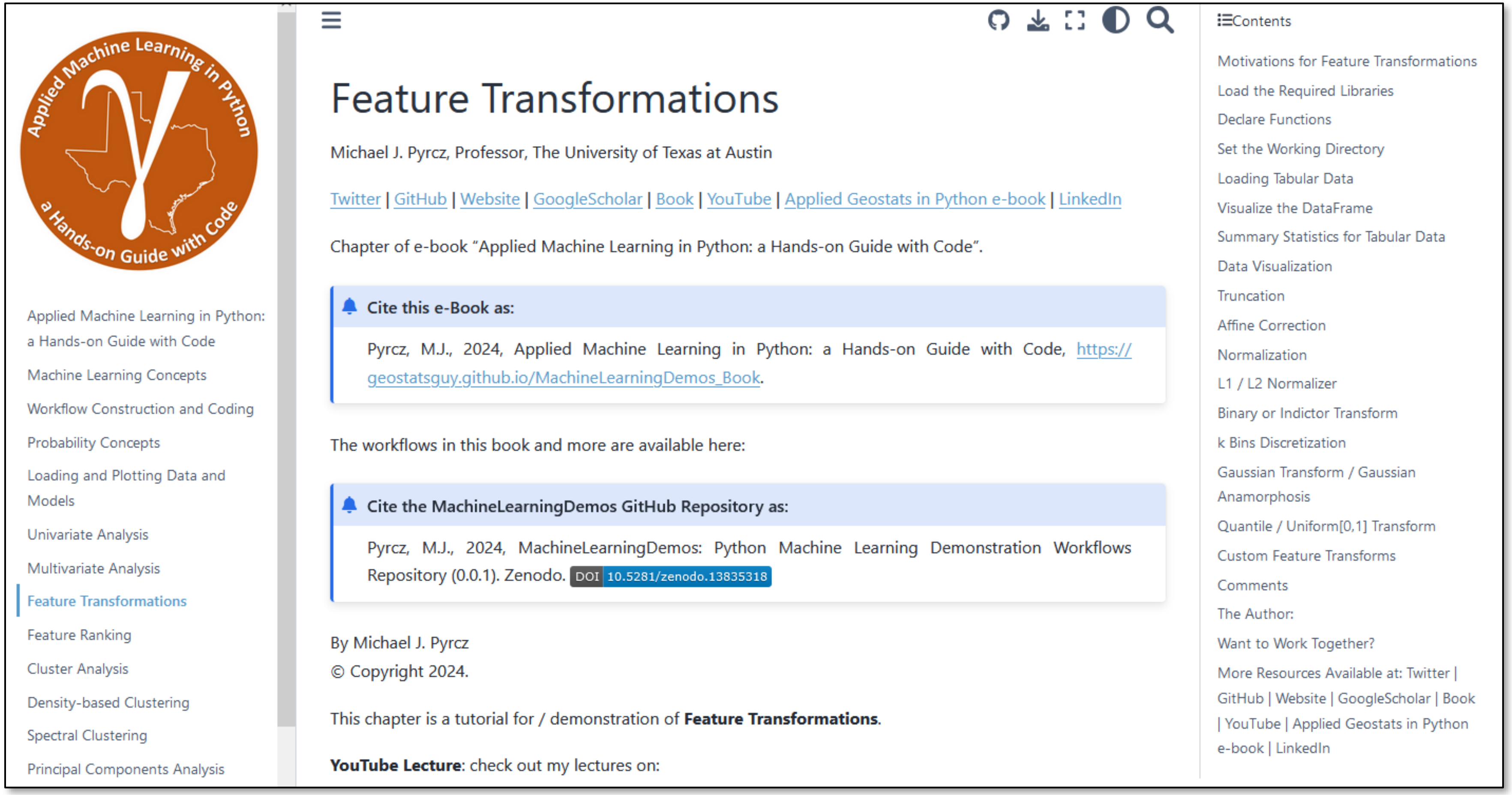
Lecture 5c: Feature Transformations

Lecture outline:

- **Feature Transformations Hands-on**

Feature Transformations Demonstration in Python

Demonstration of feature transformations with a documented workflow.



MachineLearning_feature_transformations chapter of e-book.

Feature Transformation
New Tools

Topic	Application to Subsurface Modeling
Feature Transformation	Apply feature transformations to improve the ability of your models to robustly infer patterns and predict away from training data. *Know what transformation are helpful and required for your modeling workflow.*
Feature Engineering	Ensure features have consistent scale and dimensionality. *Develop and use the most informative features to build a good prediction model.*

PGE 383 Subsurface Machine Learning

Lecture 5c: Feature Transformations

Lecture outline:

- Feature Transformations
- Feature Engineering
- Feature Transformations Hands-on