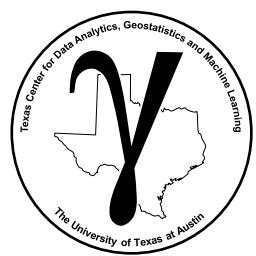


PGE 383 Subsurface Machine Learning

Lecture 5b: Feature Selection

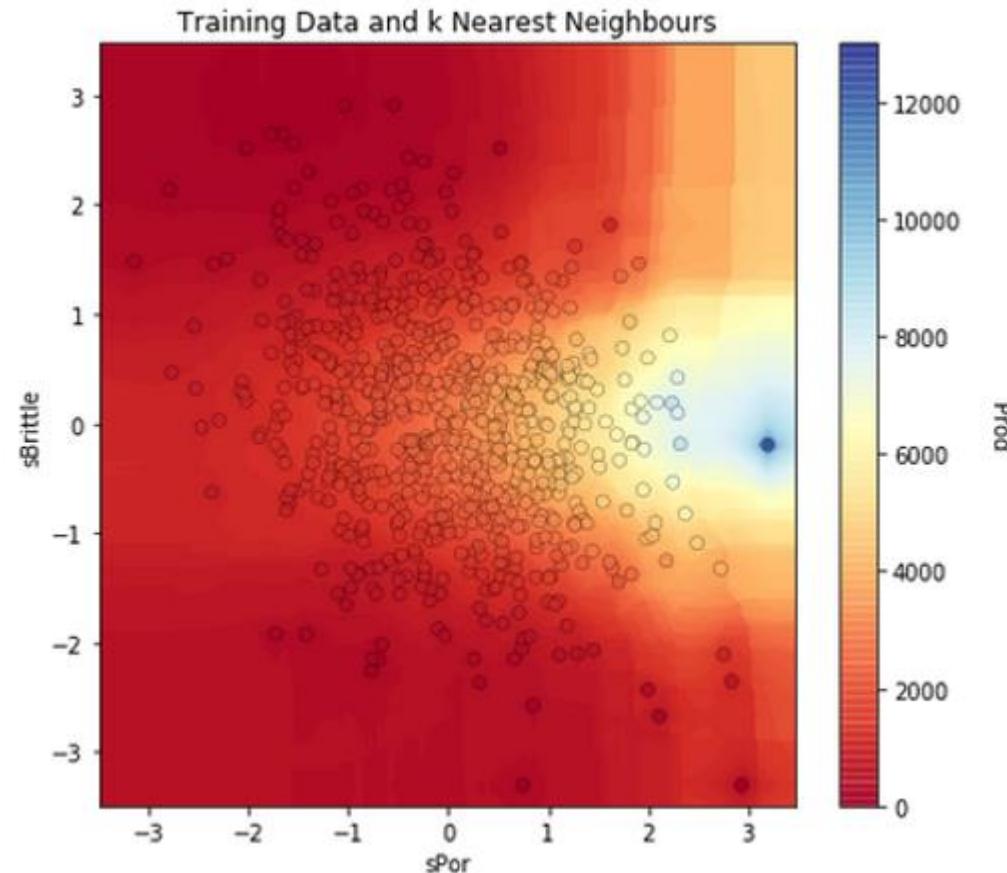
Lecture outline:

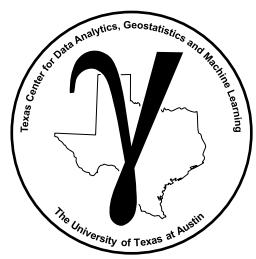
- Curse of Dimensionality
- Feature Selection
- Mutual Information
- Shapley Values
- Recursive Feature Selection
- Feature Selection Hands-on



Motivation for Multivariate Methods

We build better models when we careful select the most informative set of predictor features.



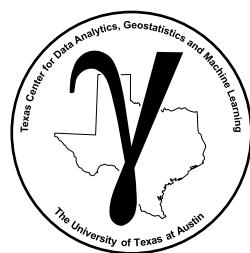


PGE 383 Subsurface Machine Learning

Lecture 5b: Feature Selection

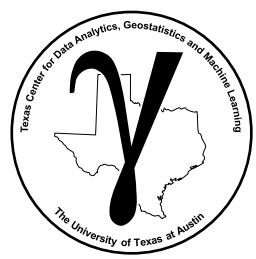
Lecture outline:

- Curse of Dimensionality



Working with Many Features

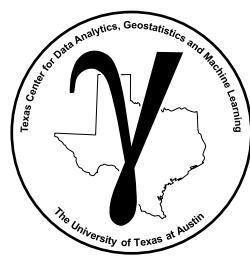
- One of the definitions of big data is variety
 - This suggests massively multivariate datasets, many features
- Traditional reservoir modeling workflows were bivariate
 - Facies, then porosity in facies and permeability constrained to porosity
 - The most complicated simulation is permeability accounting for the joint porosity simulated realization
- Unconventionals, and whole Earth models
 - Require inclusion many more variables
 - We need to model facies, porosity, geomechanical properties, geophysical properties, total organic carbon, maturity etc.
- When working with multivariate it is very challenging:
 - Visualize and detect relationships and patterns, models are less stable!



Curse of Dimensionality

The challenges of working with many features (i.e., high dimensional space)

- impossible to visualize in high dimensionality
- insufficient sampling
- low coverage of feature space
- distorted feature space
- multicollinearity between features

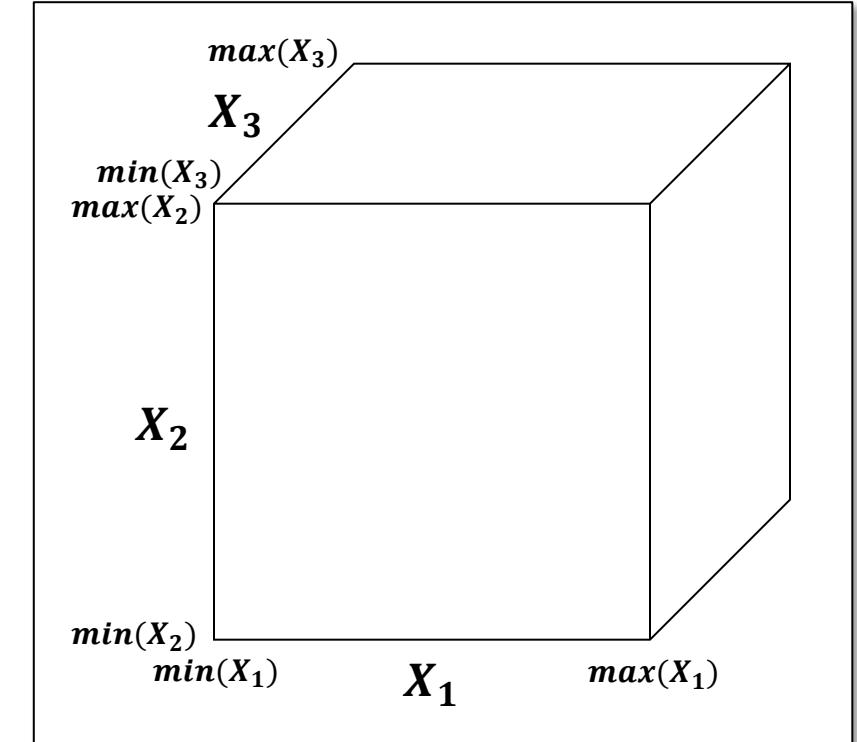


Feature Space Definition

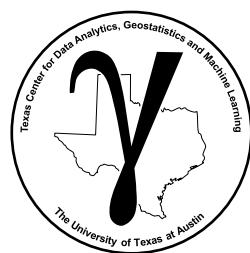
Feature Space

- The m -dimensional space, R^m , over the ranges of each of the features,
- In general our feature space includes all possible cases of our features.

$$x_1 \in [min(X_1), max(X_1)], \dots, x_m \in [min(X_m), max(X_m)]$$



Schematic of feature space.



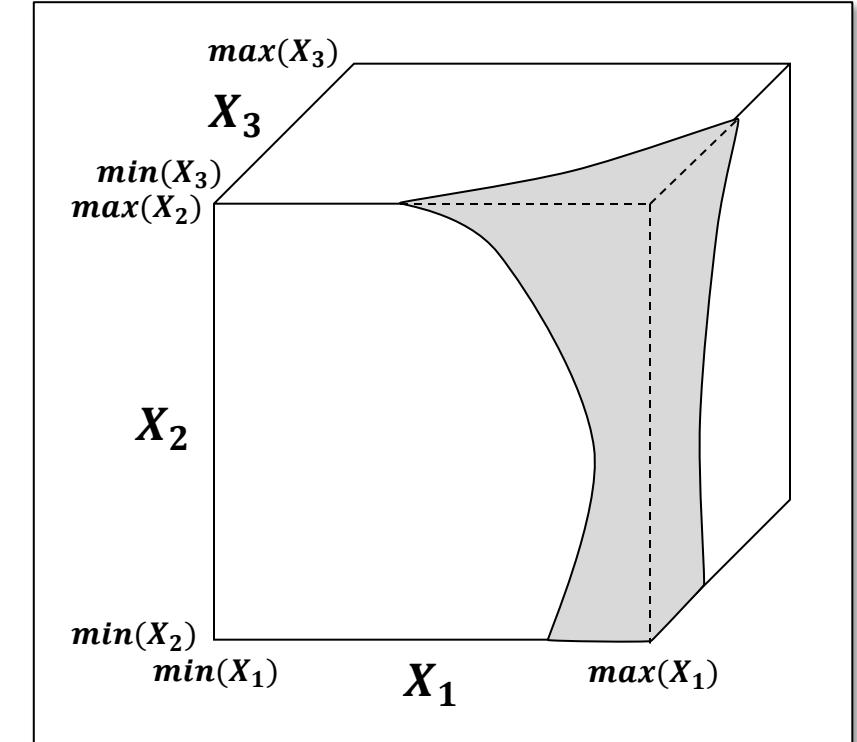
Feature Space Definition

Feature Space in Machine Learning

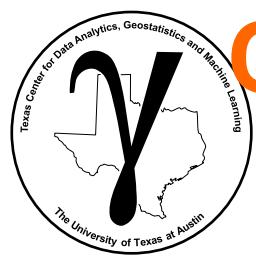
- Commonly ‘feature space’ only refers to the predictor features and does not include the response feature(s), i.e.,

The combination of values for which we need to make predictions.

- In this course, I will specify ‘predictor feature’ space.
- Typically, we will train our machines to make predictions over the predictor feature space.
- More complicated shapes of predictor feature space are possible, e.g., we could mask or remove subsets with poor data coverage.



Schematic of feature space.



Curse of Dimensionality Visualization

Consider this simple model:

- 1 predictor feature
- 1 response feature

How's our model performing?

- Accuracy in training and testing

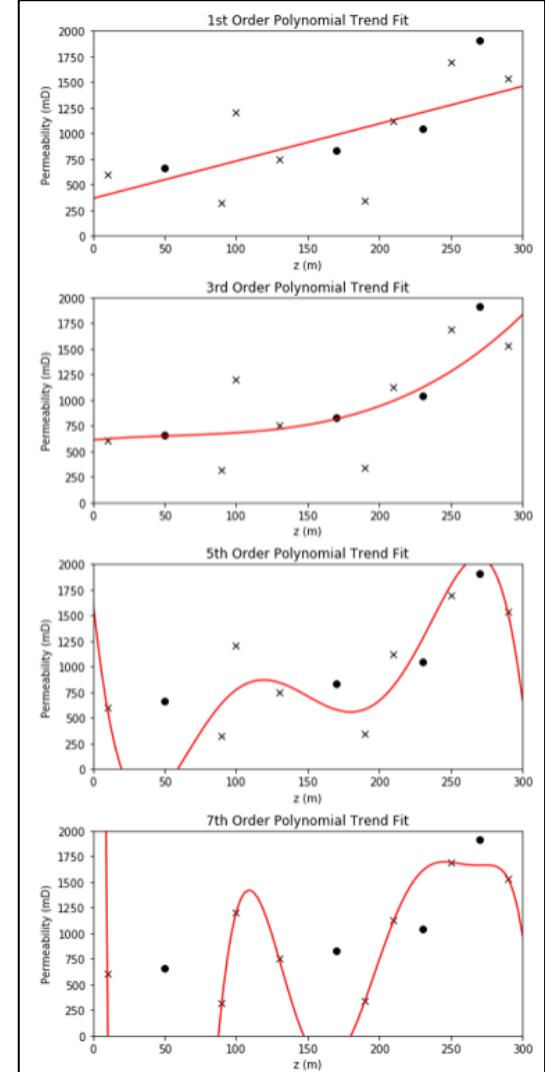
Range of Applicability?

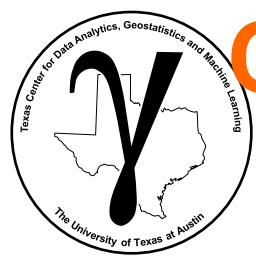
- Are we extrapolating?

Overfit

- Is the model defendable given the data?

Simple to complicated prediction models, from MachineLearning_model_tuning chapter of Applied Machine Learning in Python e-book.





Curse of Dimensionality Visualization

Consider this simple model:

- 2 predictor feature
- 1 response feature

How's our model performing?

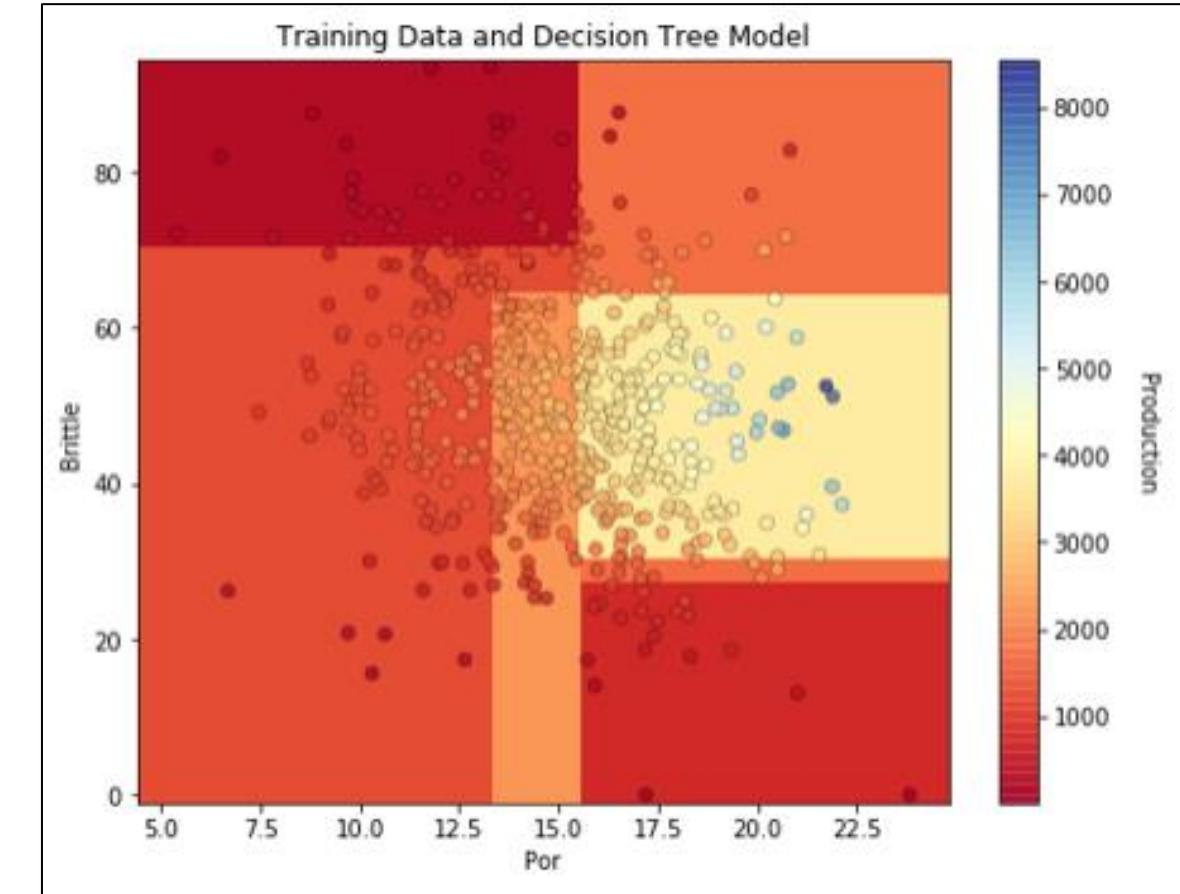
- Accuracy in training and testing

Range of Applicability?

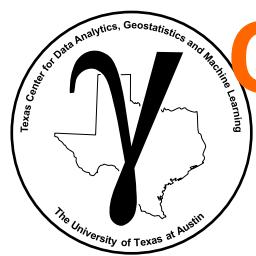
- Are we extrapolating?

Overfit

- Is the model defendable given the data?



Two predictor feature decision tree prediction model, from
MachineLearning_decision_tree chapter of
Applied Machine Learning in Python e-book.



Curse of Dimensionality Visualization

Consider this simple model:

- 4 predictor feature
- 1 response feature

How's our model performing?

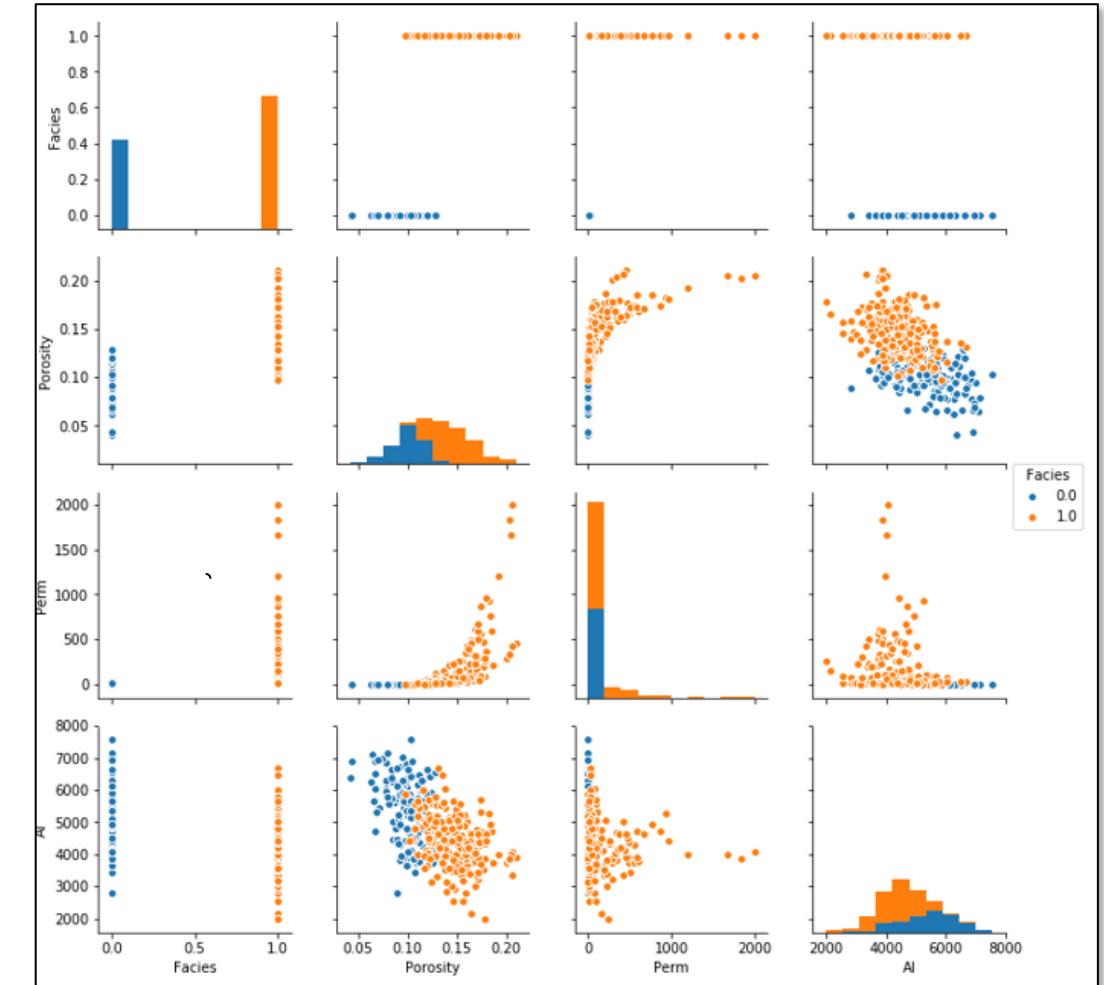
- Accuracy in training and testing

Range of Applicability?

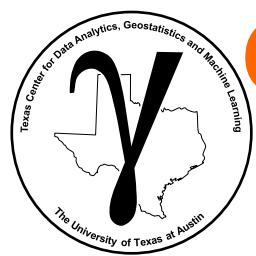
- Are we extrapolating?

Overfit

- Is the model defendable given the data?



Matrix scatter plot, from Bivariate Analysis chapter of course e-book.



Curse of Dimensionality Sampling

Recall the calculation of a histogram / normalized histogram.

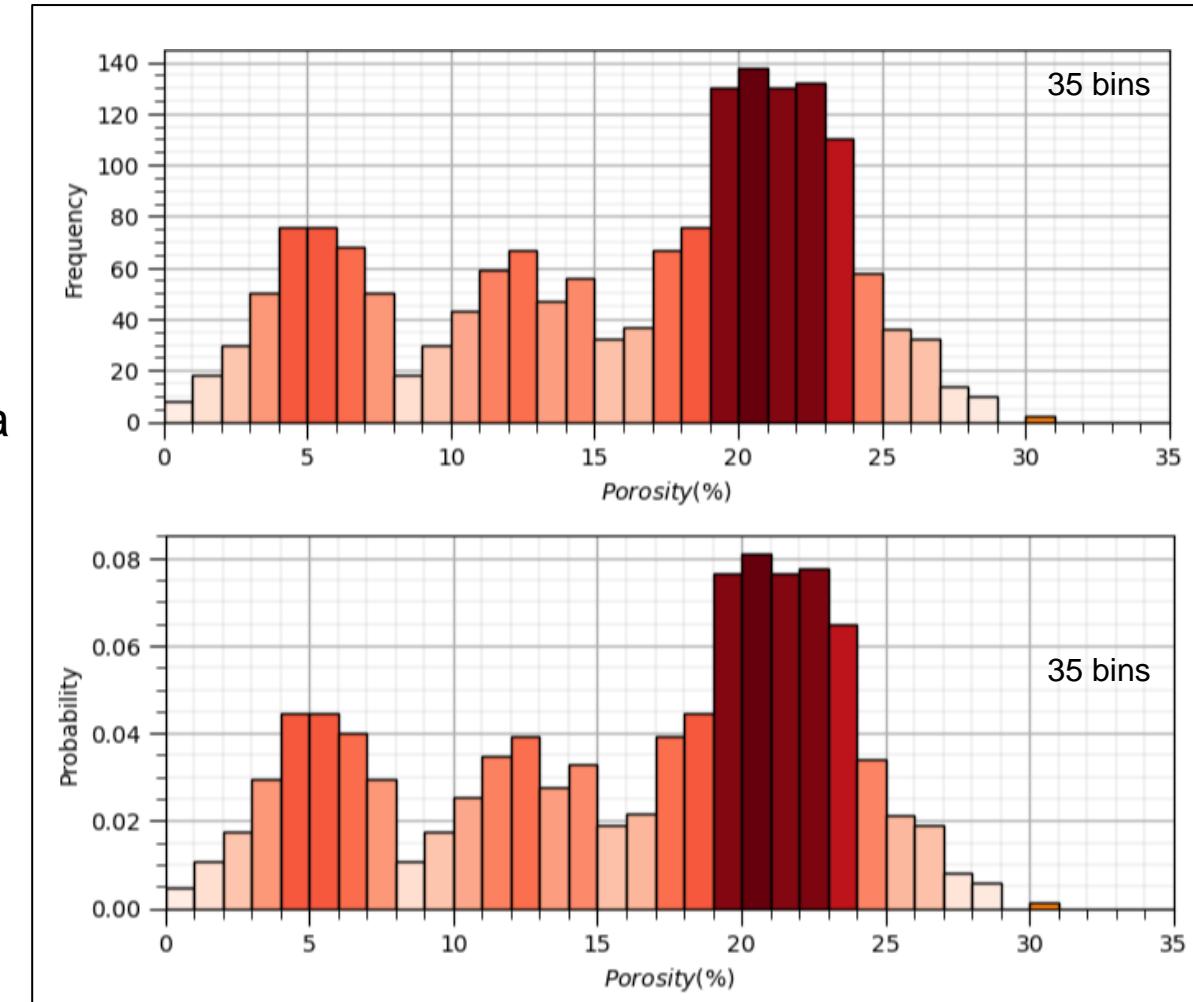
1. We establish 'bins', discretize the range of each feature.

2. We calculate probabilities with a ratio and enough samples/bin.

Number of samples needed, approx.,

$$n = n_{samples/bin} \cdot n_{bins}$$

$$n = 10 \cdot 35 = 350 \text{ data}$$

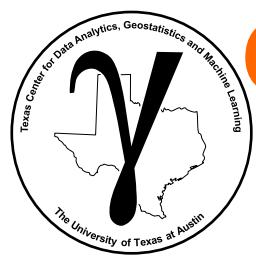


Histogram and normalized histogram, from PythonDataBasics_Bivariate_Visualization.ipynb.

$$n(X_1^i \leq X \leq X_1^{i+1})$$

Samples to infer this probability for each bin.

$$P(X_1^i \leq X \leq X_1^{i+1}) = \frac{n(X_1^i \leq X \leq X_1^{i+1})}{n}$$



Curse of Dimensionality Sampling

Calculating the Joint Probability

Consider any joint probability:

$$P(X_1 \in bin_{i_1} \cap \dots \cap X_m \in bin_{i_m})$$

- E.g. $P(10\% < \varphi < 11\%, 600 < k < 640) = n(10\% < \varphi < 11\%, 600 < k < 640)/n$

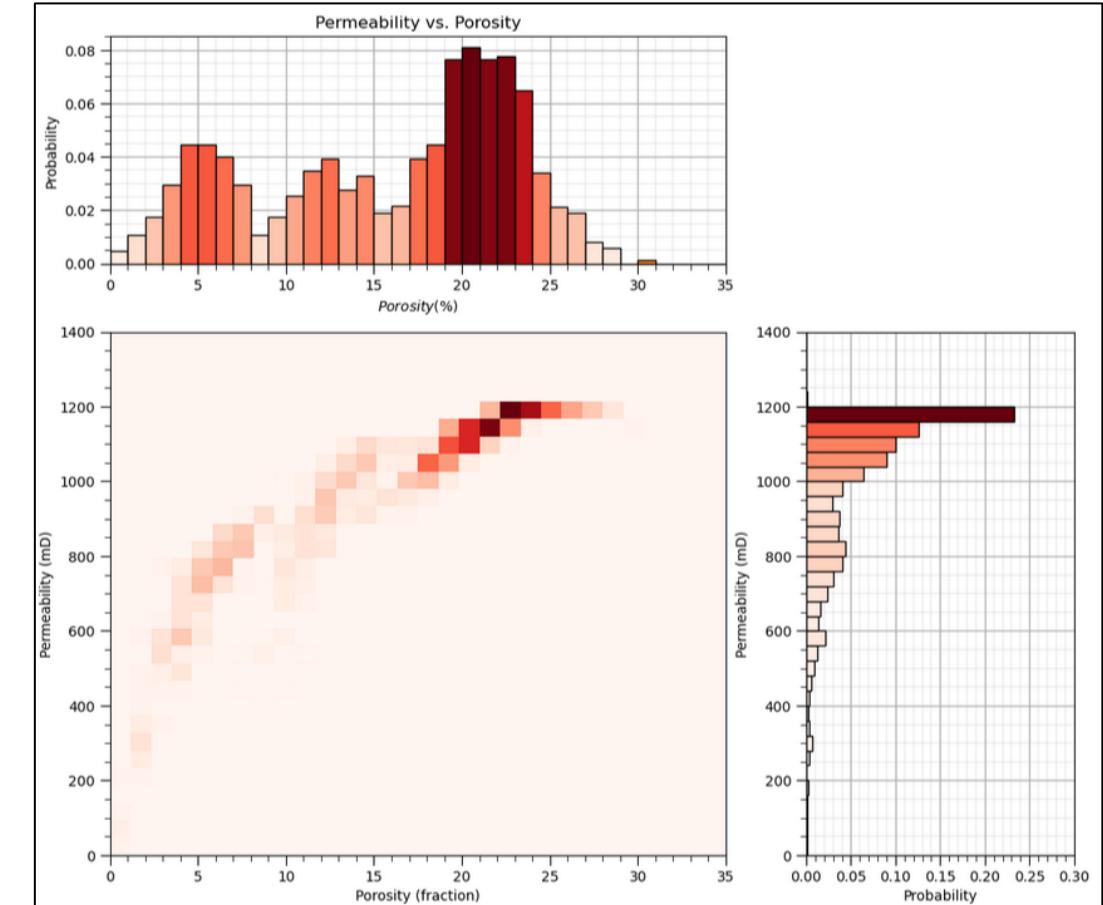
where n is the total number of samples.

- We need enough samples, n , replicates of all possible combinations to go from frequency to probability.
- Where, $n_{s/bin}$, is the nominal number of samples per combination.

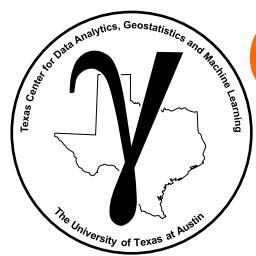
$$n = n_{s/bin} \cdot n_{bins}^m$$

- Note: this is optimistic, as it assumes uniform sampling

For this example, $n = n_{s/bin} \cdot n_{bins}^m = 10 \cdot 35^2$, given we need 10 nominally samples per bin, we need 12,250 data.



Joint PDF with normalized histograms, from PythonDataBasics_Bivariate_Visualization.ipynb.

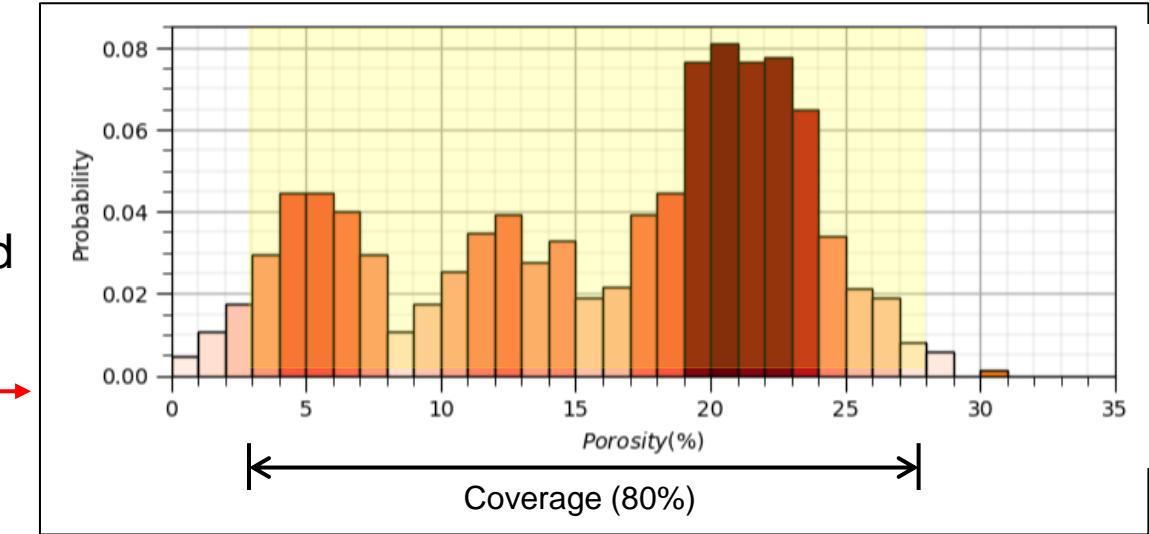


Curse of Dimensionality Coverage

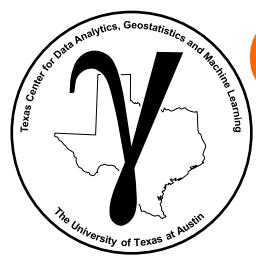
Sample Coverage

The range of the sample values

- The fraction of the possible solution space that is sampled
- Let's return to 1 feature and assume 80% coverage! →
- That's pretty good right?
- Remember, we usually, directly sample only $\frac{1}{10^7}$ of the volume of the subsurface
- Yes, the concept of coverage is subjective, how much data to cover? What about gaps? etc.



Histogram and normalized histogram, from
PythonDataBasics_Bivariate_Visualization.ipynb.



Curse of Dimensionality Coverage

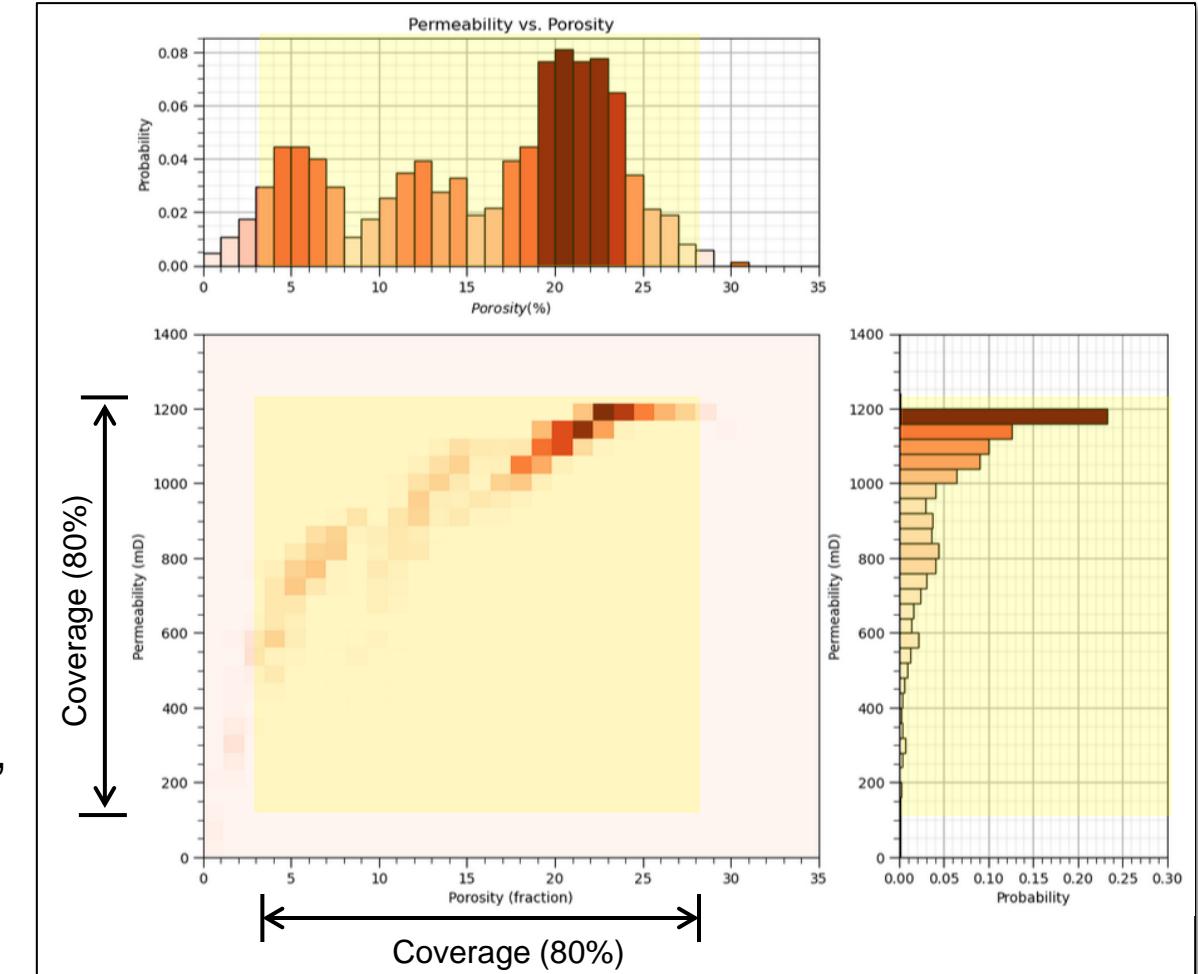
Sample Coverage

It is common not to have samples that cover the entire predictor feature space

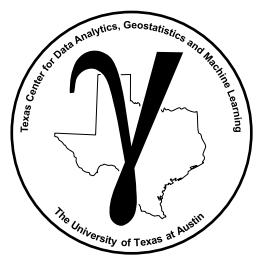
- Consider coverage over each feature, c_1
- How much of the solution space is covered?

$$c = c_1^m$$

- coverage is decreasing, exponential decay with decay constant, $\lambda = 1$, as we increase the number of features, m !



Coverage for this example is, $c = c_1^m = 0.8^2 = 0.64$, image from PythonDataBasics_Bivariate_Visualization.ipynb.



Curse of Dimensionality

Distorted Space

Distances in High Dimensional Space

Hyperdimensional space is distorted,

- Take the ratio of the volume of an inscribed hypersphere in a hypercube.

$$\frac{\pi^{m/2}}{m2^{m-1}\Gamma(m/2)} \rightarrow 0 \text{ as } m \rightarrow \infty$$

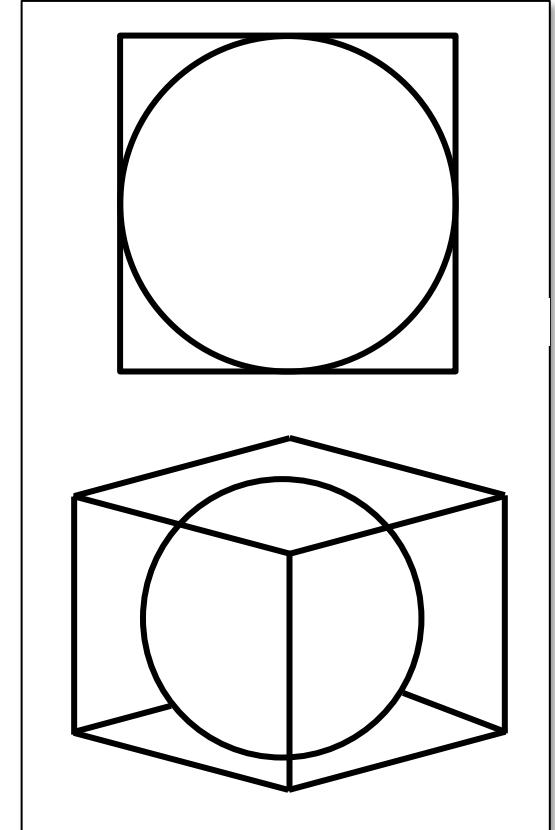
Recall, $\Gamma(n) = (n - 1)!$

- High dimensional space is all corners and no ‘middle’ and most of high dimensional space is far from the middle (all corners!).

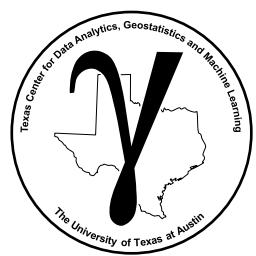
Distance in hyperdimensional space loses variance,

$$\lim_{m \rightarrow \infty} E\{dist_{max}(m) - dist_{min}(m)\} \rightarrow 0$$

- The limit of the expectation of the range of pairwise distances over random points in hyperdimensional space tends to zero.
 - Distances are almost all the same, Euclidian distance is no longer meaningful



Ratio of (hyper)sphere inscribed in (hyper)cube.



Curse of Dimensionality

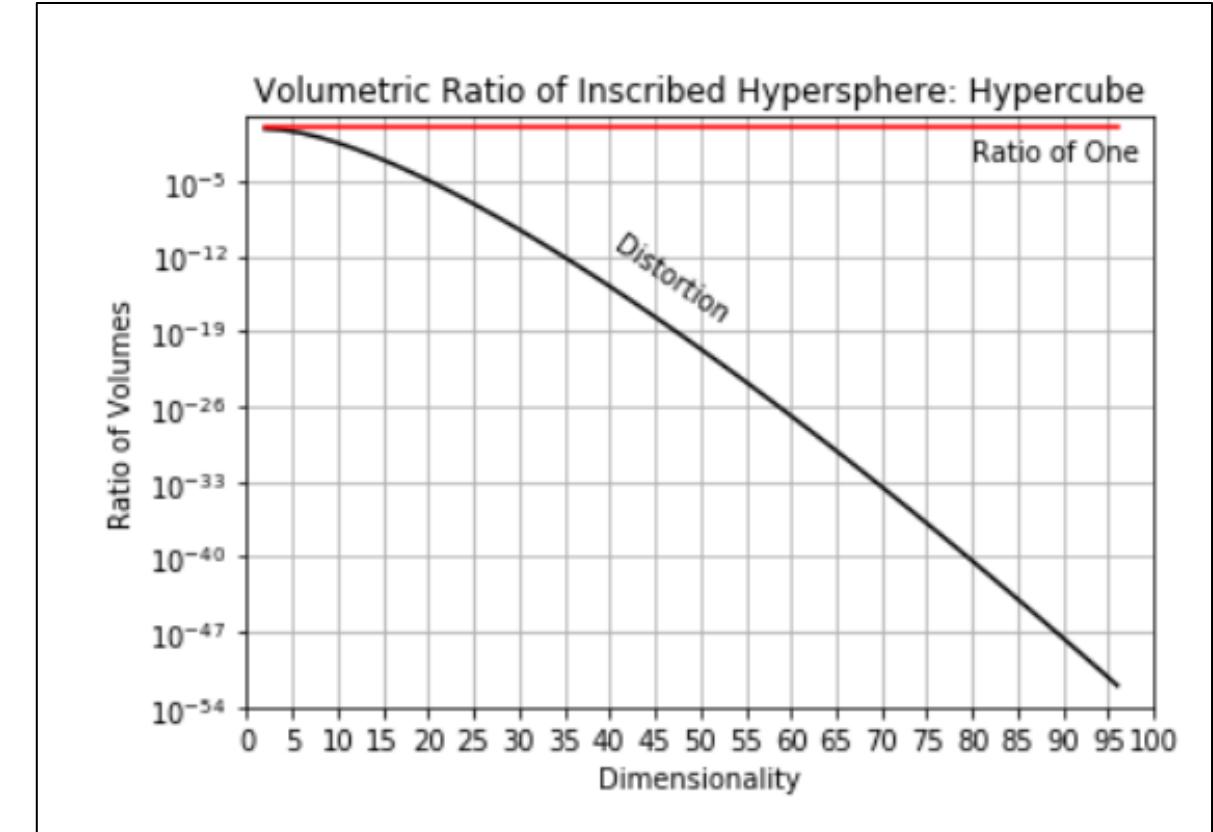
Distorted Space

Distortion of High Dimensional Space

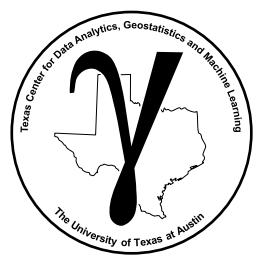
How many dimensions, features to observe the curse of dimensionality?

- Here's the ratio of center to corners from the previous equation.
- 10 or more features, the distortion is severe,

m	mD / 2D
2	1.0
5	0.28
10	0.003
20	0.00000003



Volumetric ratio of an inscribed hypersphere (n-sphere) in a hypercube (n-cube), note y-axis is log-scale.



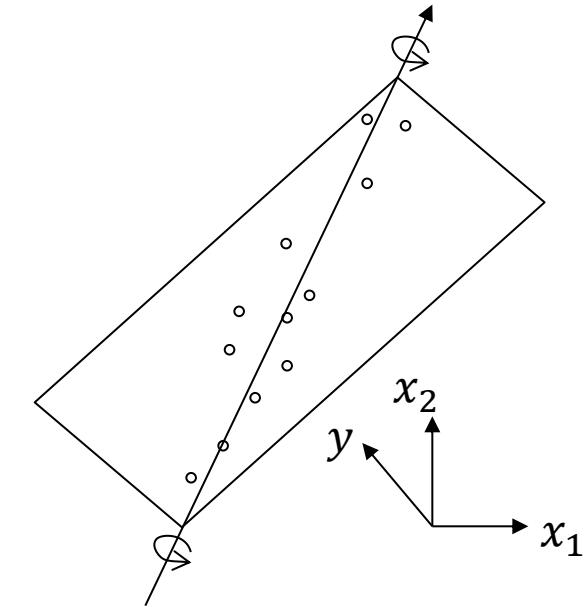
Curse of Dimensionality Multicollinearity

"The existence of such a high degree of correlation between supposedly independent variables being used to estimate a dependent variable that the contribution of each independent variable to variation in the dependent variable cannot be determined"

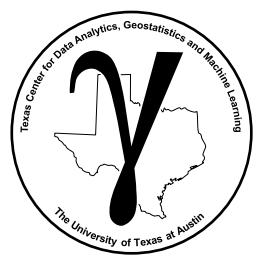
- Merriam-Webster Online Dictionary

"In statistics, multicollinearity (also collinearity) is a phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy."

- Wikipedia



Colinearity of x_1 and x_2 when modeling $y = f(x_1, x_2)$,
is like fitting a plane to a line!

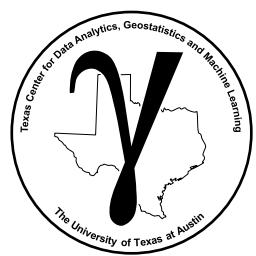


Curse of Dimensionality

We get a better model with fewer, informative features than,

'Throwing everything and the kitchen sink into the model!'

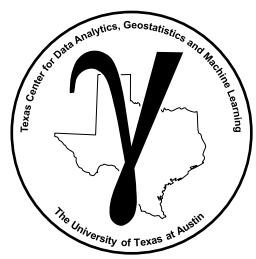
Fewer features for models are simpler, faster, easier to visualize and less likely overfit and results in **the best model!**



Curse of Dimensionality

Working with More Features is More Difficult

1. More difficult to visualize data and model
2. More data are required to infer the joint probabilities
3. Less data coverage of feature space
4. More difficult to interrogate / check the model
5. More likely redundant features resulting in model instability
6. More computational effort, more computational resources and longer run times
7. More complicated model is more likely overfit
8. More professional time for model construction

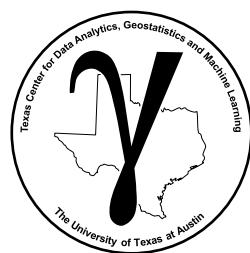


PGE 383 Subsurface Machine Learning

Lecture 5b: Feature Selection

Lecture outline:

- Feature Selection



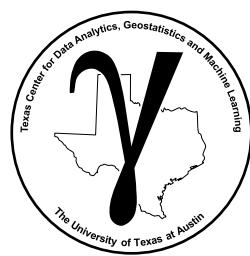
Feature Ranking

Feature Ranking

- Feature ranking is a set of methods that assign relative importance or value to each feature with respect to information contained for inference and importance in predicting a response feature.
- There are a wide variety of possible methods to accomplish this.
- My recommendation is a **wide-array** approach with multiple metric, while understanding the assumptions and limitations of each method.

Here's the general types of metrics that we will consider for feature ranking:

1. Visual Inspection of Data Distributions and Scatter Plots
2. Statistical Summaries
3. Model-based
4. Recursive Feature Elimination

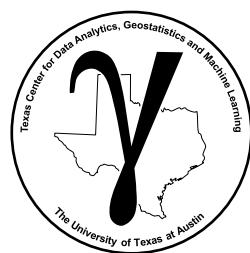


Feature Ranking

Expert Knowledge

Do NOT forgot to integrate expert knowledge

- If additional information is known about physical processes, causation, reliability and availability of features this should be integrated into assigning feature ranks.
- We are learning as we perform our analysis, testing new hypotheses.



Feature Ranking

Sample Coverage

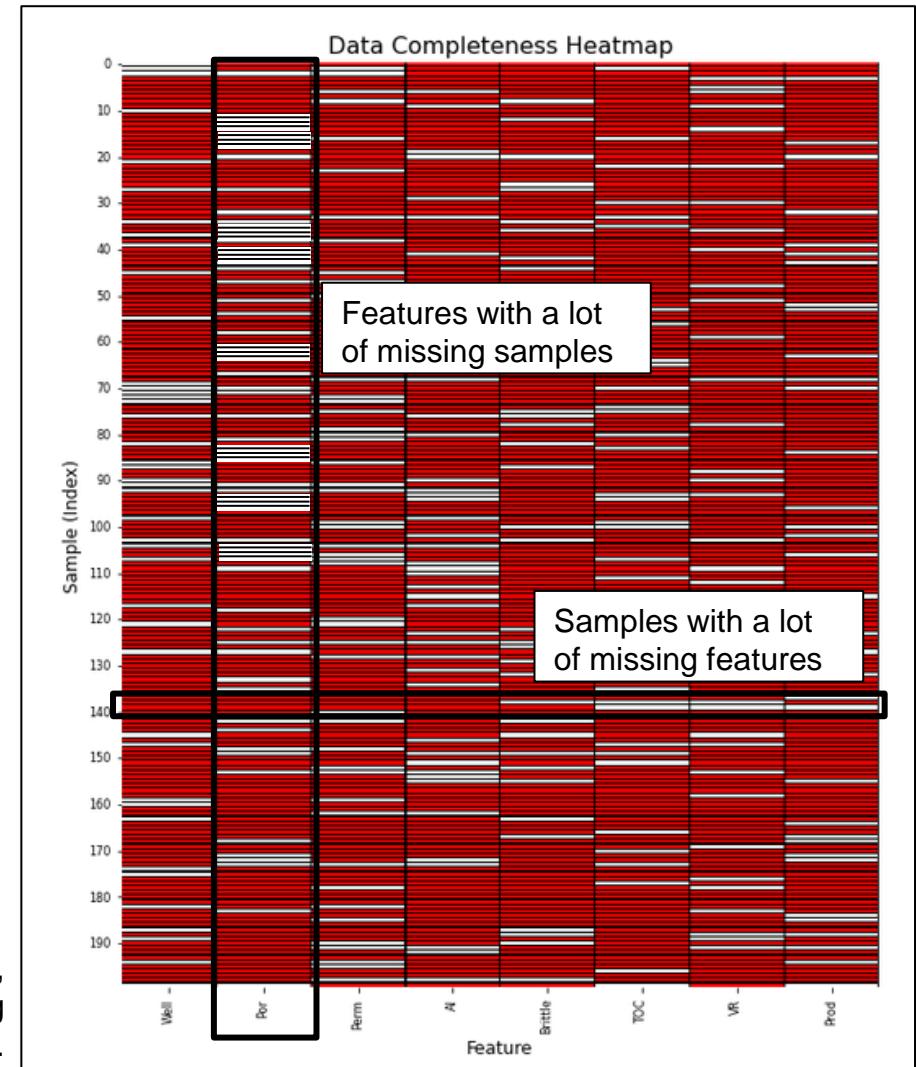
When we include features with a lot of missing values we either:

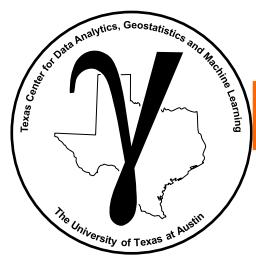
- increase the data imputation challenge
- loose a lot of samples if likewise deletion is used

We can use a heat map to check data coverage

- identify features with a lot of missing samples
- identify samples with a lot of missing features.

Sample coverage heat map, y-axis is sample, and x-axis is feature, red is present and white is missing, from MachineLearning_feature_ranking chapter of e-book.



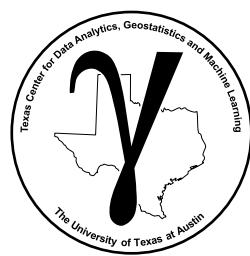


Feature Ranking

Visual Inspection of Metrics

Summary statistics of one variable at a time,

- are there data issues?
- do we trust the features? do we trust the features all equally?
- are there issues that need to be taken care of before we develop any multivariate workflows?
- coverage, are there a lot of missing values?



Feature Ranking Metrics

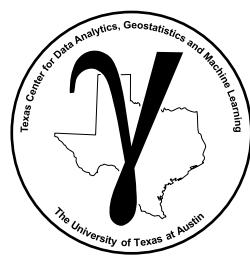
Visual Inspection of Metrics

	count	mean	std	min	25%	50%	75%	max
Well	200.0	100.500000	57.879185	1.000000	50.750000	100.500000	150.250000	200.000000
Por	200.0	14.991150	2.971176	6.550000	12.912500	15.070000	17.402500	23.550000
Perm	200.0	4.330750	1.731014	1.130000	3.122500	4.035000	5.287500	9.870000
AI	200.0	2.968850	0.566885	1.280000	2.547500	2.955000	3.345000	4.630000
Brittle	200.0	48.161950	14.129455	10.940000	37.755000	49.510000	58.262500	84.330000
TOC	200.0	0.991950	0.478264	0.000000	0.617500	1.030000	1.350000	2.180000
VR	200.0	1.964300	0.300827	0.930000	1.770000	1.960000	2.142500	2.870000
Prod	200.0	3864.407081	1553.277558	839.822063	2686.227611	3604.303507	4752.637556	8590.384044
const	200.0	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Sample statistics and valid (non-null) counts for each feature.

Example,

- the number of valid (non-null) values for each feature, e.g., no missing values
- general behaviors such as central tendency, mean, and dispersion, variance, e.g., permeability has no skew
- issues with negative values, extreme values, and values that are outside the range of plausible values for each property, e.g., no issues



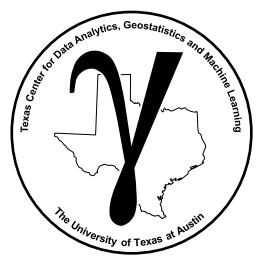
Feature Ranking Metrics

Univariate Distributions

As with summary statistics, this ranking method is a qualitative check for issues with the data and to assess our confidence with each feature.

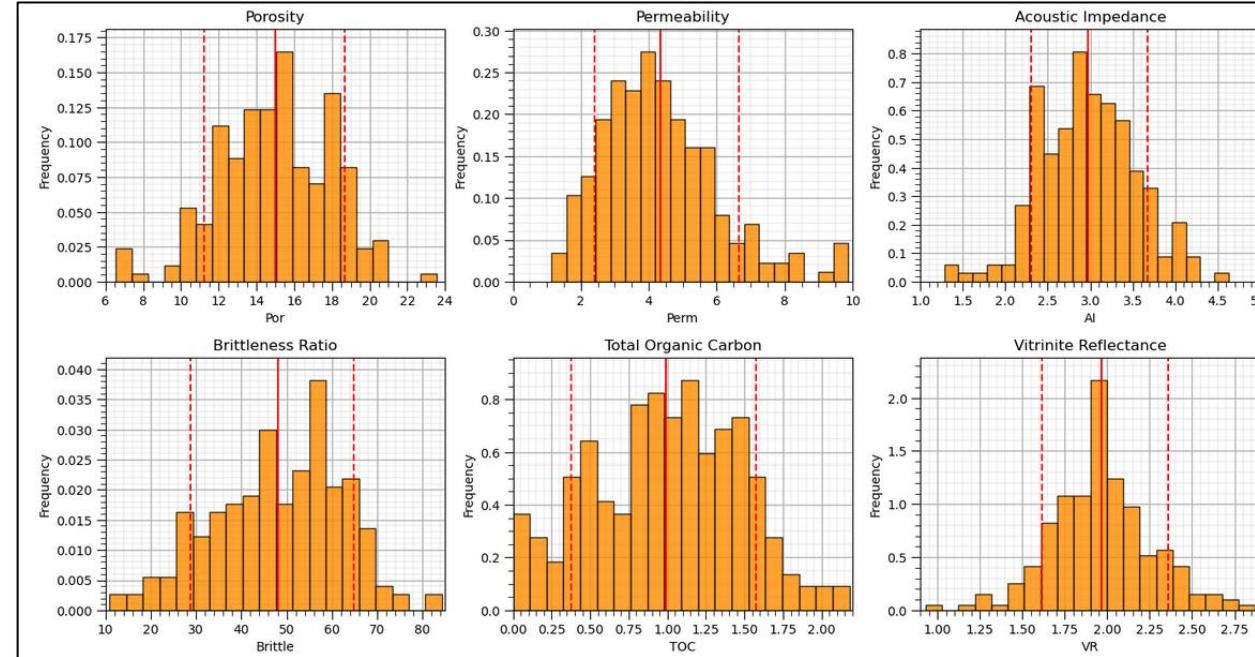
It is better to not include a feature with low confidence of quality as it may be misleading (while adding to model complexity as discussed previously).

Assess our ability to use methods that have distribution assumptions



Feature Ranking Metrics

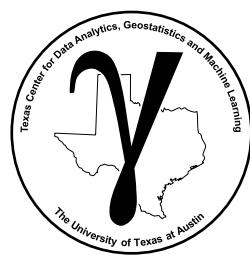
Univariate Distributions



Feature histograms with P10, P50 and P90 shown as vertical lines, from MachineLearning_feature_ranking chapter of e-book.

Example,

- check for obvious outliers, e.g., no issues
- distribution skew, e.g., permeability is positively skewed
- data spikes, indicating truncation, e.g., TOC has a small zero truncation spike
- Gaussianity or non-Gaussian form, i.e., could transform data if required by workflow

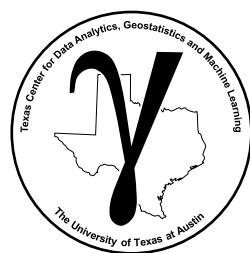


Feature Ranking Metrics

Bivariate Statistics

Matrix scatter plots are a very efficient method to observe the bivariate relationships between the variables.

- this is another opportunity through data visualization to identify data issues, e.g., outliers
- we can assess if we have collinearity, between any two features at a time, or constraint relationships between pairs of features
- Bivariate Gaussian is assumed for methods such as correlation and partial correlation



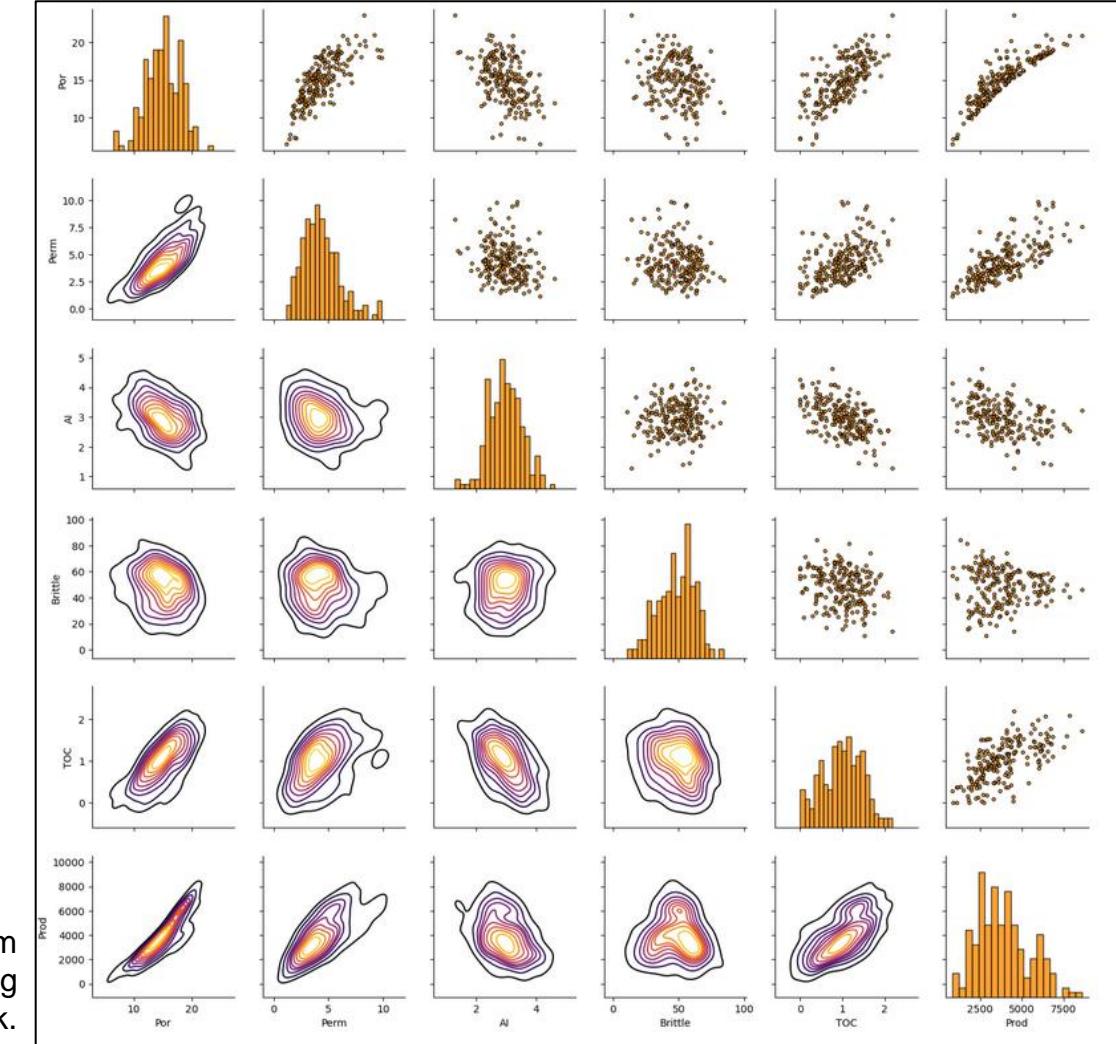
Feature Ranking Metrics

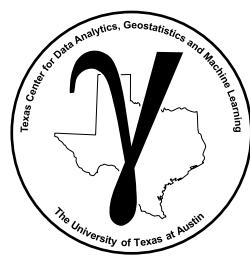
Bivariate Statistics

Example,

- collinearity, variables that are closely related to each other, e.g., porosity and permeability
- linear vs. non-linear relationships, e.g., porosity and production vs. brittleness and production
- constraint relationships and heteroscedasticity between variables, e.g., porosity and production, permeability and production

Matrix scatter plot, from
MachineLearning_feature_ranking
chapter of e-book.





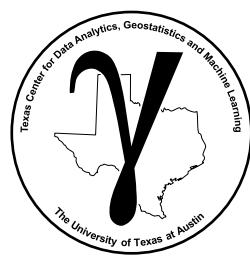
Feature Ranking Metrics

Pairwise Covariance

- Pairwise covariance provides a measure of the strength of the linear relationship between each predictor feature and the response feature
- We now specify our goal of this study is to predict production, our response variable, from the other available predictor features
- We are thinking predictively now, not inferentially, we want to estimate the function, \hat{f} to accomplish this

$$C_{xy} = \frac{1}{n-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

- sensitive to the dispersion / variance of both the predictor and response



Feature Ranking Metrics

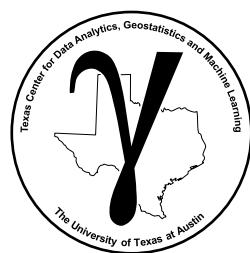
Pairwise Covariance

Sensitive to feature variance and feature variance is somewhat arbitrary. For example,

- variance of porosity in fraction vs. percentage, or permeability in Darcy (D) vs. millidarcy (mD).
- if we apply a constant multiplier, c , to a variable, X , that the variance will change according to this relationship (the proof is based on expectation formulation of variance):

$$\sigma_{cX}^2 = c^2 \sigma_X^2$$

- By moving from porosity percentage to fraction we decrease the variance of porosity by a factor of 10,000 and from D to mD by a factor of 1,000,000.
- The relative variance of each feature is potentially arbitrary, with the exception when all the features are in the same units.



Feature Ranking Metrics

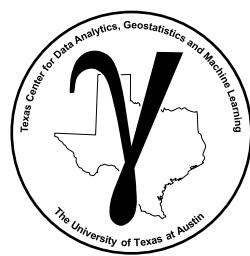
Pairwise Correlation Coefficient

Pairwise correlation coefficient provides a standardized measure of the strength of the linear relationship between each predictor feature and the response feature.

$$\rho_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)\sigma_x\sigma_y}, -1.0 \leq \rho_{xy} \leq 1.0$$

The correlation coefficient:

- measures the linear relationship, $[-1, 1]$
- removes the sensitivity to the dispersion / variance of both the predictor and response features, by normalizing by the product of the standard deviation of each feature



Feature Ranking Metrics

Rank Correlation Coefficient

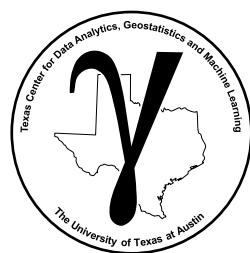
The rank correlation coefficient applies the rank transform to the data prior to calculating the correlation coefficient.

- To calculate the rank transform simply replace the data values with the ranks, where n is the maximum value and 1 is the minimum value.

$$\rho_{R_x, R_y} = \frac{\sum_{i=1}^n (R_{x_i} - \bar{R}_x)(R_{y_i} - \bar{R}_y)}{(n-1)\sigma_{R_x}\sigma_{R_y}}, -1.0 \leq \rho_{xy} \leq 1.0$$

The rank correlation:

- measures the monotonic relationship, relaxes the linear assumption
- removes the sensitivity to the dispersion / variance of both the predictor and response, by normalizing by the product of the standard deviation of each.



Feature Ranking Metrics

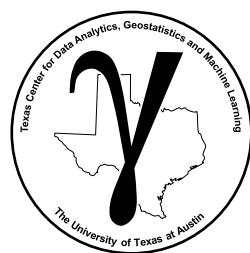
Partial Correlation Coefficient

This is a linear correlation coefficient that controls for the effects all the remaining variables

- $\rho_{XY.Z}$ and is the partial correlation between X and Y after controlling for Z .

Steps to calculate partial correlation coefficients,

1. perform linear, least-squares regression to predict X from $Z_{1,\dots,m-2}$.
2. calculate the residuals in Step #1, $X - X^*$
3. perform linear, least-squares regression to predict Y from $Z_{1,\dots,m-2}$.
4. calculate the residuals in Step #1, $Y - Y^*$
5. calculate the correlation coefficient, $\rho_{XY.Z} = \rho_{X - X^*, Y - Y^*}$



Feature Ranking Metrics

Partial Correlation Coefficient

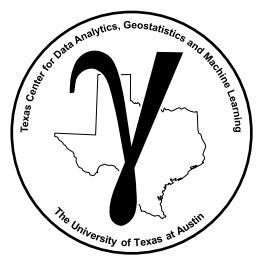
The partial correlation, provides a measure of the linear relationship between X and Y while controlling for the effect of Z other features on both, X and Y

Assumptions to apply partial correlation, we assume:

- linear relationships between all variables
- no significant outliers
- approximately bivariate normality between the variables

We are in pretty good shape, but we have some departures from bivariate normality.

- We apply a Gaussian transform in the demonstration to improve this

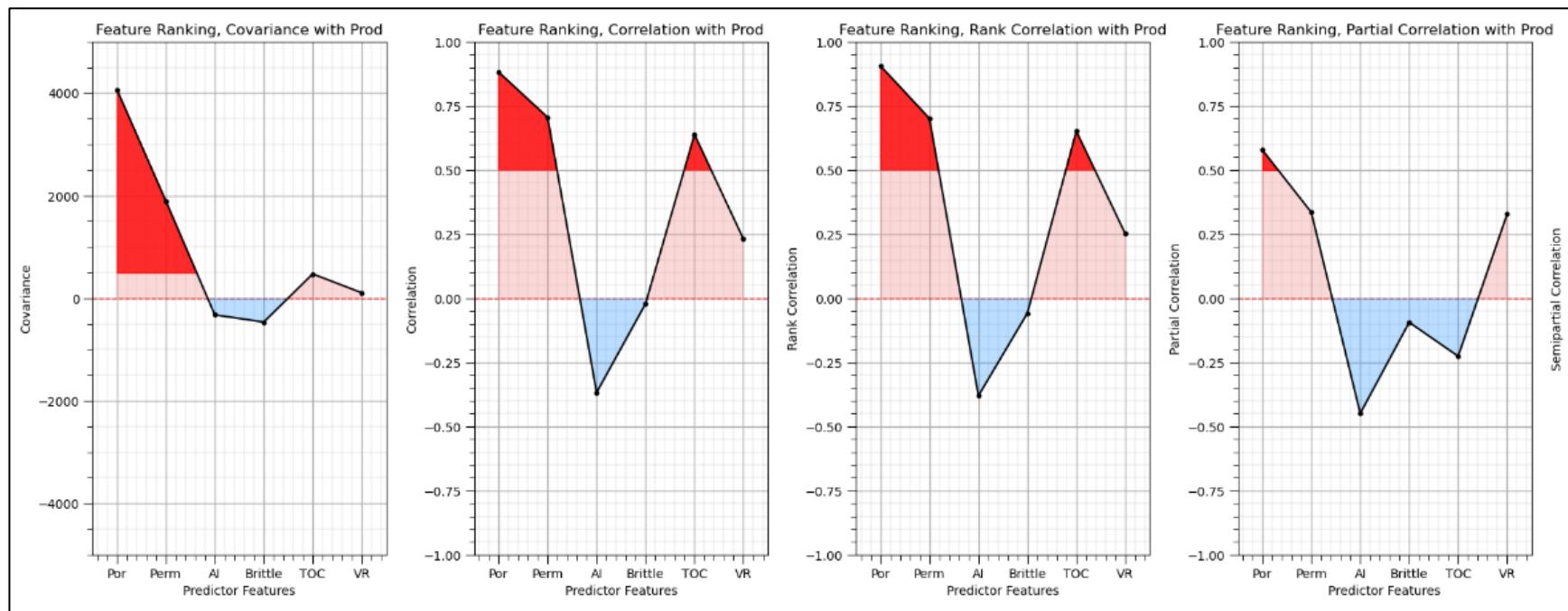


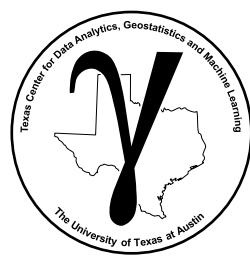
Feature Ranking Metrics

Feature Ranking Metrics Summary

Side-by-side comparison of feature ranking metrics, for example,

- porosity, permeability are important, but redundant
- brittleness is impacted by nonlinearity
- outliers are not an issue, rank correlation is same as correlation





Feature Ranking Metrics

Conditional Statistics

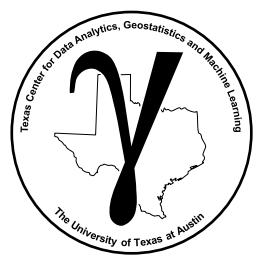
Access the conditional distributions and probabilities of all predictor features given the response feature.

$$F_{X_\alpha | Y}(x_\alpha | y) = \frac{F_{X_\alpha, Y}(x_\alpha, y)}{F_Y(y)}$$

- We can access the difference between the conditional distributions given low and high case of the response feature.

$$F_{X_\alpha | Y}(x_\alpha | y_{low}) \sim F_{X_\alpha | Y}(x_\alpha | y_{high})$$

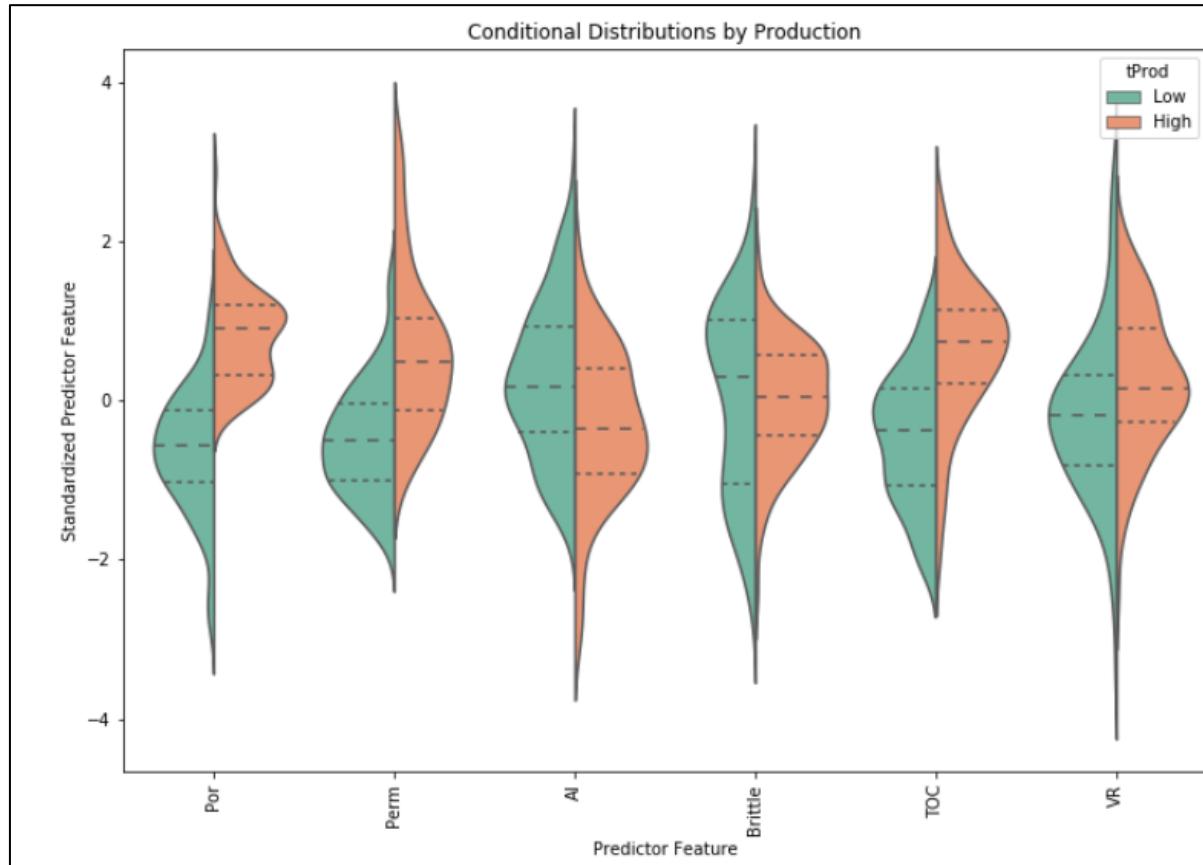
then X_α does not provide information on Y .



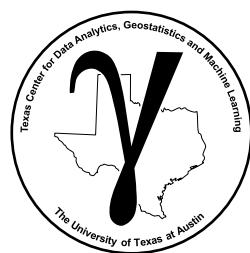
Feature Ranking Metrics

Conditional Statistics

Standardized each feature, truncate the response feature (if continuous) the calculate a violin or box plot.



Violin plot for 6 predictor features vs. low and high production rate, from MachineLearning_feature_ranking chapter of e-book



Feature Ranking Metrics

Model-based Ranking – B coefficients

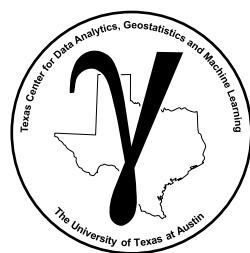
Evaluate the B coefficients, model parameters from multilinear regression.

$$Y^* = \sum_{i=1}^m B_i X_i + c$$

- These are the linear regression coefficients without standardization of the variables.
- Sensitive to feature variance.
- We are capturing interactions between variables.

Caution: model-based feature ranking is dependent on having a good model.

- Always check the model prediction accuracy!



Feature Ranking Metrics

Model-based Ranking – β (beta) coefficients

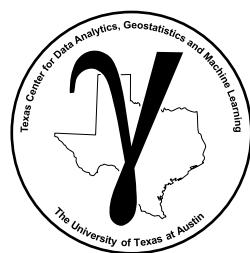
Evaluate the B coefficients, model parameters from multilinear regression of standardized features,

$$Y^{s*} = \sum_{i=1}^m \beta_i X_i^s + c$$

- These are the linear regression coefficients with standardization of the variables, X_i^s and Y^{s*} (variance = 1)
- Not sensitive to variance of the features
- We are capturing linear redundancy between predictor variables.

Caution: model-based feature ranking is dependent on having a good model.

- Always check the model prediction accuracy!



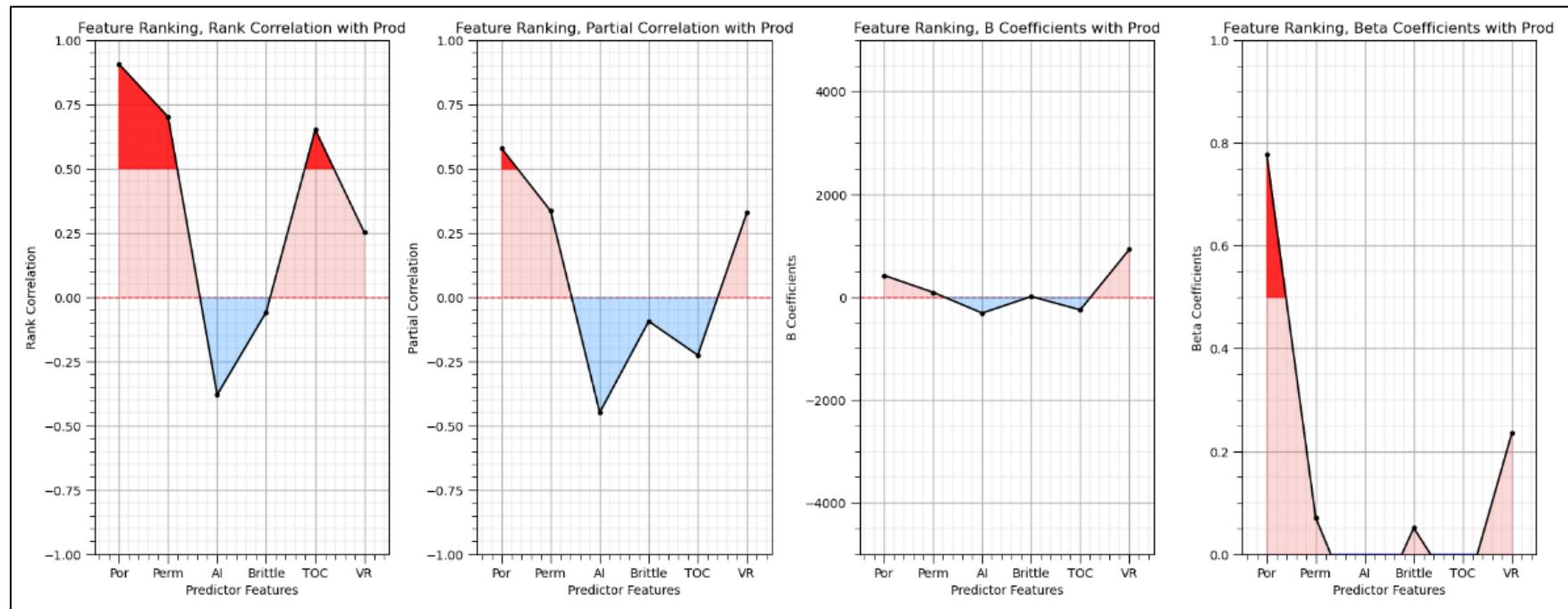
Feature Ranking Metrics

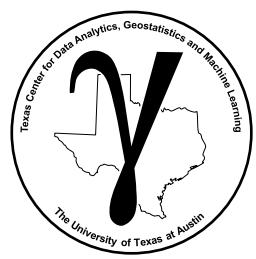
Ranking Methods – Summary of Results

Now what do we see?

- Beta demotes permeability!
- Porosity, acoustic impedance and vitrinite reflectance retain high metrics

Matrix scatter plot, from
MachineLearning_feature_ranking
chapter of e-book.



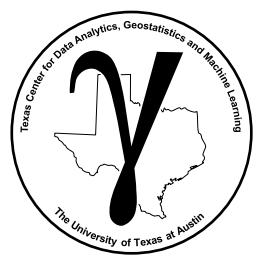


PGE 383 Subsurface Machine Learning

Lecture 5b: Feature Selection

Lecture outline:

- Mutual Information

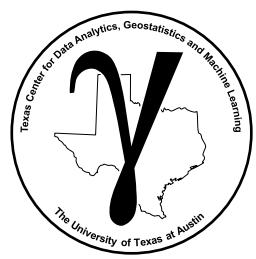


Mutual Information

Mutual Information

From probability and information theorem, a more general measure of amount of information from X_α about Y

- Nonparametric measure without assumption of the form of the relationship
- Units are ‘Shannons’ / ‘bits’
- Measure of the difference between the joint $P(x, y)$ and the product of the marginals $P(x) \cdot P(y)$, integrated over all $x \in X$ and $y \in Y$.



Mutual Information

Mutual Information

Derived from the general definition of independence for joint and marginal,

$$P(X, Y) = P(X) \cdot P(Y)$$

Joint and marginal definition of independence.

- Recall:

$$P(X, Y) = P(Y|X)P(X)$$

Reordered the definition of conditional probability and substitute.

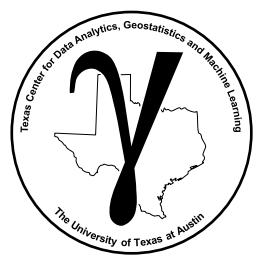
- After substitution, the expression for conditional and marginal independence:

$$P(Y, X) = P(X) \cdot P(Y)$$

$$P(Y|X) = P(Y)$$

Interpretation:

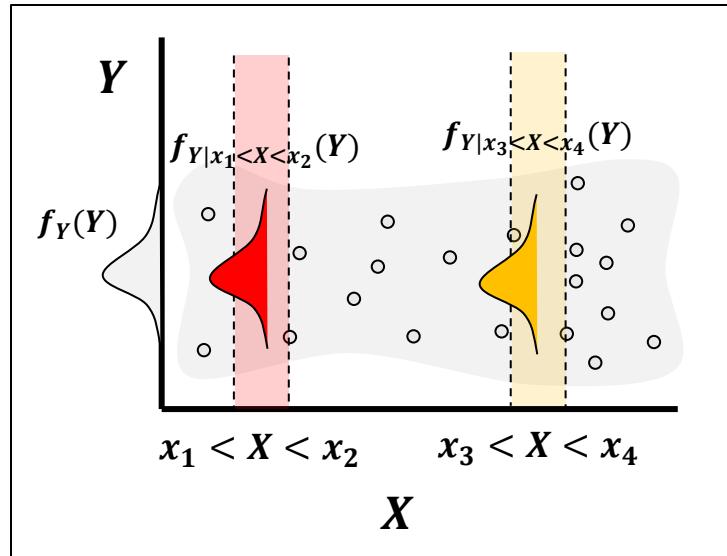
Knowing something about X tells us nothing about Y !



Mutual Information

Mutual Information

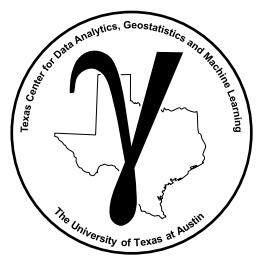
Independence and conditional probabilities.



General definition of independence
visualized with conditional and marginal
distributions.

$$f_{Y|x_1 \leq X \leq x_2}(Y) = f_{Y|x_3 \leq X \leq x_4}(Y) = \dots = f_Y(Y)$$

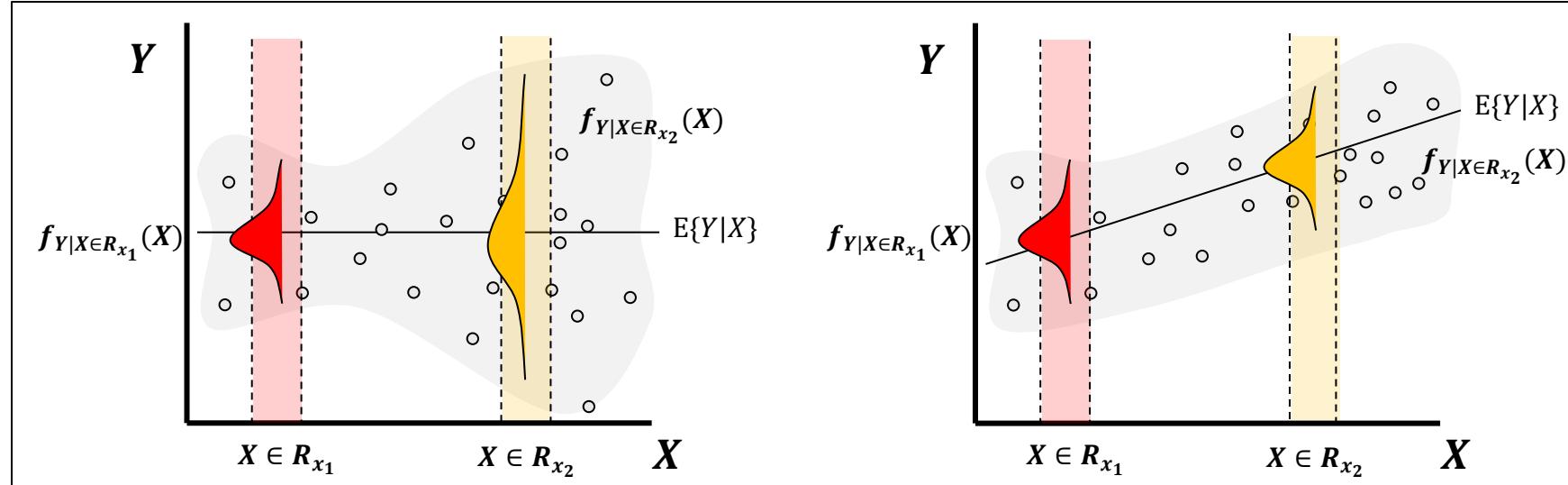
for any choice of bins x_1, x_2, x_3 and x_4 , the conditional distribution does not change from marginal, $f_Y(Y)$.



Mutual Information

Mutual Information

Examples of information sharing, general dependence, by visualization of conditional probabilities.



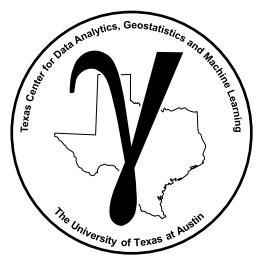
$$E\{Y|X \in R_{x_1}\} = E\{Y|X \in R_{x_2}\}$$

$$\text{Var}\{Y|X \in R_{x_1}\} \neq \text{Var}\{Y|X \in R_{x_2}\}$$

$$E\{Y|X \in R_{x_1}\} \neq E\{Y|X \in R_{x_2}\}$$

$$\text{Var}\{Y|X \in R_{x_1}\} = \text{Var}\{Y|X \in R_{x_2}\}$$

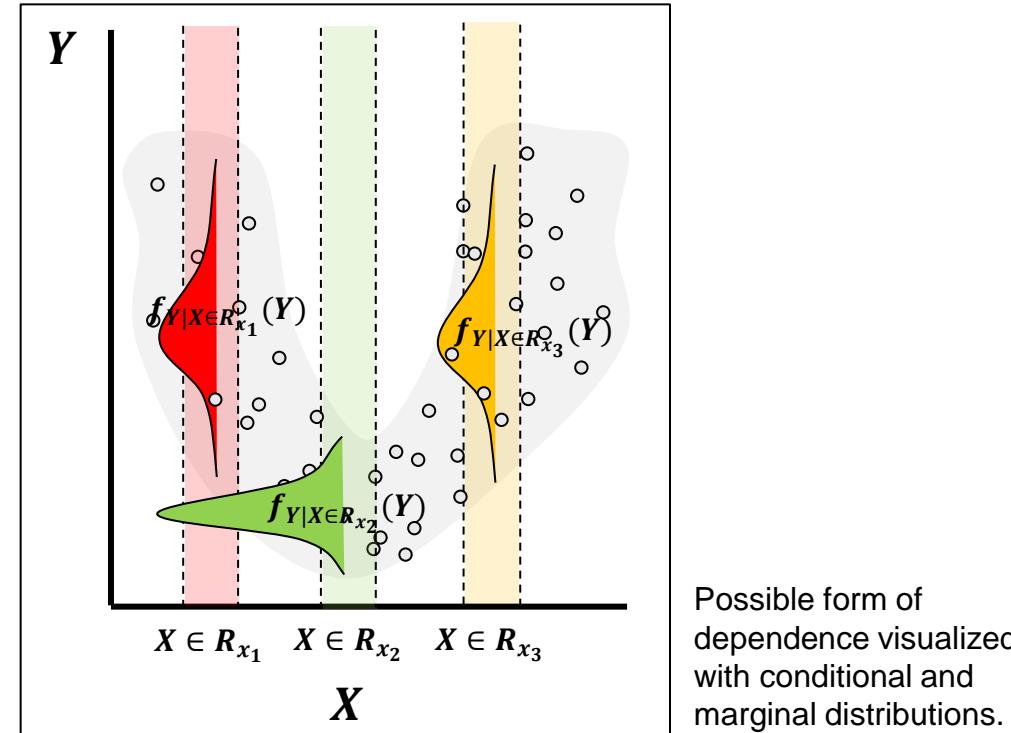
- If conditional distribution $f_{Y|X \in R_{x_1}}(y|X \in R_{x_1}) \neq f_{Y|X \in R_{x_2}}(y|X \in R_{x_2})$ for any choice of bins R_{x_1}, R_{x_2} .
- In more general terms, independence is $f_{Y|X \in R_{x_1}} = f_Y(Y)$



Mutual Information

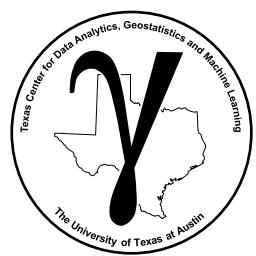
Mutual Information

Limitation of correlation analysis, assumes monotonic (rank), and linear (Pearson)



Possible form of dependence visualized with conditional and marginal distributions.

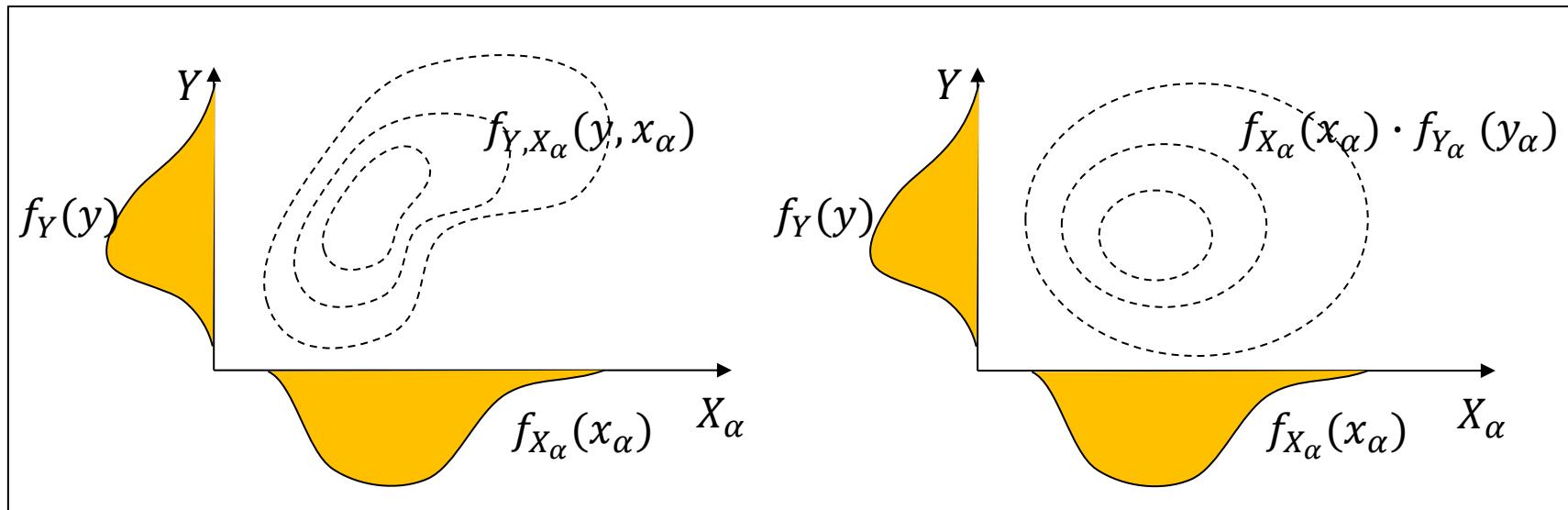
- For a quadratic relationship, $Y = X^2 + c$, highly nonlinear, nonmonotonic
 - $\rho_{X,Y} = 0$ and $\rho_{R_X, R_Y} = 0$! But, knowing about X , helps know about Y !



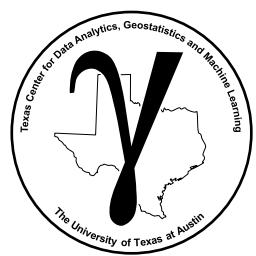
Mutual Information

Mutual Information – Proposed non-parametric metric

- Non-parametric model, i.e., no reliance on linearity given our previous observations
- Compare the actual joint density, $f_{Y,X_\alpha}(y, x_\alpha)$, to joint density assuming independence, $f_{X_\alpha}(x_\alpha) \cdot f_{Y_\alpha}(y_\alpha)$



Schematic of marginal and joint distributions for independence (left) and non-linear dependence (right).



Mutual Information

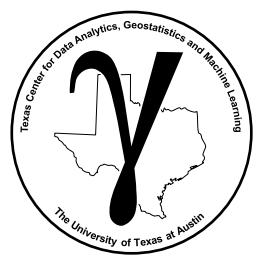
Mutual Information – Proposed non-parametric metric

- Non-parametric model, i.e., no reliance on linearity given our previous observations
- Compare the actual joint density, $f_{Y,X_\alpha}(y, x_\alpha)$, to joint density assuming independence, $f_{X_\alpha}(x_\alpha) \cdot f_{Y_\alpha}(y_\alpha)$

$$I(X_\alpha; Y) = \int_Y \int_{X_\alpha} P_{X_\alpha, Y}(x_\alpha, y) \cdot \log \left(\frac{P_{X_\alpha, Y}(x_\alpha, y)}{P_{X_\alpha}(x_\alpha) \cdot P_Y(y)} \right) dx dy$$

Weighting by local density **Measure of mismatch** if $\frac{P_{X_\alpha, Y}(x_\alpha, y)}{P_{X_\alpha}(x_\alpha) \cdot P_Y(y)} = 1$, $\log(1) = 0$

- Summary of mismatch between actual joint density and that expected from marginals with independence assumption weighted by local joint density

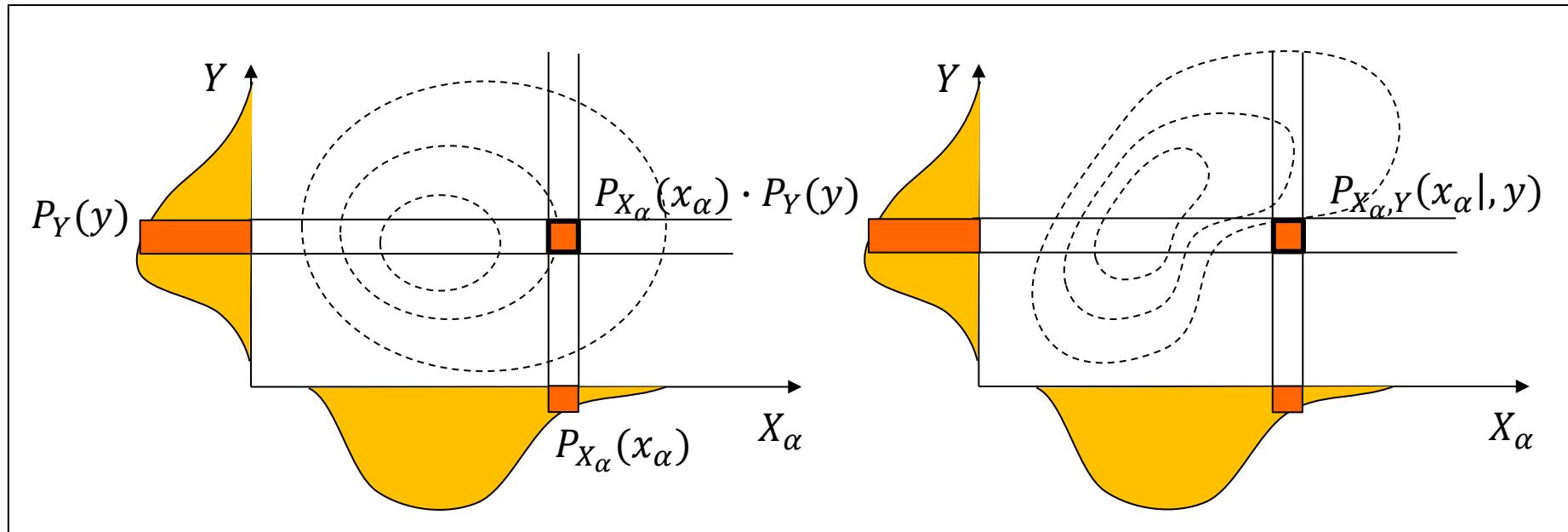


Mutual Information

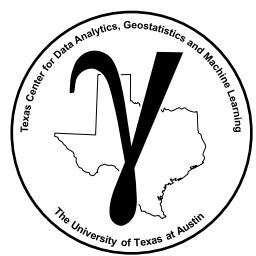
Mutual Information Calculation

Bin the joint PDF and sum over all combinations of X and Y bins.

$$I(X_\alpha; Y) = \sum_{y \in Y} \sum_{x \in X_\alpha} P_{X_\alpha, Y}(x_\alpha, y) \log \left(\frac{P_{X_\alpha, Y}(x_\alpha, y)}{P_{X_\alpha}(x_\alpha) \cdot P_Y(y)} \right)$$



Schematic of marginal and joint distributions for independence (left) and non-linear dependence (right).



Mutual Information

Mutual Information

Calculation given continuous Marginal PDFs, $f_{X_\alpha}(x_\alpha)$, $f_Y(y)$ and Joint PDF, $f_{X_\alpha,Y}(x_\alpha, y)$

$$I(X_\alpha; Y) = \int_Y \int_{X_\alpha} f_{X_\alpha,Y}(x_\alpha, y) \cdot \log \left(\frac{f_{X_\alpha,Y}(x_\alpha, y)}{f_{X_\alpha}(x_\alpha) \cdot f_Y(y)} \right) dx dy$$

Calculation given a sample dataset, with binning decision:

$$I(X_\alpha; Y) = \sum_{y \in Y} \sum_{x \in X_\alpha} P_{X_\alpha,Y}(x_\alpha, y) \log \left(\frac{P_{X_\alpha,Y}(x_\alpha, y)}{P_{X_\alpha}(x_\alpha) \cdot P_Y(y)} \right)$$

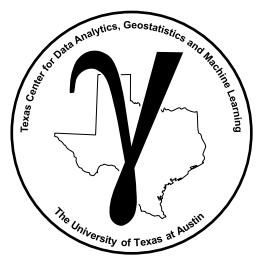
Properties of mutual Information,

- Nonnegativity:
- Symmetry:

$$I(X_\alpha; Y) \geq 0$$

1. amount of information must be ≥ 0
2. Jensen's Inequality
3. same as Kullback-Leibler

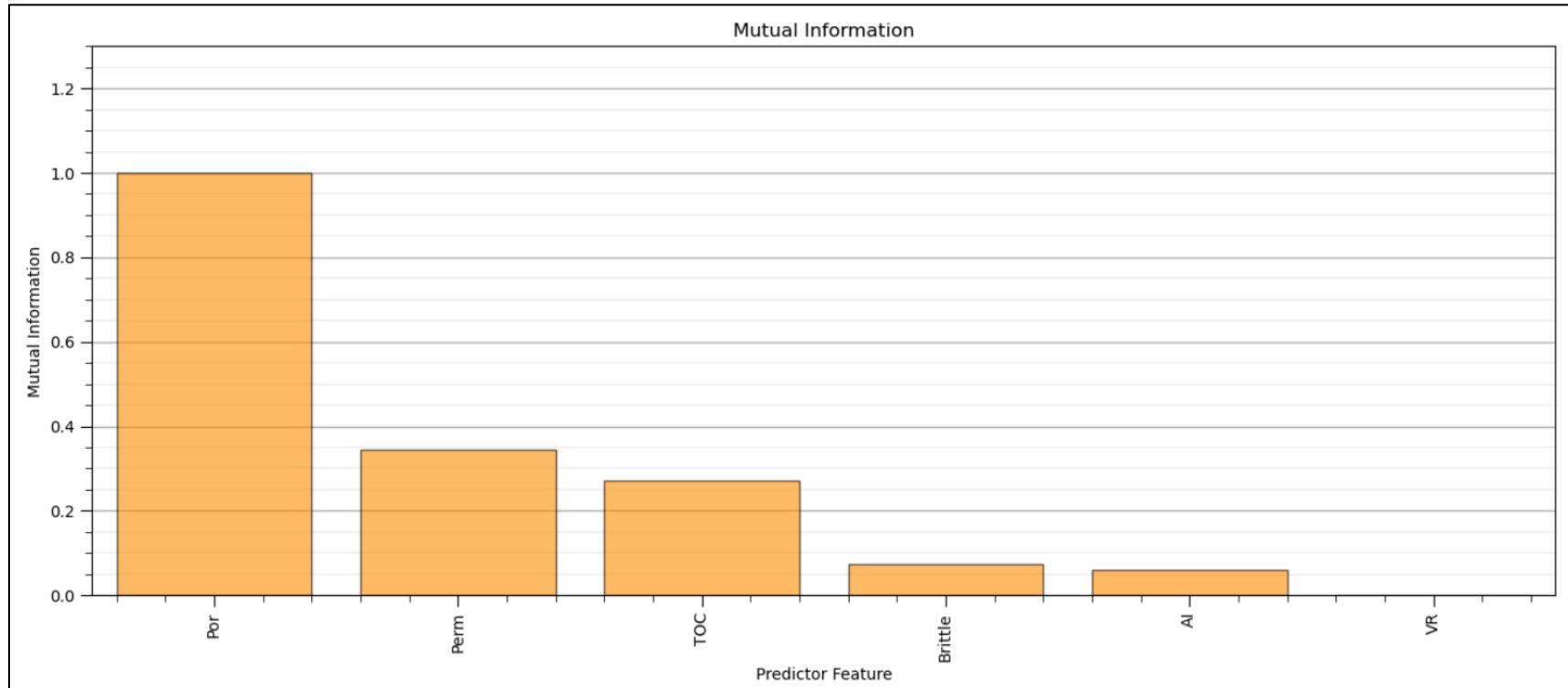
$$I(X_\alpha; Y) = I(Y; X_\alpha)$$



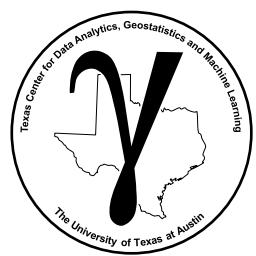
Mutual Information

Mutual Information

Pairwise mutual information, each predictor feature and the response feature, select features with highest mutual information with the response feature.



Normalized mutual information between production and each response feature. Maximum set to 1.0., From MachineLearning_feature_ranking chapter of e-book.



Mutual Information

Mutual Information and Entropy

Entropy, $H(\cdot)$, is a measure of uncertainty or amount of information needed to describe a system and mutual information may be related to marginal and conditional entropy.

$$H(Y) = - \sum_{y \in Y} P_Y(y) \log(P_Y(y)) \text{ and } H(X) = - \sum_{x \in X} P_X(x) \log(P_X(x))$$

$$H(Y|X) = \sum_{y \in Y} P_{Y|X}(y|x) \log(P_{Y|X}(y|x))$$

- We can now define mutual information relative to entropy

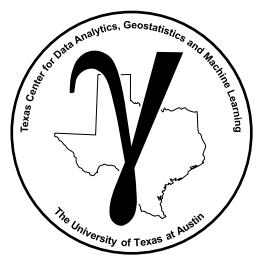
Uncertainty reduction
in Y given X is known

$$I(X_\alpha; Y) = H(Y) - H(Y|X_\alpha)$$

Uncertainty in Y

Uncertainty in Y given X is known





Mutual Information

Mutual Information for Feature Ranking with Max-Dependency Criteria (Peng et al., 2005).

Workflow to find the subset of features, S , p predictor features that maximize mutual information with a response feature, Y .

$$\max I(S, Y), D = (\{x_{p_1}, i = 1, \dots, p\}, Y),$$

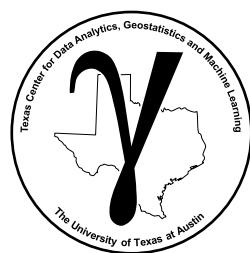
- For $p = 1$, must calculate the mutual information for all $I(X_i; Y)$ and select maximum.

$$I(X_{p_1}; Y) = \int_Y \int_{x_{p_1}} f_{X_{p_1}, Y}(x_{p_1}, y) \cdot \log \left(\frac{f_{X_{p_1}, Y}(x_{p_1}, y)}{f_{X_{p_1}}(x_{p_1}) \cdot f_Y(y)} \right) dx_{p_1} dy$$

- For $p > 1$, use incremental/one at a time trial and error approach, add one feature at a time and calculate. For example, given we found the first, X_{p_1} , we find the second:

$$I(X_{p_1}, X_{p_2}; Y) = \int_Y \int_{x_{p_1}} \int_{x_{p_2}} f_{X_{p_1}, X_{p_2}, Y}(x_{p_1}, x_{p_2}, y) \cdot \log \left(\frac{f_{X_{p_1}, X_{p_2}, Y}(x_{p_1}, x_{p_2}, y)}{f_{X_{p_1}, X_{p_2}}(x_{p_1}, x_{p_2}) \cdot f_Y(y)} \right) dx_{p_2} dx_{p_1} dy$$

- Note this can be difficult for high dimensional cases as there won't be sufficient data to sample the high dimensional joint distribution.



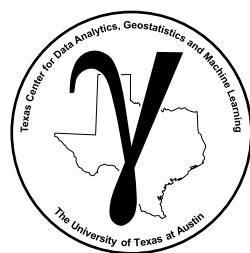
Mutual Information

Mutual Information for Feature Ranking with Max-Dependency Criteria (Peng et al., 2005)

1. Calculate all mutual information, $I(X_i; Y)$, for $i = 1, \dots, m$ predictor features. **Select the predictor feature, X_{p_1} , with maximum mutual information with response, Y .**
2. Calculate all joint mutual information, $I(X_{p_1}, X_i; Y)$, for $i = 1, \dots, m, i \neq p_1$. **Select the next predictor feature, $X_{p_2}, p_2 \neq p_1$**
3. **Continue until p predictor features are selected, X_1, \dots, X_p .**

This is a heuristic, short cut method, that avoids investigation of the full combinatorial of possible feature subsets.

- more on solution heuristics later



Mutual Information

Maximum Relevance Minimum Redundancy (MRMR) (Peng et al., 2005)

Peng et al. (2005) suggested a practical approach that avoids the high dimensional $I(\cdot)$

$$I(X_1, \dots, X_p; Y)$$

terms while accounting for the relevance and redundancy between predictor features.

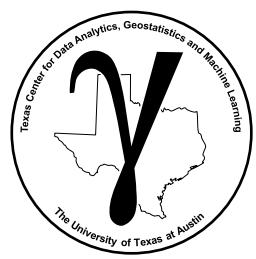
- For continuous predictor features and categorical response, first discretize the feature into a few states, $X_i \rightarrow X_{i,k}$, $Y \rightarrow Y_k$

$$MRMR = \max_S \left[\frac{1}{|S|} \sum_{X_i \in S} I(X_i, Y) - \frac{1}{|S|^2} \sum_{X_i \in S} \sum_{X_j \in S} I(X_i, X_j) \right]$$

Relevance of X_i with respect to Y

Redundancy of X_i with each other

where S is the predictor feature subset and $|S|$ is the number of features in the subset S .

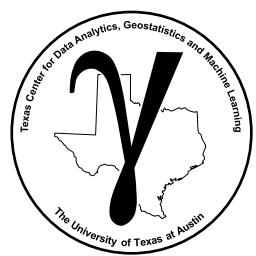


PGE 383 Subsurface Machine Learning

Lecture 5b: Feature Selection

Lecture outline:

- Shapley Values



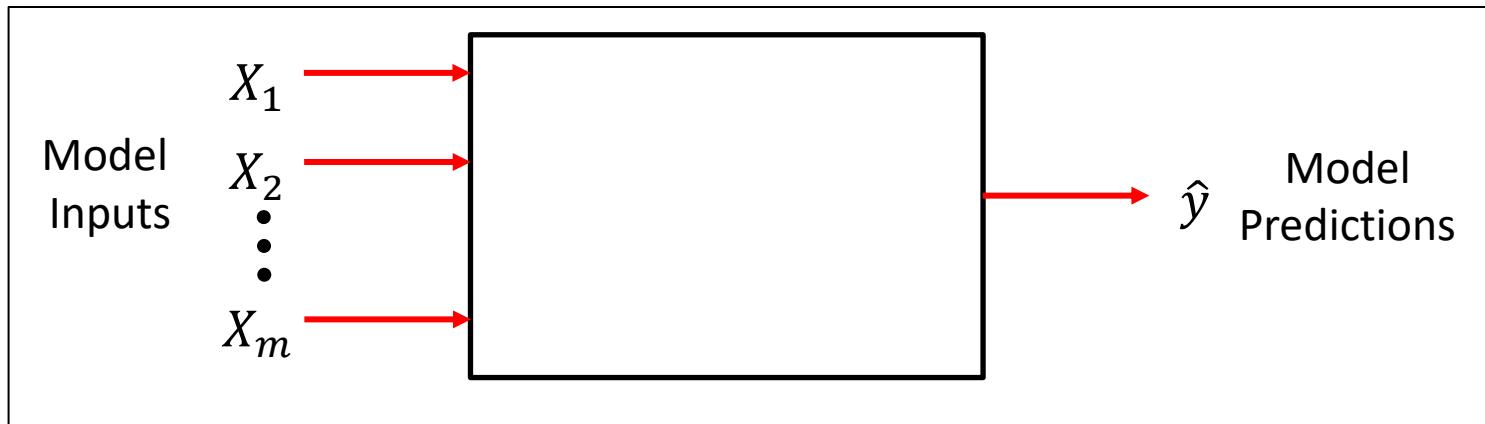
Shapley Values

Shapley Value Motivation

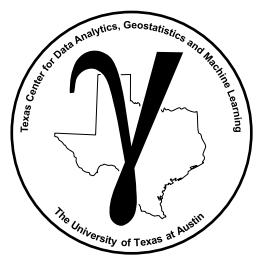
- **Model-based, Local and Global Feature Importance** by learning contribution of each feature to the prediction
- **Explainable Machine Learning:** complicated models are often required but have low interpretability.

Two choices to improve model interpretability:

1. reduce the complexity of the models, but may also reduce model accuracy
2. develop improved, agnostic (for any model) model diagnostics



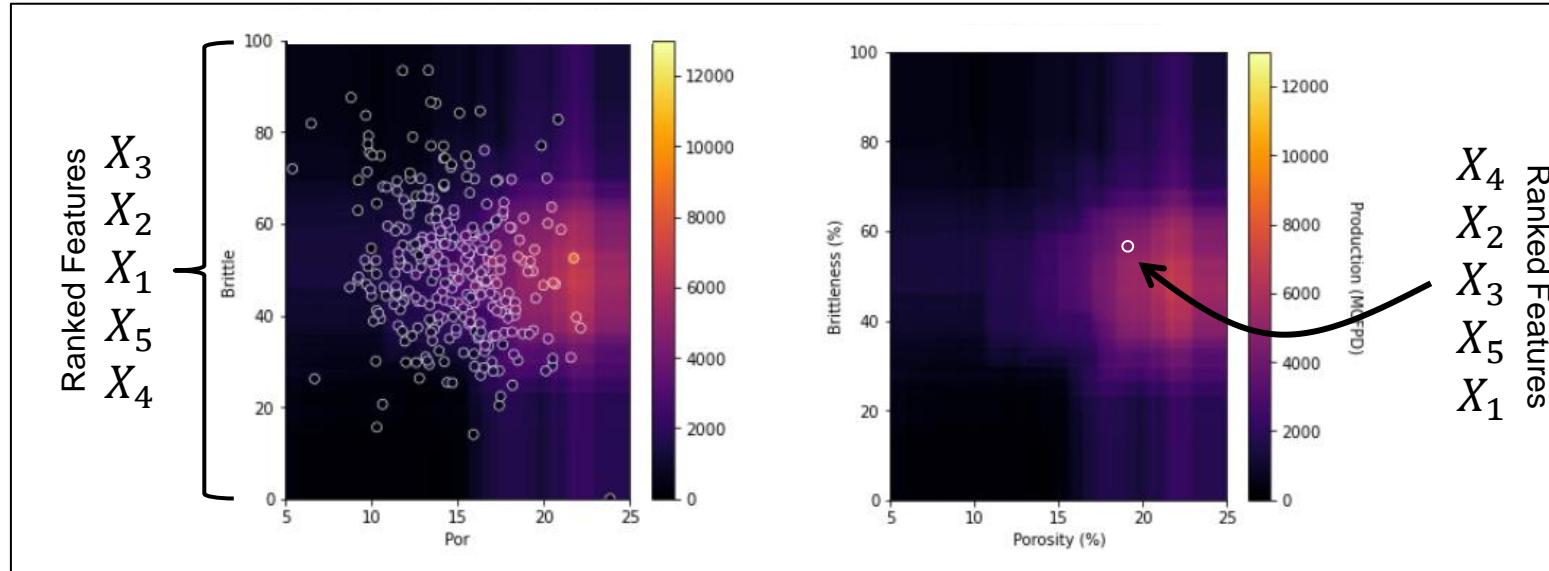
Shapley is a general method for feature importance, based on the model as a black box (model agnostic).



Shapley Values

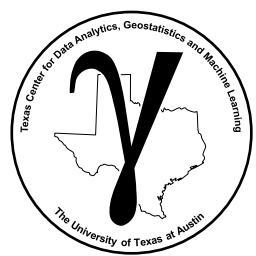
Local vs. Global Feature Importance

- **Global Feature Importance** – impact of each feature summarized over multiple locations, testing / withheld data
- **Local Feature Importance** – impact of each feature for a single prediction, 1 testing / withheld datum



Given a model we can calculate global feature importance over various testing, withheld data (left) or local feature importance for one testing datum (right).

- Other model-based feature importance measures summarize over all testing predictions, Global Feature Importance. Shapley is local but can be summarized for global feature importance.



Shapley Values

Shapley Values

- **Game theory approach**
 - Calculate the contribution of each predictor feature to push the response prediction away from the mean value of the response over training
 - Based on the Shapley value for allocating resources between ‘players’ based on a summarization of marginal contributions.

Dividing up payment between players.

- Marginal contributions and Shapley values are in units of the response feature.

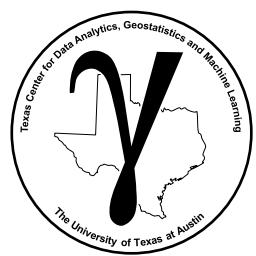
Marginal Contribution – the impact of one player on the result in a single case. For example:

1st order marginal contribution

Marginal Contribution Player 1 = Outcome Player 1 – Outcome No Players

2nd order marginal contribution

Marginal Contribution Player 1 = Outcome Player 1 & 2 – Outcome Player 2 Only



Shapley Values

What is the contribution of John Lennon to the Beatles?

4th order marginal contribution

The Beatles
- George + Paul + Ringo

Δ Records Sold

3rd order marginal contributions

John + George + Ringo
- George + Ringo

Δ Records Sold

John + Ringo + Paul
- Ringo + Paul

Δ Records Sold

John + Paul + George
- Paul + George

Δ Records Sold



The Beatles Abbey Road album cover silhouette
(from <https://www.etsy.com/listing/1168864036/the-beatles-logo>).

2nd order marginal contributions

John + Paul
- Paul Solo

Δ Records Sold

John + George
- George Solo

Δ Records Sold

John + Ringo
- Ringo Solo

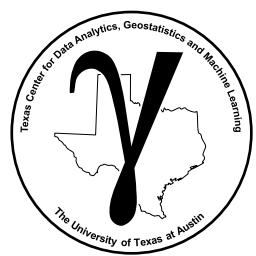
Δ Records Sold

1st order marginal contribution

John Solo
- None

Δ Records Sold

Shapley Value, John Lennon's contribution to The Beatles in records sold is the average of John's 1st, 2nd, 3rd and 4th order marginal contributions.



Shapley Values

Shapley Values

Contribution as the summarization / weighted average of all possible marginal contributions.

2 people work together and make \$125,000. How do we split the profits?

$$f(\text{person}_1) = \$50,000, f(\text{person}_2) = \$75,000, f(\text{person}_1, \text{person}_2) = \$125,000$$

Allocation person 1: $\frac{1}{2}f(\text{person}_1) + \frac{1}{2}(f(\text{person}_1, \text{person}_2) - f(\text{person}_2))$

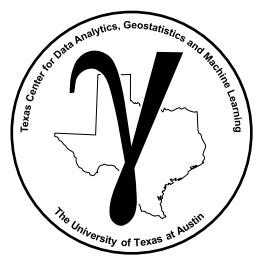
$$\frac{1}{2}(\$50k - 0) + \frac{1}{2}(\$125k - \$75k) = \$50k$$

Allocation person 2: $\frac{1}{2}f(\text{person}_2) + \frac{1}{2}(f(\text{person}_1, \text{person}_2) - f(\text{person}_1))$

$$\frac{1}{2}(\$75k - 0) + \frac{1}{2}(\$125k - \$50k) = \$75k$$

Solution:

Pay person 1 \$50k and person 2 \$75k of the total \$125k.



Shapley Values

Shapley Values

Contribution as the summarization / weighted average of all possible marginal contributions.

2 people work together and make \$150,000. How do we split the profits?

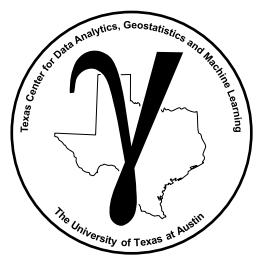
Synergy

$$f(\text{person}_1) = \$50,000, f(\text{person}_2) = \$70,000, f(\text{person}_1, \text{person}_2) = \$150,000$$

$$\begin{aligned} \text{Allocation person 1: } & \frac{1}{2}f(\text{person}_1) + \frac{1}{2}(f(\text{person}_1, \text{person}_2) - f(\text{person}_2)) \\ & \frac{1}{2}(\$50k - 0) + \frac{1}{2}(\$150k - \$70k) = \$65k \\ \text{Allocation person 2: } & \frac{1}{2}f(\text{person}_2) + \frac{1}{2}(f(\text{person}_1, \text{person}_2) - f(\text{person}_1)) \\ & \frac{1}{2}(\$70k - 0) + \frac{1}{2}(\$150k - \$50k) = \$85k \end{aligned}$$

Solution:

Pay person 1 \$65k and person 2 \$85k of the total \$150k.



Shapley Values

Shapley Values

Contribution as the summarization / weighted average of all possible marginal contributions.

2 people work together and make \$90,000. How do we split the profits?

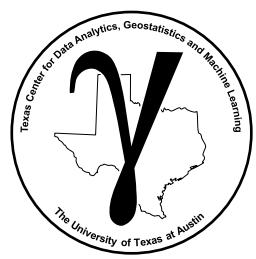
Dynergy

$$f(\text{person}_1) = \$50,000, f(\text{person}_2) = \$70,000, f(\text{person}_1, \text{person}_2) = \$90,000$$

$$\begin{aligned} \text{Allocation person 1: } & \frac{1}{2}f(\text{person}_1) + \frac{1}{2}(f(\text{person}_1, \text{person}_2) - f(\text{person}_2)) \\ & \frac{1}{2}(\$50k - 0) + \frac{1}{2}(\$90k - \$70k) = \$35k \\ \text{Allocation person 2: } & \frac{1}{2}f(\text{person}_2) + \frac{1}{2}(f(\text{person}_1, \text{person}_2) - f(\text{person}_1)) \\ & \frac{1}{2}(\$70k - 0) + \frac{1}{2}(\$90k - \$50k) = \$55k \end{aligned}$$

Solution:

Pay person 1 \$35k and person 2 \$55k of the total \$90k.

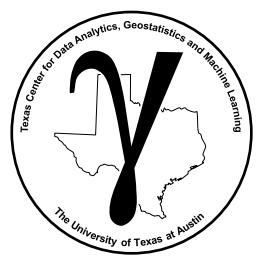


Shapley Values

Shapley Values

We work out the contribution of each player through summarization over marginal contributions.

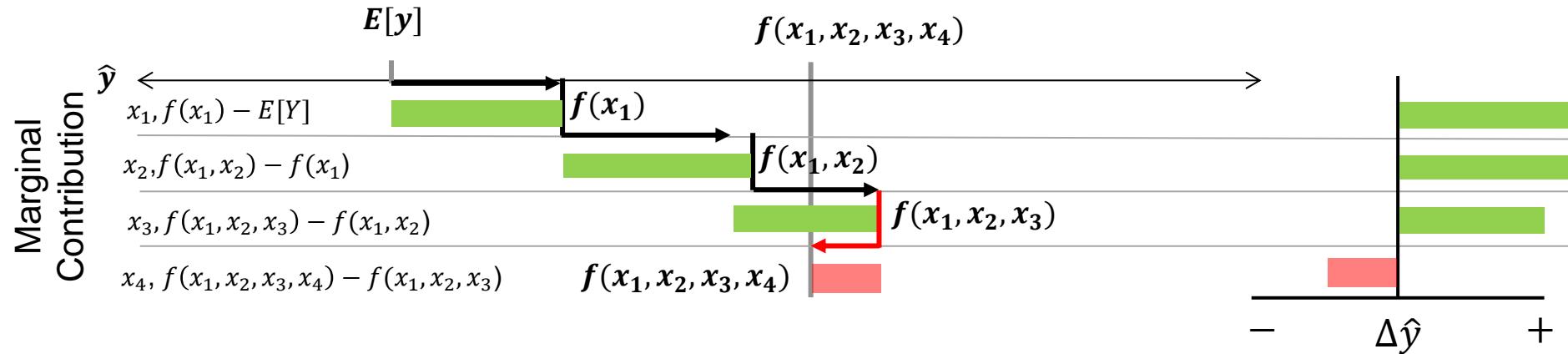
- now change player → feature, x_i
- and earnings → model prediction, $f(x)$



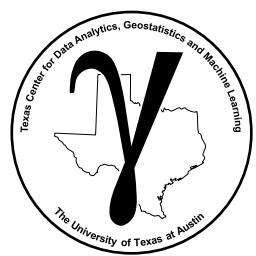
Shapley Values

Feature Contribution via Local Feature Importance

Local Feature Importance – representing a specific prediction case (x_1, \dots, x_m) .

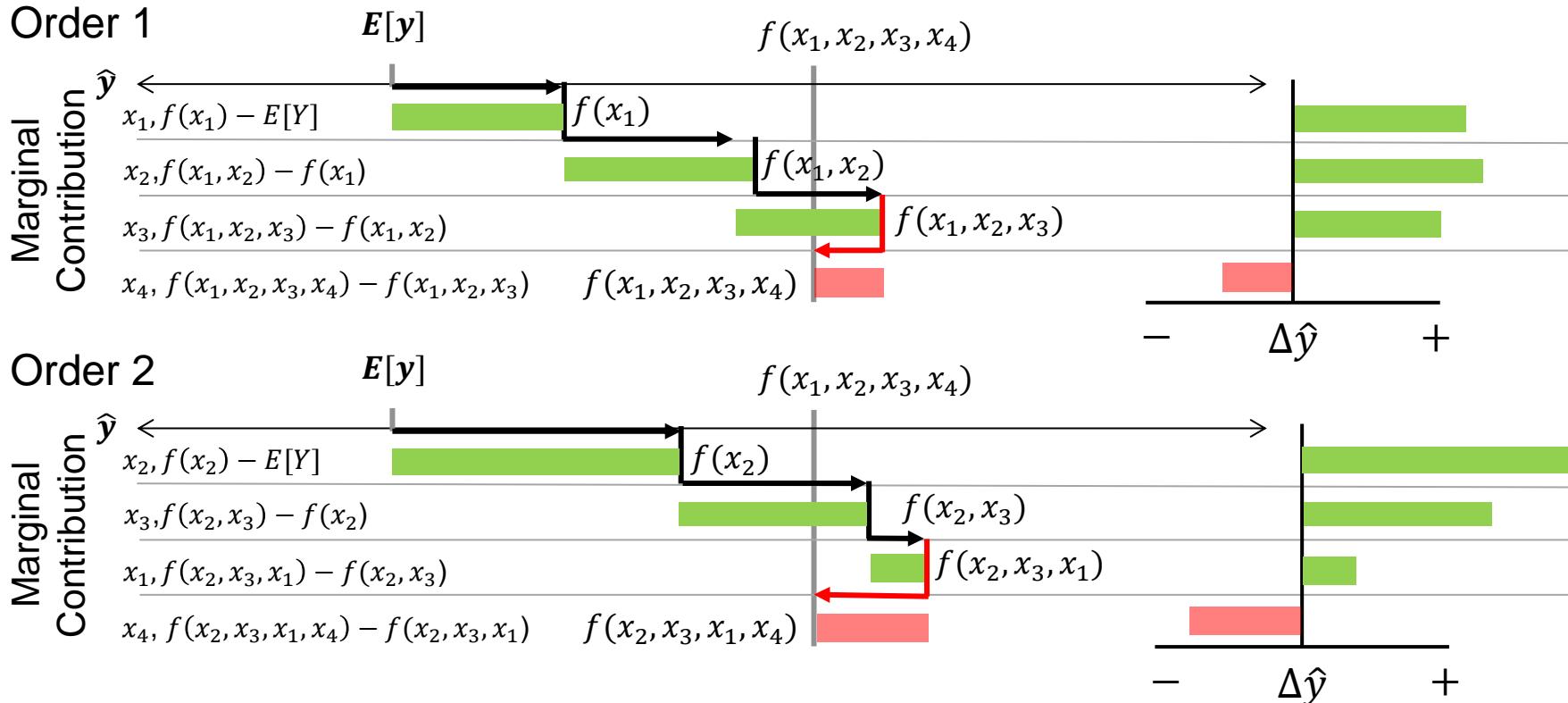


- Recall $E[y]$ is the expectation of all response training values, i.e. no information from the predictor features.
- Issues:
 1. What if we change the order of adding features?
 2. We do not want to build multiple models, we want to assess feature importance for one specific model, $f(x_1, x_2, x_3, x_4)$.
 3. We may want a global importance for all possible predictions, not a specific case, x_1, \dots, x_m .

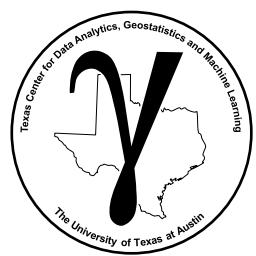


Shapley Values

The Combinatorial of Feature Contributions

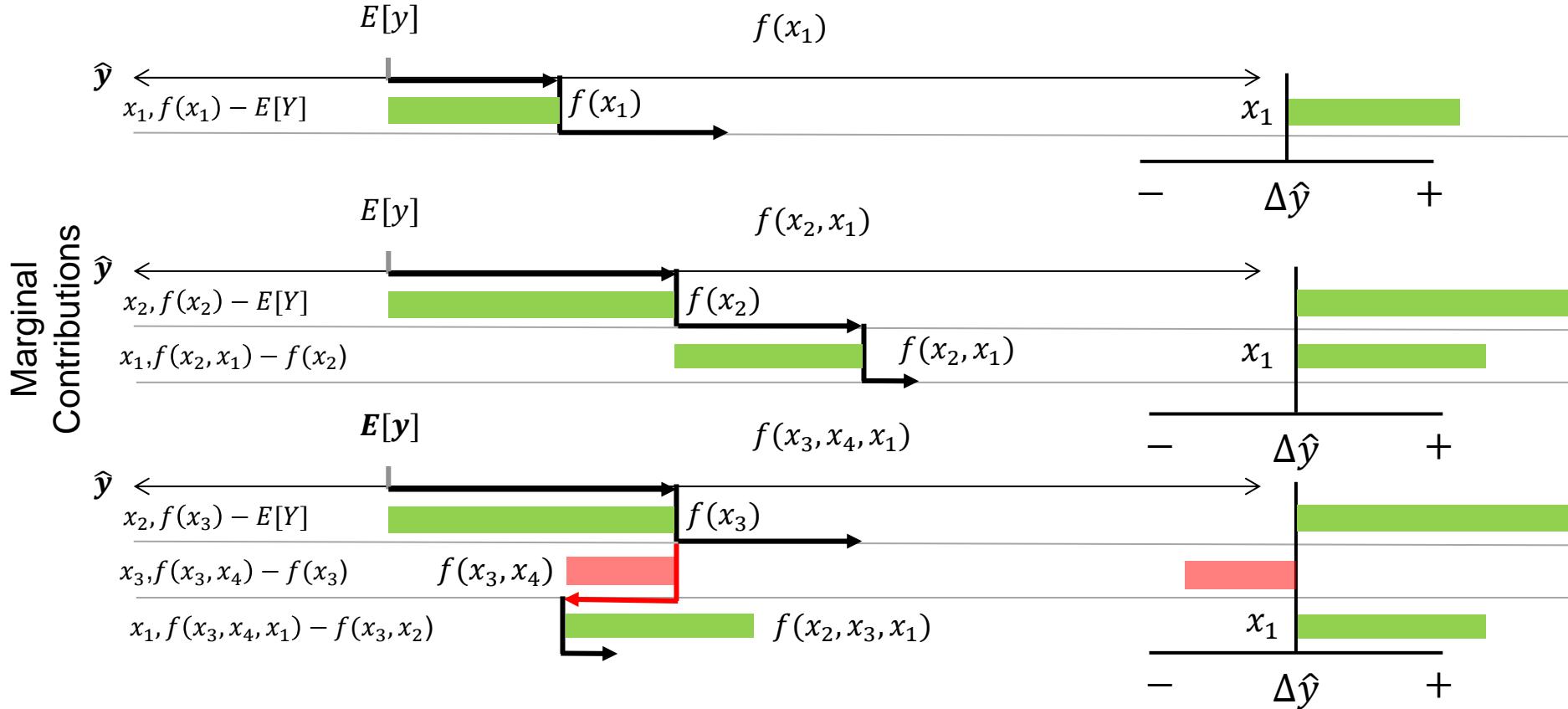


- Due to interactions between predictor features. The order matters!
- We average the marginal contribution over the combinatorial of orders

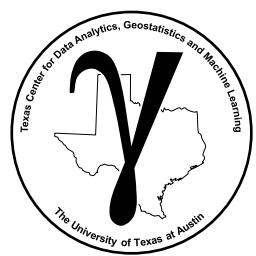


Shapley Values

The Combinatorial of Feature Contributions includes these models with 1, 2, 3 feature subsets:



- We will average the marginal contribution over the combinatorial of orders feature subsets.



Shapley Values

Shapley Value Issue, Multiple Models or Feature Imputation

We need to take a single model, $f(x_1, x_2, x_3, x_4)$, and make an estimate for all possible combinations feature subsets!

Note: the **naïve approach** is to train the full combinatorial of models. We don't want to do that if our goal is feature importance to diagnose our model, f . We want to support model explain-ability.

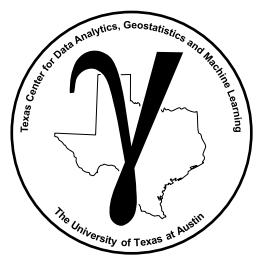
The variety of approaches are similar to imputation methods:

$$f(x_1, x_2, x_3) = f(x_1, x_2, x_3, x_4 = E[x_4])$$

$$f(x_1, x_2, x_3) = f(x_1, x_2, x_3, x_4 = P50_{x_4})$$

There is a unique method with tree-based models,

- Remove x_4 by averaging response prediction over all branches with x_4 .



Shapley Values

Shapley Equation

Averaging over all possible subsets, orders of marginal contribution

- Shapley value, ϕ_i , for the local importance of the i feature:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|! (|F| - |S| - 1)}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

where:

$|S|$ size of the subset before we add the i^{th} feature

$|F|$ number of features

$[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$ the marginal contribution of i^{th} feature

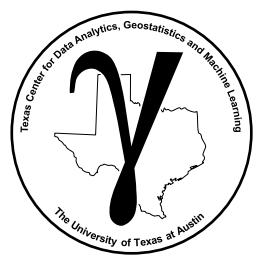
$\frac{|S|! (|F| - |S| - 1)}{|F|!}$ is the weighting by the combinations for this occurrence

$S \subseteq F \setminus \{i\}$ is all possible subsets without i feature, so we can add i

$S \cup \{i\}$ is subset S with i added and S is a subset without i

Our model prediction
with i feature

Our model prediction
without i feature



Shapley Values

Shapley Equation

Let's explain the weighting applied to each case.

$$\frac{|S|! (|F| - |S| - 1)!}{|F|!}$$

Example $F = X_1, X_2, X_3, X_4$, $S = X_1, X_2$, $i = X_3$

Weight by the number
same / reorder S cases
divided by the total number
of possible combinations.

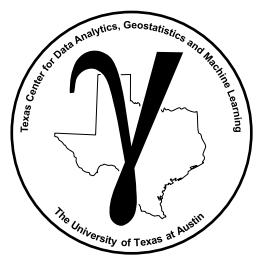
$$X_1, X_2 + X_3 = X_2, X_1 + X_3$$

so we can just say 2x weight.

$$|S|! (|F| - |S| - 1)! : 2 - X_1, X_2, \textcolor{red}{X}_3, X_4 \text{ and } X_2, X_1, \textcolor{red}{X}_3, X_4$$

$$2! 1! = 2$$

$$|F|! : 24 - [X_1, X_2, \textcolor{red}{X}_3, X_4], [X_1, \textcolor{red}{X}_3, X_2, X_4], \dots, [X_4, \textcolor{red}{X}_3, X_2, X_1]$$
$$4! = 24$$



Shapley Values

Shapley Equation

$$\sum_{S \subseteq F \setminus \{i\}}$$

We sum over all possible subsets without i , example subsets:

$$i = X_3, |F| = 3$$

Let's assume values, $x_1 = 10\%$, $x_2 = 150 \text{ mD}$, $x_3 = 13\%$, $x_4 = 0.54$

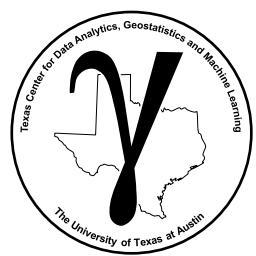
$$f(X_3 = x_3) - E[y] \longrightarrow f(X_1 = \bar{x}_1, X_2 = \bar{x}_2, X_3 = x_3) - E[y]$$

$$f(X_1 = x_1, X_3 = x_3) - f(X_1 = x_1) \longrightarrow f(X_1 = x_1, X_2 = \bar{x}_2, X_3 = x_3) - f(X_1 = x_1, X_2 = \bar{x}_2, X_3 = \bar{x}_3)$$

$$f(X_1 = x_1, X_2 = x_2, X_3 = x_3) - f(X_1 = x_1, X_2 = x_2) \longrightarrow f(X_1 = x_1, X_2 = x_2, X_3 = x_3) - f(X_1 = x_1, X_2 = x_2, X_3 = \bar{x}_3)$$

with feature X_3

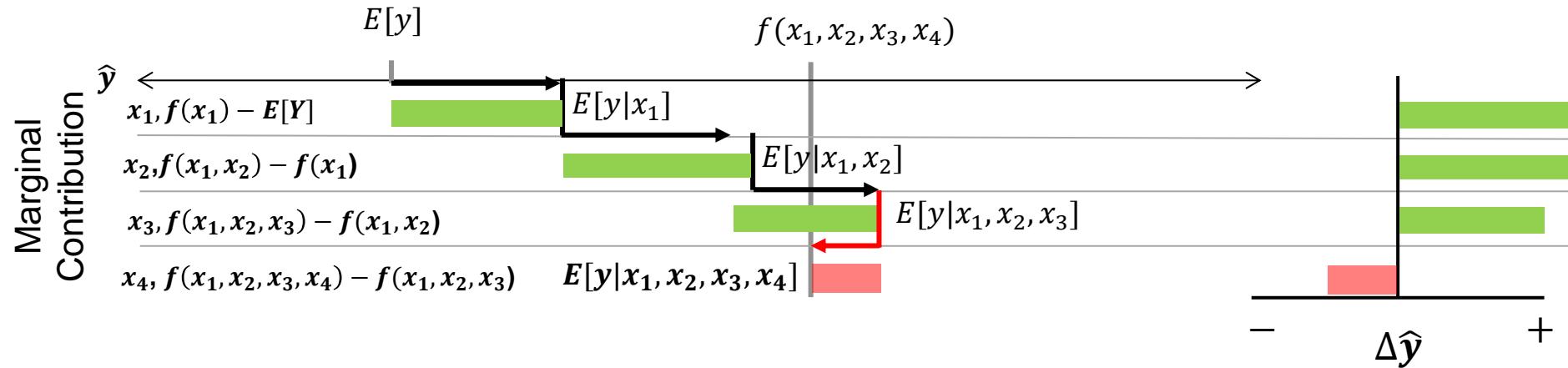
without feature X_3



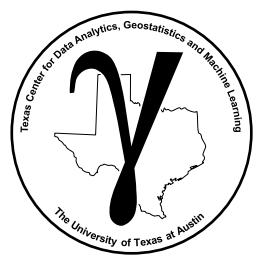
Shapley Values

Feature Contribution via Local Feature Importance

Local Feature Importance – representing a specific prediction case (x_1, \dots, x_m) .



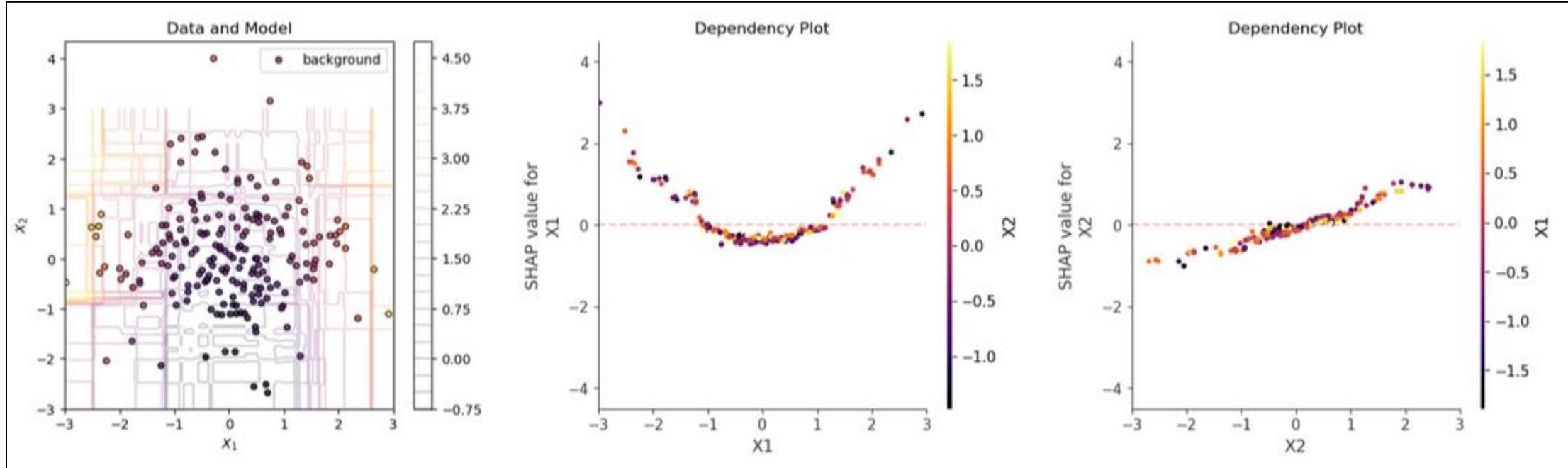
- Recall $E[y]$ is the expectation of all response training values, i.e., the best prediction if no information from the predictor features for a L2, sum of squared or mean square error loss norm.



Shapley Values

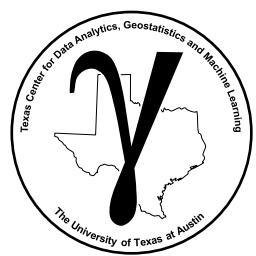
Local Feature Importance

Scatter plot the local feature importance, Shapley value, over all feature values, X_i



- Check for local feature importance over a range of values
- Observed the scatter due to interactions with other features
- Label with additional features

Data and model (left), and Shapley dependency plots for each predictor feature revealing the relationship between each predictor feature and the model predictions. From Interactive_Shapley.ipynb.

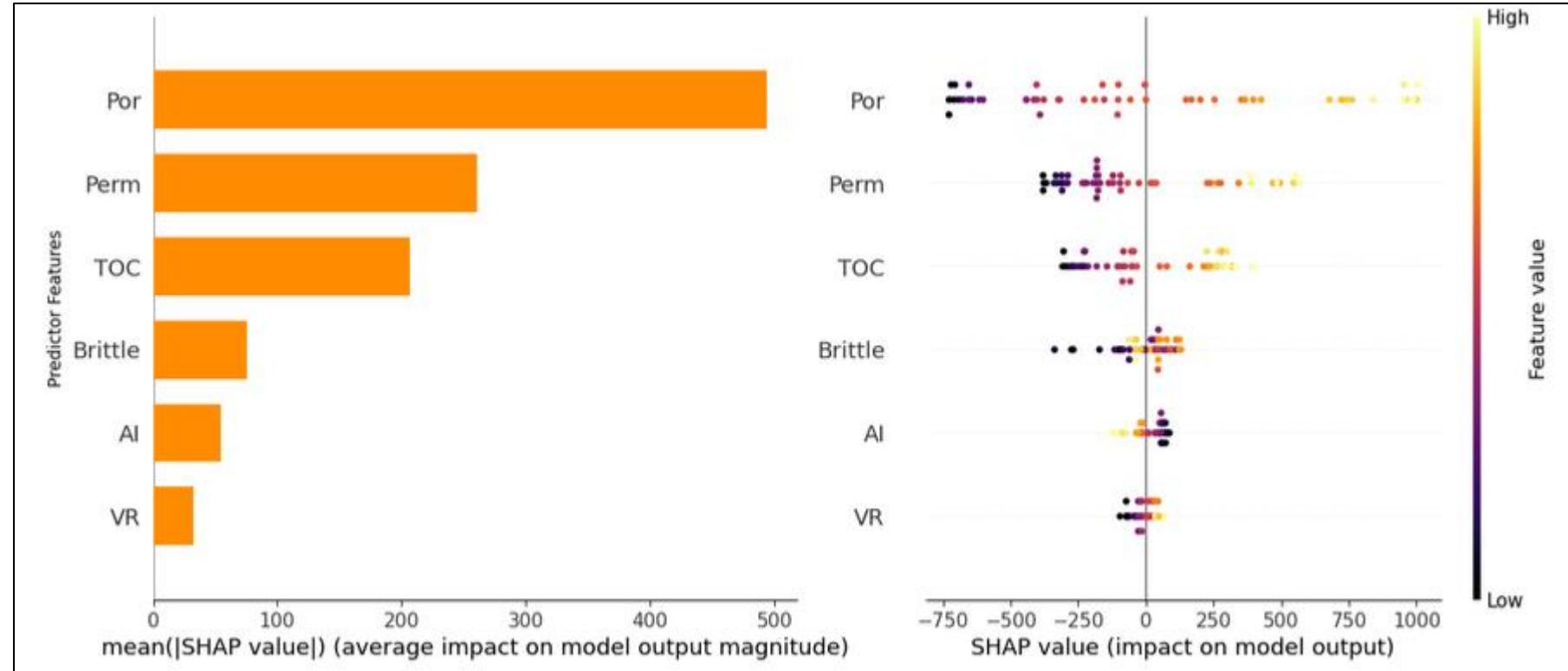


Shapley Values

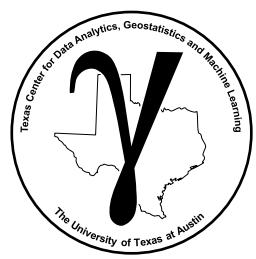
Global Feature Importance

Plot the local feature importance over all estimates, \hat{y} .

- Check for consistent SHAP and feature values, e.g., high predictor results in high prediction
- Summarize with the average SHAP value over all estimates.



Shapley-based feature ranking by mean absolute Shapley value. From MachineLearning_feature_ranking chapter of e-book.

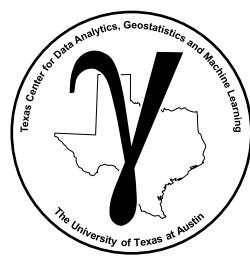


PGE 383 Subsurface Machine Learning

Lecture 5b: Feature Selection

Lecture outline:

- Recursive Feature Selection



Recursive Feature Elimination

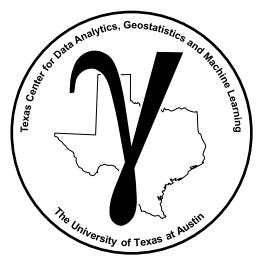
Recursive Feature Elimination

Recursive Feature Elimination (RFE) method works by recursively removing features and building a model with the remaining features.

- build a model with all features, calculate coefficient or feature importance (depending on which is available with the modeling method)
- remove the feature with the lowest coefficient or feature importance and rebuild the model
- repeat the process until only one feature remains

Any model could be used,

- the method assigns rank $1, \dots, m$ for all features as reverse order of removal, i.e., last remaining feature is most important and first removed is least important



Recursive Feature Elimination

Assumptions and Issues

Selection Bias – with many predictor features it is likely a few are randomly correlated with the response feature

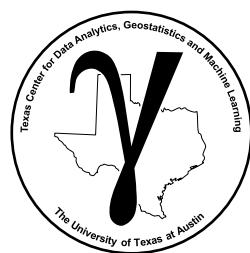
- they will rank highly and be retained
- there is a limited sampling of the feature interactions
- it would require a more complete sampling of predictor feature combinations to establish that the predictor feature is uninformative

- Also, there is a risk model overfit

More robust if resampling and validation are applied for each model

- Cross validation and the bootstrap

- An alternative is to check all possible combinations of features, this is not feasible if the combinatorial is very large



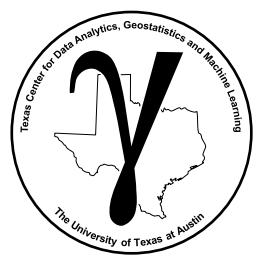
Recursive Feature Elimination

Recursive Feature Elimination

The recursive feature elimination method with a linear regression model provides these ranks:

1. Total Organic Carbon
2. Vitrinite Reflectance
3. Acoustic Impedance
4. Porosity
5. Permeability
6. Brittleness

There has been quite a bit of change from our previous metrics, let's use a more flexible model.



Recursive Feature Elimination

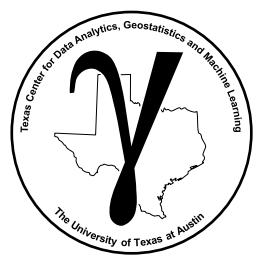
Ranking Methods – Recursive Feature Elimination

The recursive feature elimination method with a random forest model provides these ranks:

1. Porosity
2. Brittleness
3. Vitrinite Reflectance
4. Permeability
5. Total Organic Carbon
6. Acoustic Impedance

This is more consistent with our previous results. The advantages with the recursive elimination method:

- the actual model can be used in assessing feature ranks
- the ranking is based on the contribution of each feature to the model

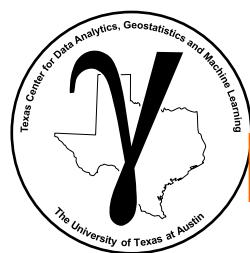


PGE 383 Subsurface Machine Learning

Lecture 5b: Feature Selection

Lecture outline:

- Feature Selection Hands-on



Feature Ranking Demonstration in Python

Demonstration of the wide array approach for feature selection with a well-documented workflow.

The screenshot shows the 'Feature Ranking' chapter from the 'Applied Machine Learning in Python: a Hands-on Guide with Code' e-book. The page has a sidebar on the left listing various machine learning concepts and techniques, with 'Feature Ranking' currently selected. The main content area displays the chapter's title, author (Michael J. Pyrcz), and a brief description. It includes sections for citing the e-book and GitHub repository, a note about workflows, and a YouTube lecture link. A large sidebar on the right contains a table of contents with many items related to machine learning and data analysis.

Applied Machine Learning in Python: a Hands-on Guide with Code

Machine Learning Concepts
Workflow Construction and Coding
Probability Concepts
Loading and Plotting Data and Models
Univariate Analysis
Multivariate Analysis
Feature Transformations
Feature Ranking
Cluster Analysis
Density-based Clustering
Spectral Clustering
Principal Components Analysis
Multidimensional Scaling
Linear Regression
Ridge Regression
LASSO Regression
Bayesian Linear Regression
Naive Bayes
Polynomial Regression
k-Nearest Neighbors
Decision Trees
Bagging Tree and Random Forest
Gradient Boosting
Support Vector Machines
Time Series Analysis and Modeling

Feature Ranking

Michael J. Pyrcz, Professor, The University of Texas at Austin

[Twitter](#) | [GitHub](#) | [Website](#) | [GoogleScholar](#) | [Book](#) | [YouTube](#) | [Applied Geostats in Python e-book](#) | [LinkedIn](#)

Chapter of e-book "Applied Machine Learning in Python: a Hands-on Guide with Code".

Cite this e-Book as:

Pyrcz, M.J., 2024, Applied Machine Learning in Python: a Hands-on Guide with Code, https://geostatsguy.github.io/MachineLearningDemos_Book.

The workflows in this book and more are available here:

Cite the MachineLearningDemos GitHub Repository as:

Pyrcz, M.J., 2024, MachineLearningDemos: Python Machine Learning Demonstration Workflows Repository (0.0.1). Zenodo. DOI [10.5281/zenodo.13835318](https://doi.org/10.5281/zenodo.13835318)

By Michael J. Pyrcz
© Copyright 2024.

This chapter is a tutorial for / demonstration of Feature Ranking.

YouTube Lecture: check out my lectures on:

- [Introduction to Machine Learning](#)
- [Curse of Dimensionality, Dimensionality Reduction, Principal Component Analysis](#)
- [Multidimensional Scaling and Random Projection](#)
- [Feature Transformations](#)

These lectures are all part of my [Machine Learning Course](#) on YouTube with linked well-documented Python workflows and interactive dashboards. My goal is to share accessible, actionable, and repeatable educational content. If you want to know about my motivation, check out [Michael's Story](#).

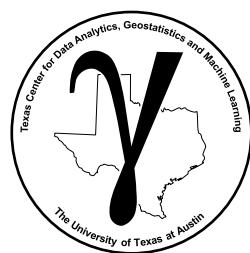
Motivation for Feature Ranking

There are often many predictor features (input variables), available for us to work with for building our prediction models. There are good reasons to be selective, throwing in every possible feature is not a good idea! In general, for the best prediction model, careful selection of the fewest features that provide the most amount of information is the best practice.

Contents

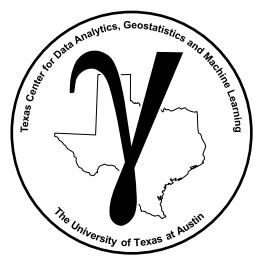
- Motivation for Feature Ranking?
- Load the Required Libraries
- Design Custom Color Map
- Declare Functions
- Set the Working Directory
- Loading Tabular Data
- Visualize the DataFrame
- Summary Statistics for Tabular Data
- Coverage
- Summary Statistics
- Univariate Distributions
- Bivariate Distributions
- Pairwise Covariance
- Pairwise Correlation Coefficient
- Pairwise Spearman Rank Correlation Coefficient
- Partial Correlation Coefficient
- Semipartial Correlation Coefficient
- Feature Ranking with Feature Transformations
- Conditional Statistics
- Variance Inflation Factor (VIF)
- β Coefficients / Beta Weights
- β Coefficients / Beta Weights
- Feature Importance
- Mutual Information
- Mutual Information Accounting For Relevance and Redundancy OFAT
- Variants
- Summary of All Bivariate Metrics
- Recursive Feature Elimination
- Shapley Values for Feature Ranking
- Comments
- The Author:
- Want to Work Together?
- More Resources Available at: Twitter | GitHub | Website | GoogleScholar | Book | YouTube | Applied Geostats in Python e-book | LinkedIn

MachineLearning_feature_ranking chapter of e-book.



Multivariate New Tools

Topic	Application to Subsurface Modeling
Curse of Dimensionality	<p>Reduce problem to lowest dimension possible.</p> <p><i>Feature ranking determined that porosity may be predicted from acoustic impedance and rock type alone.</i></p>
Feature Selection	<p>Apply wide array methods to explore the importance of each predictor feature with respect to the response feature.</p> <p><i>Partial correlation reveals that rock type provides little additional information to acoustic impedance.</i></p>



PGE 383 Subsurface Machine Learning

Lecture 5b: Feature Selection

Lecture outline:

- Curse of Dimensionality
- Feature Selection
- Mutual Information
- Shapley Values
- Recursive Feature Selection
- Feature Selection Hands-on