PGE 383 Subsurface Machine Learning

Lecture 9: Linear Regression

Lecture outline:

- Linear Regression
- L^1 and L^2 Norms
- Linear Regression Training
- Linear Regression Assumptions and Limitations
- Linear Regression Diagnostics
- Linear Regression Hands-on

Announcements

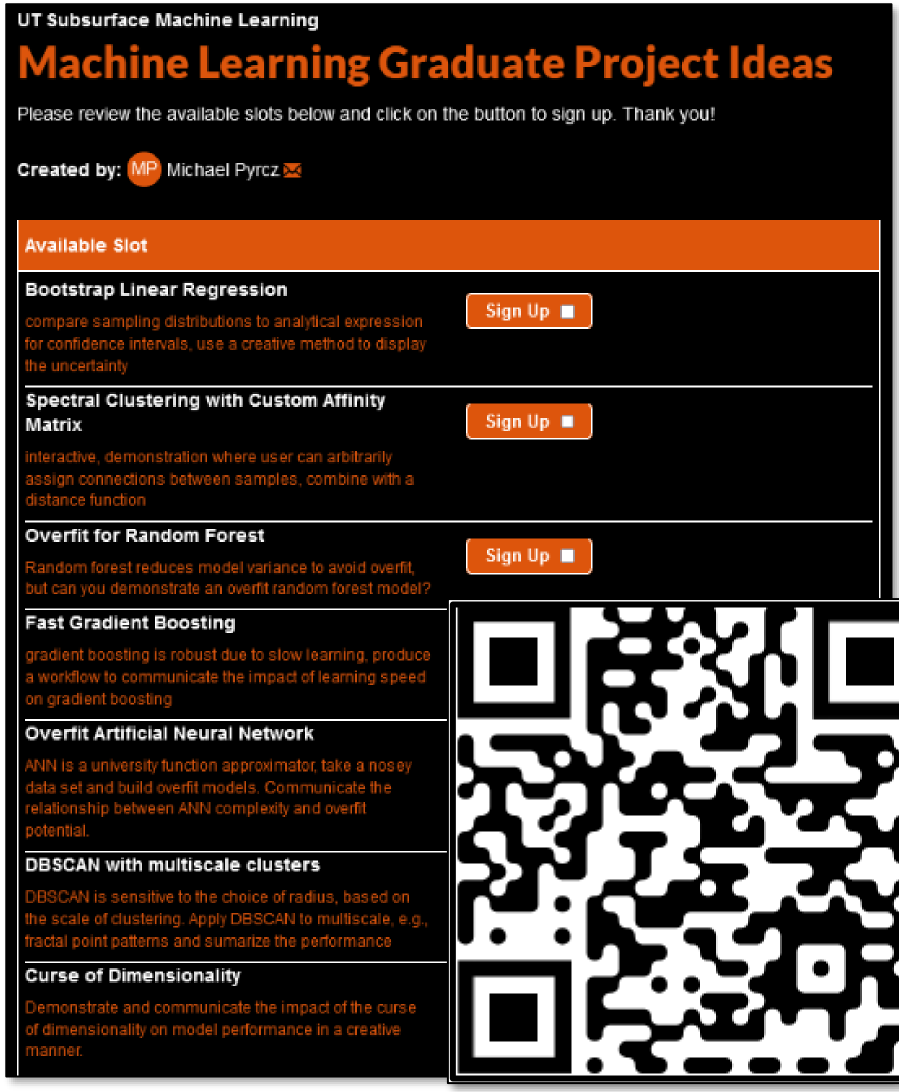
Start thinking about topics.

Goal learn and build a tutorial, education content.

Ideas:

- Test a concept
- Demonstrate a machine not currently in the course
- Demonstrate a limitation
- Compare machines
- Break a machine
- Visualize, Animate Machines
- Learn and Have Fun

See my wish list. →



https://www.signupgenius.com/go/508044FAFA72BA3F94-55631931-machine

Motivation

Let's start with the simplest prediction model and explore the concepts of:

- what is a machine?
- norms / model training
- model variance
- model bias
- confidence intervals
- prediction intervals

We will build from here to ridge regression and the LASSO.

PGE 383 Subsurface Machine Learning

Lecture 9: Linear Regression

Lecture outline:

- Linear Regression

Recall Machine Learning

toolkit

"… is the study of algorithms and mathematical models

learning

that computer systems use to

progressively improve their performance on a specific task.

Machine learning algorithms build a mathematical model

training with data

of sample data, known as "training data",

general

in order to make predictions or decisions

without being explicitly programmed to perform the task."

not a panacea

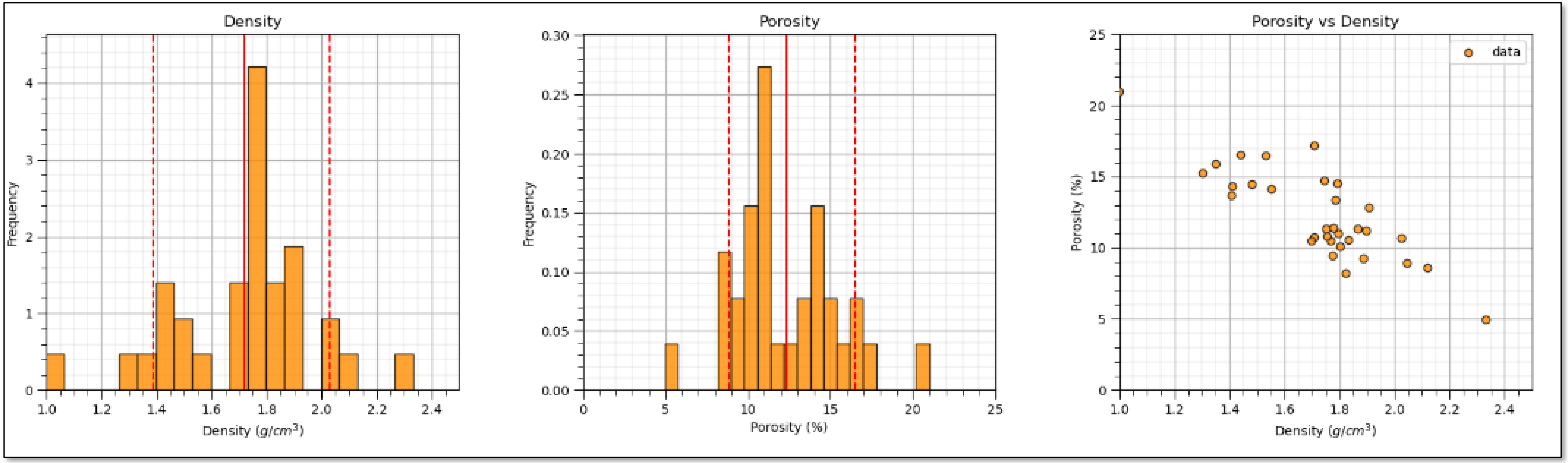
"… where it is

infeasible to develop an algorithm of specific instructions for performing the task."

Quote from Machine Learning - Wikipedia

Our First Prediction Machine

Loaded up a simple porosity vs. density dataset in Python.



Density (left) and porosity histograms (center) and scatter plot porosity and density (right), from Linear Regression chapter of Applied Macchine Learning in Python e-book.

Our First Prediction Machine

Run one line of Python to build a linear regression model

Let's first calculate the linear regression model

```
slope, intercept, r_value, p_value, std_err = st.linregress(den,por)

print('The model parameters are, slope (b1) = ' + str(round(slope,2)) + ', and the intercept
```

```
The model parameters are, slope (b1) = -9.1, and the intercept (b0) = 28.35
```

- The model is simply a line:

$$y = b_1 \cdot x + b_0$$

Response Feature

$$\phi = b_1 \cdot \rho + b_0$$

Predictor Feature

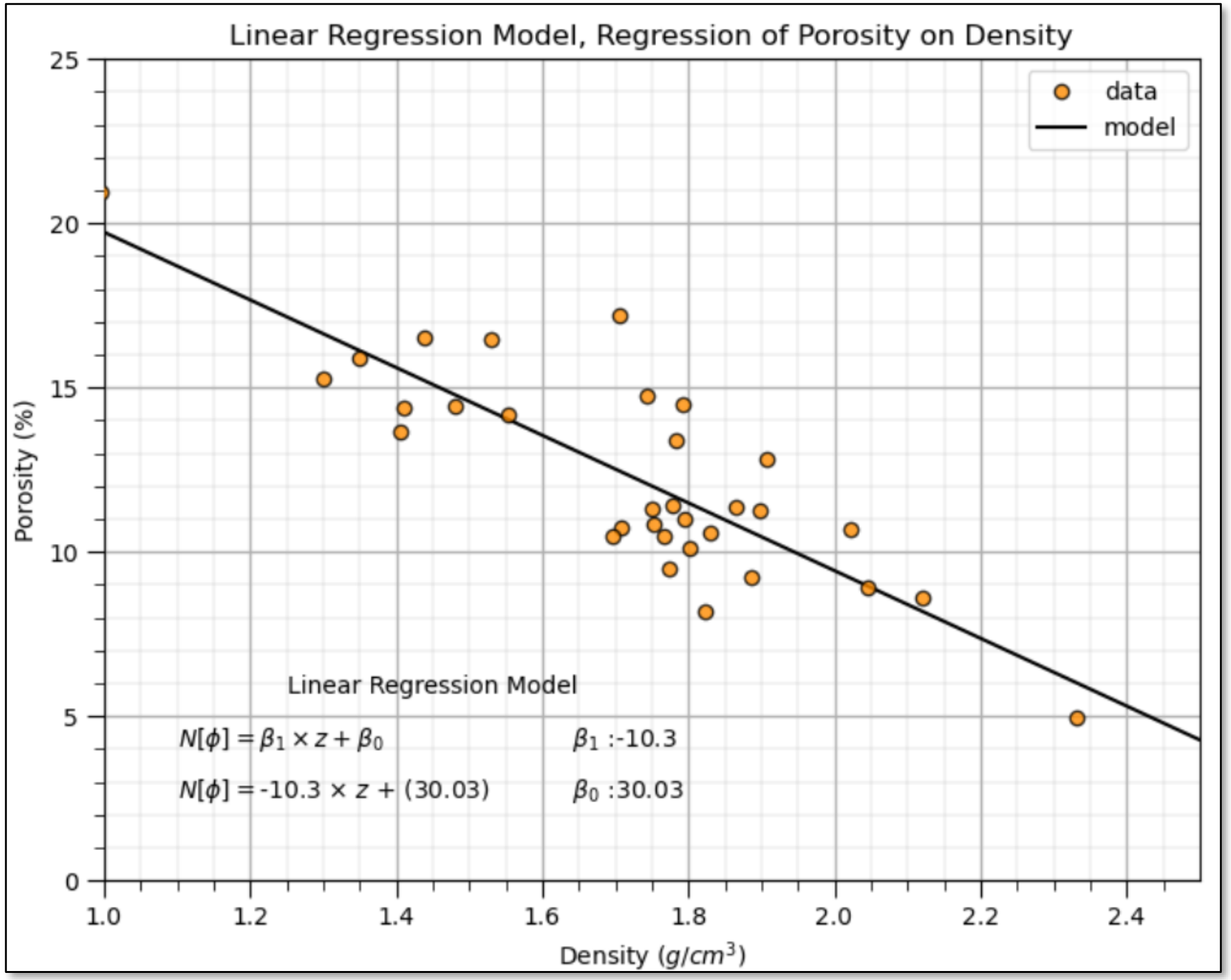
where ϕ is porosity and ρ is density

Our First Prediction Machine

Let's look at the model, our machine!

- If we change the data, the model will update. It learns from data!

- Nothing intimidating about linear regression!



Linear regression model to predict porosity from density, from Linear Regression chapter of Applied Machine Learning in Python e-book.

Linear Regression

Aspects of Linear Regression,

Linear regression is predictive machine learning, focus on making a prediction.

- linear regression is supervised learning, required labels,

$$Y = f(X_1, \dots, X_m) + \epsilon$$

- we are predicting a response, Y, from a set of features, $X_1, \dots, X_m$

We can predict continuous and categorical response features,

- continuous $Y \longrightarrow$ regression
- categorical $Y \longrightarrow$ classification, but logistic regression,

$$logit\big(P(Y=1)\big) = b_0 + b_1 X$$

For Cardinality = 2.

where $logit(p) = \frac{p}{1-p}$

We model log-odds of the outcome.

Why cover linear regression?

- Good to start with simple prediction methods to demonstrate the fundamentals.
- All the confidence intervals, prediction intervals, parameters tests are known!

"We can learn a lot of fundamentals from linear regression."

Linear Regression

Model Parameters Set to Minimize Mismatch with Training Data Locations

Objective, Find b_1 and b_0, fit a linear function, to: $y = b_0 + b_1 \cdot x$

- minimize Δy_i over all the data with $L^2 Norm$
- Δy_i is prediction error

Minimize: $\boldsymbol{\Delta y_i = y_i - y_{est}}$

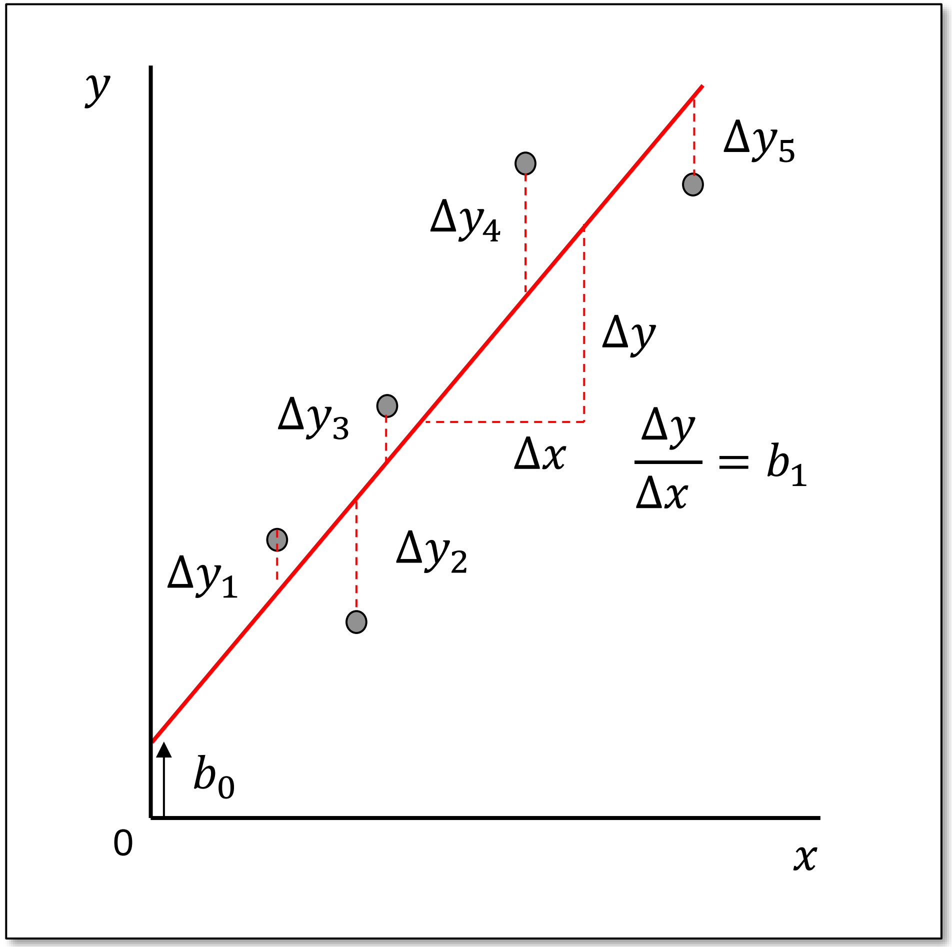
data model

Sum of Square Error

$$\sum_{i=1}^{n} (\Delta y_i)^2 = \sum_{i=1}^{n} (y_i - (b_0 + b_1 x_i))^2$$

Error Norm

This is our model's loss function to be minimized to train our model parameters with the training data.



Linear regression data, model and errors.

PGE 383 Subsurface Machine Learning

Lecture 9: Linear Regression

Lecture outline:

- L^1 and L^2 Norms

Norms

To train our models to training data, we require a single summary measure of mismatch with the training data, training error.

- the Error is observed at each training data location:

$$\Delta y_i = y_i - \hat{y}_i, \forall\ i = 1, \dots, n \text{ training data.}$$

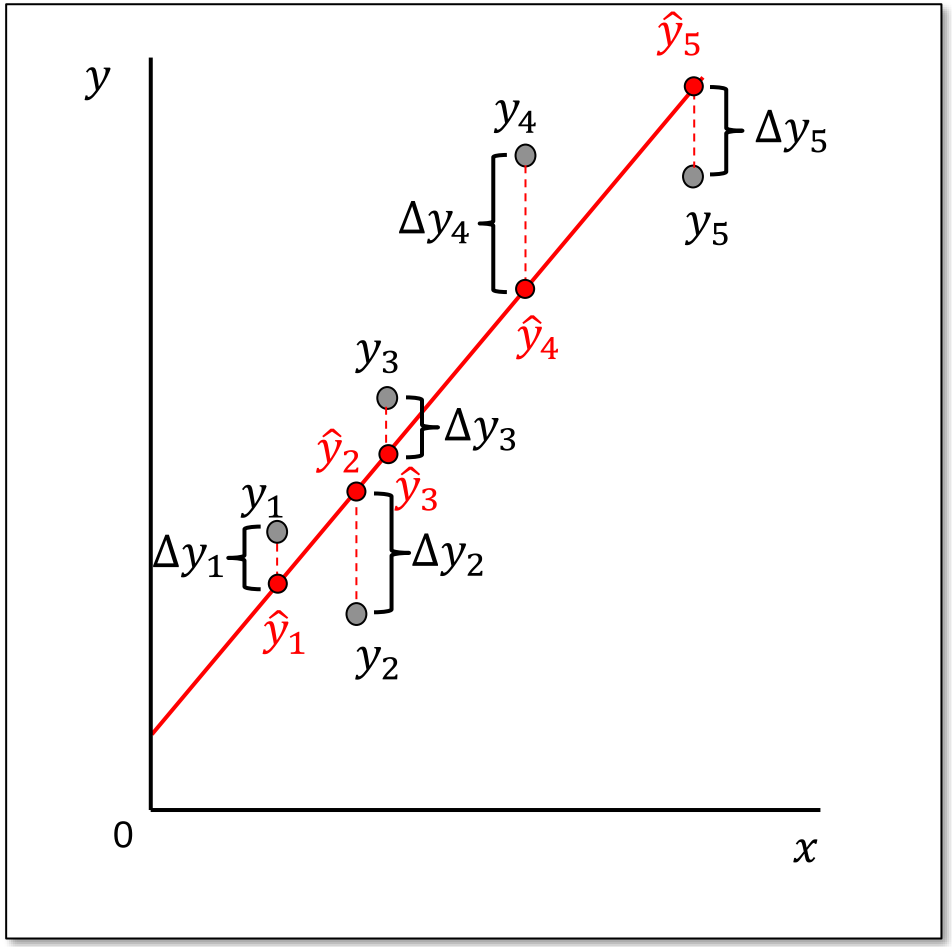
data   model estimate

- We need a summary over all training data, $i = 1, \dots, n$

  that we can minimize!

- We can't just sum the error over all n,
$$\sum_{i=1}^{n} \Delta y_i$$

Negative and positive error would cancel out!



Linear regression errors.

L^2 Norm

The L^2 Norm is Commonly Applied

Sum of square residuals (SSR) $\sum_{i=1}^{n}(\Delta y_i)^2$

Also expressed as,

- the Euclidean norm $\sqrt{\sum_{i=1}^{n}(\Delta y_i)^2}$ ← equivalent to distance if we had difference in x, y, z position.

- mean square error (MSE) $\frac{1}{n}\sum_{i=1}^{n}(\Delta y_i)^2$

- root mean square error (RMSE) $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\Delta y_i)^2}$

Minimization of the L^2 Norm for a linear model is known as the method of least squares.

L^1 Norm

The L^1 Norm is an Alternative Norm

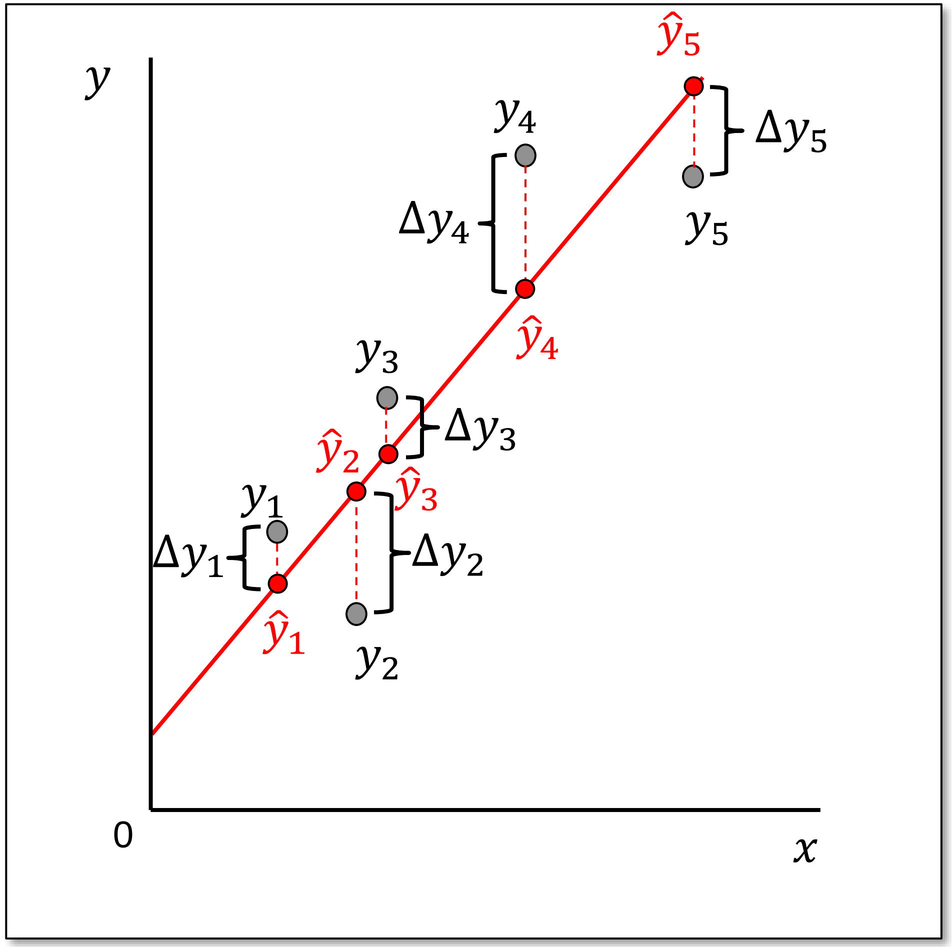
Manhattan norm, $$\sum_{i=1}^{n} |\Delta y_i|$$

Also expressed at the,

- Mean absolute error (MAE), $$\frac{1}{n}\sum_{i=1}^{n} |\Delta y_i|$$
- Sum of absolute residual (SAR), $$\sum_{i=1}^{n} |\Delta y_i|$$

Mahattan norm for the case of error.

Minimization with L^1 Norm is known as minimum absolute difference



Linear regression errors.

Norm Definition

Norm of a vector maps vector values to a summary measure $[\mathbf{0}, \infty)$, indicating size or length.

Currently our vector is error, residual at each training data, $\Delta y_i, \mathrm{i} = 1, \ldots, \mathrm{n}$.

- given a measure of distance, for example Euclidian distance

$$\|\Delta y\|_{\mathbf{2}} = \sqrt{\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} \Delta {y_i}^{\mathbf{2}}}$$

- we can generalize as a p-norm

$$\|\Delta y\|_{\boldsymbol{p}} = \left(\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} |\Delta y_{\boldsymbol{\alpha}}|^{\boldsymbol{p}} \right)^{\mathbf{1}/\boldsymbol{p}}$$

$$\begin{bmatrix} v_i \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \longrightarrow \textbf{norm}$$

$$\begin{bmatrix} \Delta y_i \\ \Delta y_2 \\ \vdots \\ \Delta y_n \end{bmatrix} \longrightarrow \textbf{error norm}$$

Norm (above) and error norm (below).

Later we will additional norms (beyond an error summary) to our loss function, known as regularization.

L^1 and L^2 Norms

Least Absolute Deviations (L1)	Least Squares (L2)
Robust	Not very robust
Unstable solution	Stable solution
Possibly multiple solutions	Always one solution
Feature selection built-in	No feature selection
Sparse outputs	Non-sparse outputs
No analytical solutions	Analytical solutions

Summary table comparing L^1 **and** L^2 **Norms** from
http://www.chioka.in/differences-between-the-l1-norm-and-the-l2-norm-least-absolute-deviations-and-least-squares/

Robust: resistant to outliers.

Unstable: for small changes in training the trained model predictions may ‘jump’

Multiple Solutions: multiple paths same lengths in the city!

Sparse Output: model coefficients tend to 0.0.

Experiential Learning with Norms

Interactive Code in a Python Jupyter Notebook

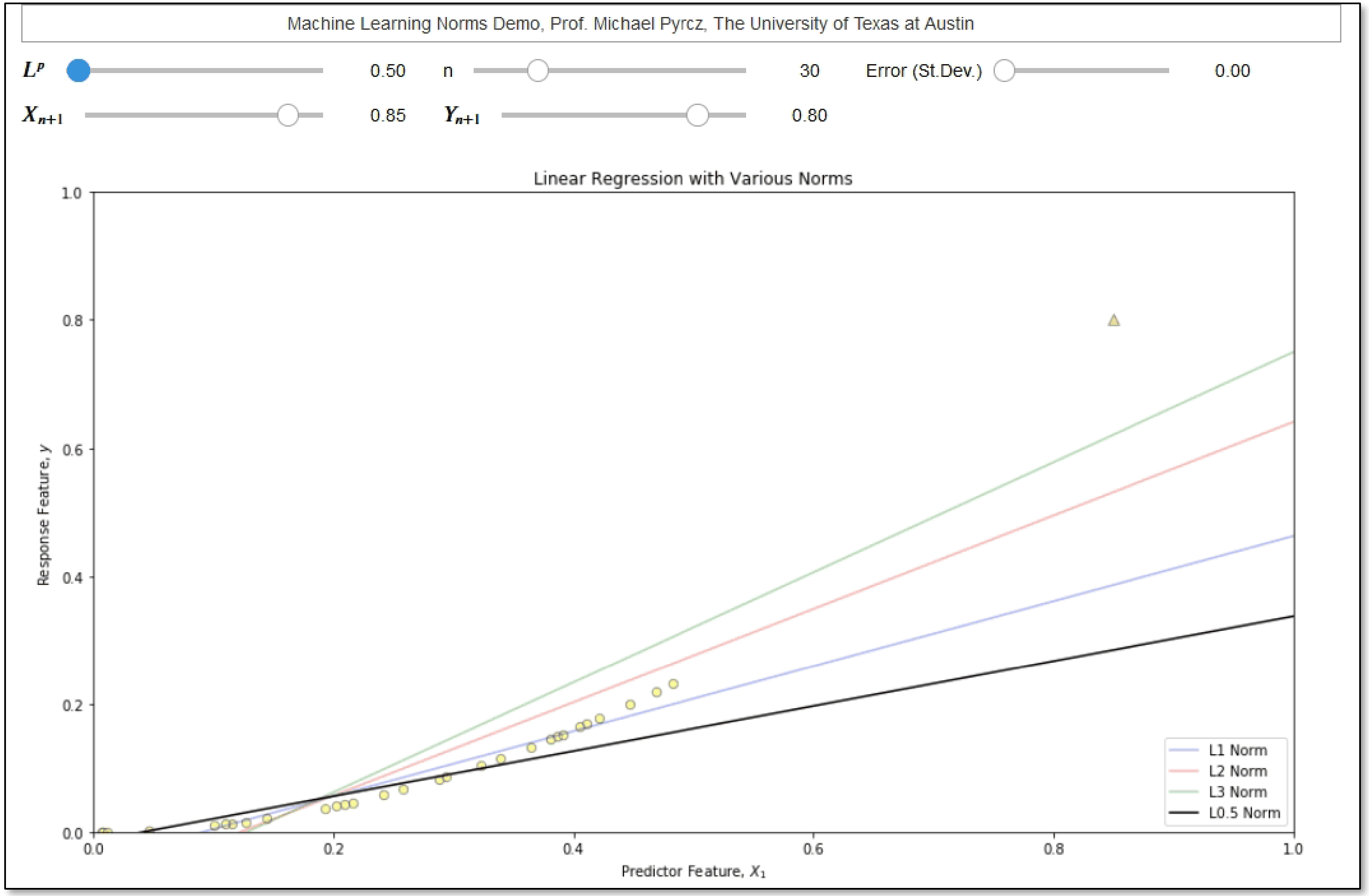
I coded linear regression from scratch with the general p-norm,

$$\|\Delta y\|_p = \left(\sum_{i=1}^{n} |\Delta y_\alpha|^p \right)^{1/p}$$

- Modify the number of data, error in the data and an outlier

Change p and observe the impact on the linear regression model, recall,

- $p = 1.0$, is L^1
- $p = 2.0$, is L^2



Interactive code to observe the impact of choice of norm, Interactive_Norms.ipynb.

PGE 383 Subsurface Machine Learning

Lecture 9: Linear Regression

Lecture outline:

* Linear Regression Training

Linear Regression
Analytical Form Derivation

Derivation of linear regression, solving for b_1:

$$SSE = \sum_{i=1}^{n} (y_i - \overbrace{(b_0 + b_1 x_i)}^{\hat{y}_i})^2$$

$$\frac{\partial}{\partial b_1}\left[\sum_{i=1}^{n}(y_i - b_0 - b_1 x_i)^2\right] = \sum_{i=1}^{n} -2x_i(y_i - b_0 - b_1 x_i)$$

Minimize by setting derivative equal to 0.0.

Recall:

$$\frac{d}{db_1}u^2 = 2u \cdot \frac{du}{db_1}$$

$$0 = \sum_{i=1}^{n} -2x_i(y_i - b_0 - b_1 x_i)$$

$$0 = \sum_{i=1}^{n} (x_i y_i - b_0 x_i - b_1 {x_i}^2)$$

substitute $b_0 = \bar{y} - b_1 \bar{x}$

$$0 = \sum_{i=1}^{n} (x_i y_i - (\bar{y} - b_1 \bar{x}) x_i - b_1 {x_i}^2) = \sum_{i=1}^{n} (x_i y_i - \bar{y} x_i + b_1 \bar{x} x_i - b_1 {x_i}^2)$$

Linear Regression Analytical Form Derivation

Derivation of linear regression, solving for $\boldsymbol{b_1}$:

$$0 = \sum_{i=1}^{n}(x_i y_i - \bar{y}x_i + b_1\bar{x}x_i - b_1 {x_i}^2) = \sum_{i=1}^{n}(x_i y_i - \bar{y}x_i) + \sum_{i=1}^{n}(b_1\bar{x}x_i - b_1 {x_i}^2)$$

extract $-b_1$

$$0 = \sum_{i=1}^{n}(x_i y_i - \bar{y}x_i) - b_1\sum_{i=1}^{n}({x_i}^2 - \bar{x}x_i)$$

Common, Equivallent Form

Covariance of X and Y

$$\boldsymbol{b_1} = \frac{\sum_{i=1}^{n}(x_i y_i - \bar{y}x_i)}{\sum_{i=1}^{n}({x_i}^2 - \bar{x}x_i)} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sigma_{XY}}{\sigma_X^2}$$

Variance of X

Linear Regression Analytical Form Derivation

Derivation of linear regression, solving for b_0:

$$SSE = \sum_{i=1}^{n} (y_i - (b_0 + b_1 x_i))^2$$

$$\frac{\partial}{\partial b_0}\left[\sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2\right] = \sum_{i=1}^{n} -2(y_i - b_0 - b_1 x_i)$$

To minimize MSE

$$0 = \sum_{i=1}^{n} -2(y_i - b_0 - b_1 x_i)$$

$$0 = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i) = \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} b_0 - \sum_{i=1}^{n} b_1 x_i = \sum_{i=1}^{n} y_i - \mathrm{n} b_0 - \sum_{i=1}^{n} b_1 x_i$$

$$b_0 = \frac{\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} b_1 x_i}{n} = \frac{\sum_{i=1}^{n} y_i}{n} - b_1 \frac{\sum_{i=1}^{n} x_i}{n} = \bar{y} - b_1 \bar{x}$$

Solve for the intercept given the average of the predictor and the response features.

PGE 383 Subsurface Machine Learning

Lecture 9: Linear Regression

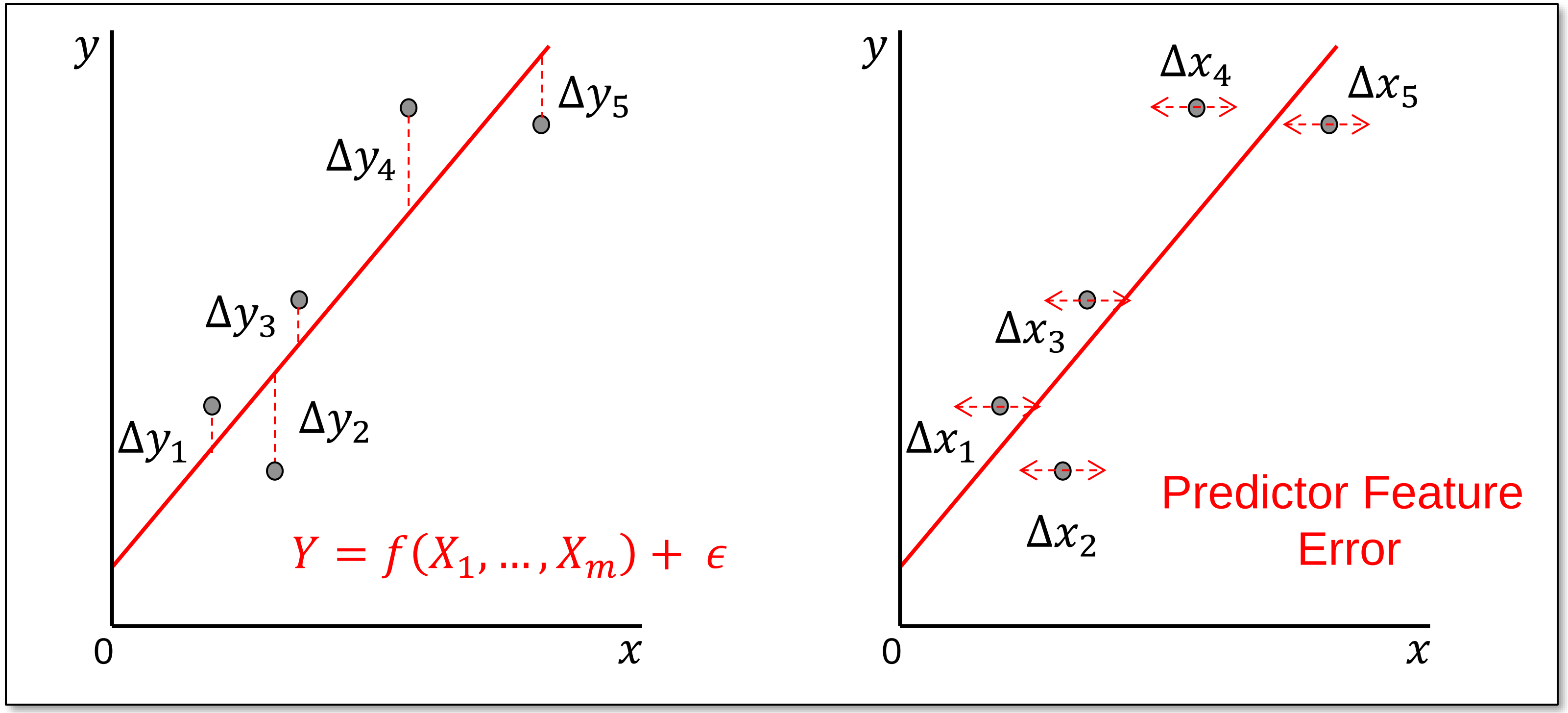
Lecture outline:

- **Linear Regression Assumptions and Limitations**

Linear Regression

The Model Includes Important Assumptions About The Data and the Model

- **Error-free:** predictor variables are error free, not random variables, $Y = f(X_1, \ldots, X_m) + \epsilon$ ← Error only in $\hat{y}$



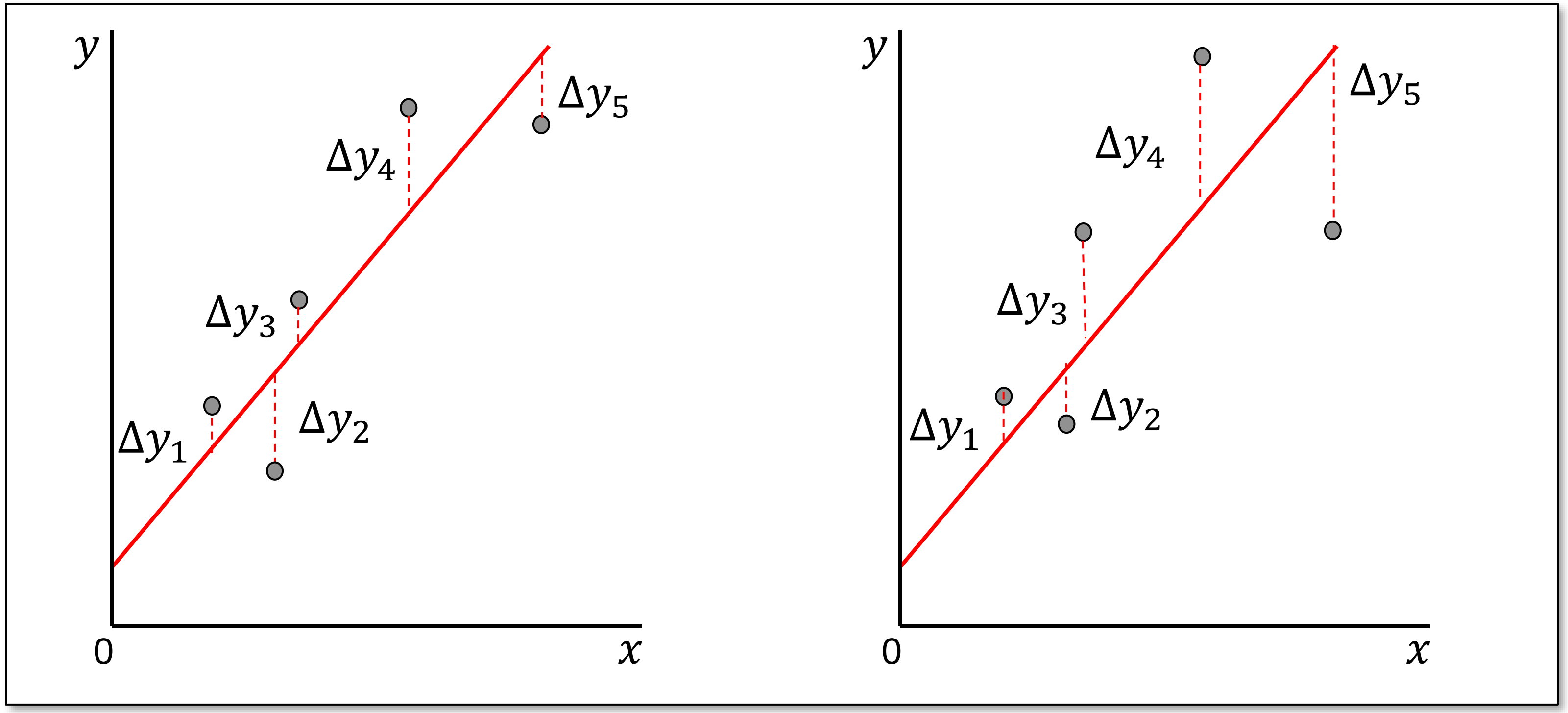
Error in prediction of y, $\hat{y}$ (left), error in predictor features, Δx_i (right).

Linear Regression

The Model Includes Important Assumptions About The Data and the Model

- **Constant Variance:** error in response is constant over predictor features(s) values

$$Y = f(X_1, \dots, X_m) + \epsilon \quad \longleftarrow \quad \textbf{Error is not } f(X_1, \dots, X_m)$$



Homoscedastic, constant error (left) and heteroscedastic, error related to x (right).

Linear Regression

The Model Includes Important Assumptions About The Data and the Model

- **Linearity:** response feature is linear combination of predictor feature(s)

- **Independence of Error:** error in response features are uncorrelated with each other

- **No multicollinearity:** none of the features are redundant with other features

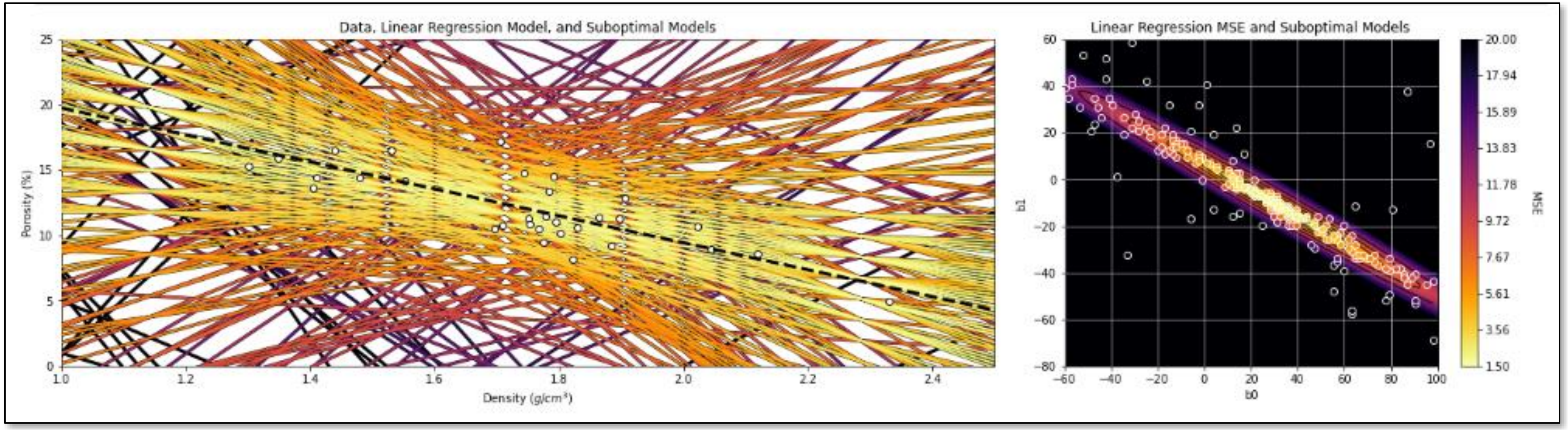
Linear Regression

What does the linear regression loss function look like?

- For a data set I calculated the L^2 loss function over a combinatorial of b_1 and b_1 values.

- Then I randomly sampled and plotted many models inverse weighted by the loss (more likely to take godo models).

- I plotted the models lines colored by MSE loss and better models on top.



Visualization of a linear regression L2 loss function.

Lecture 9: Linear Regression

Lecture outline:

- **Linear Regression Diagnostics**

Linear Regression

The Model Can Be Tested for Significance and the Proportion of Variance Explained.

- r^2, coefficient of determination is the proportion of variance explained by the model in linear regression

Variance explained by the model

Variance of model estimates, $\sigma_{\hat{y}}^2$

$$SS_{reg} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

Total variance in the dataset

Variance over the data, σ_y^2

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Note, the degrees of freedom cancel out, $\frac{SS_{reg}}{SS_{tot}}$ is same as $\frac{\sigma_{reg}^2}{\sigma_{tot}^2}$

$$r^2 = \frac{SS_{reg}}{SS_{tot}} = \frac{explained\ variation}{total\ variation}$$

f-test for significance of any linear predictor to the response.

- also note for linear regression, $r^2 = (\rho)^2$, we can relate r^2 to the Pearson's correlation coefficient, ρ.

r^2 Coefficient of Determination

The Model Can Be Tested for Significance and the Proportion of Variance Explained.

- r^2, coefficient of determination is the proportion of variance explained by the model in linear regression

- This works only for linear models, where:

Variance explained

Variance of residual, unexplained

$$\sigma^2_{tot} = \sigma^2_{reg} + \sigma^2_{res}$$

Variance total

$$\sigma^2_{res} = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

$$\sigma^2_{reg} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

$$\sigma^2_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

- For nonlinear models this will not likely hold, $\sigma^2_{tot} \neq \sigma^2_{reg} + \sigma^2_{res}$

- Then $\frac{\sigma^2_{reg}}{\sigma^2_{tot}}$ may exceed $[0,1]$, render values < 0 and > 1.

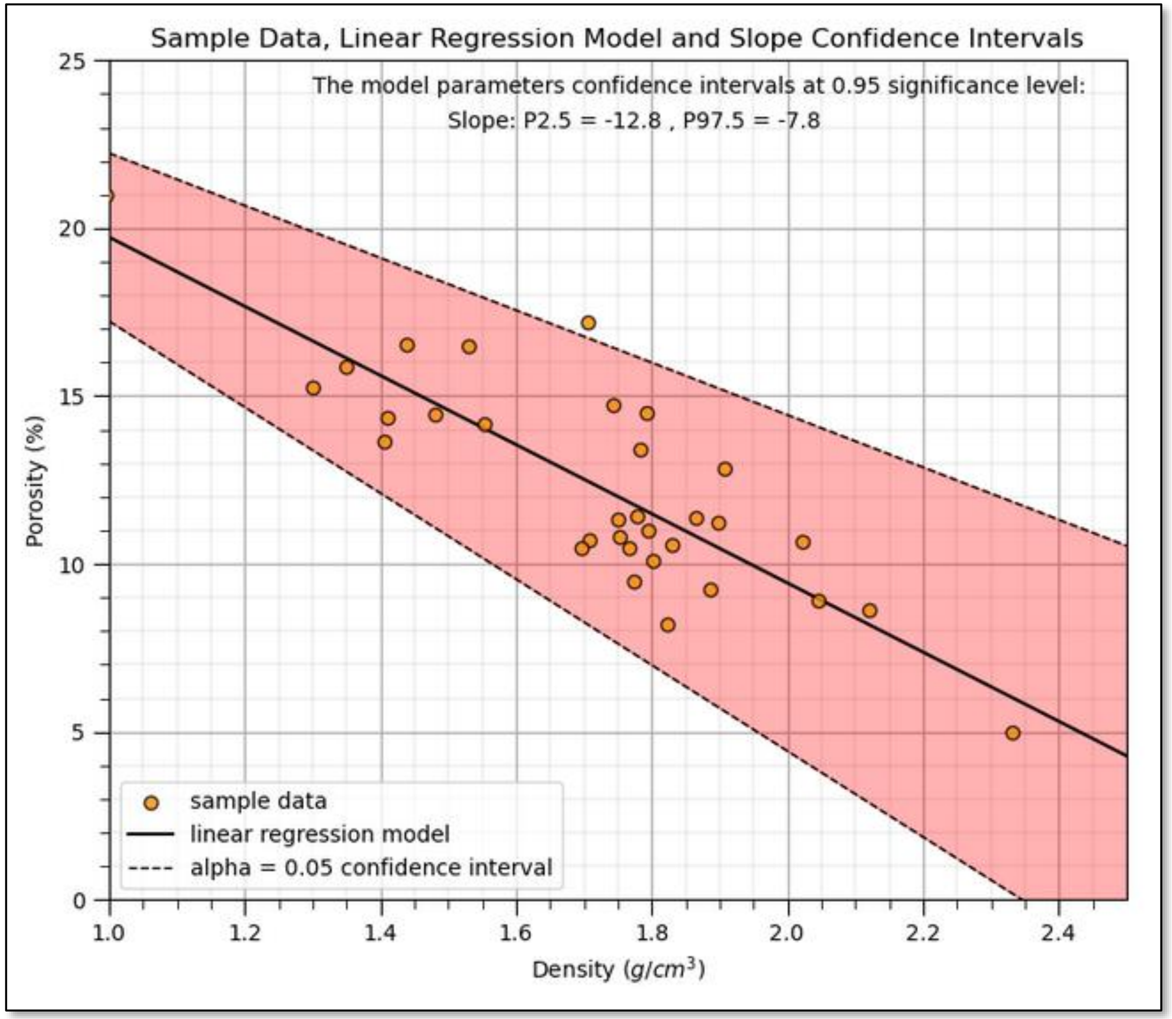
- For most of our models we will use more robust measures, e.g. mean square error (MSE)

Confidence Interval Definition

The **uncertainty in a summary statistic / model parameter** represented as a range, lower and upper bound, based on a specified probability interval known as the **confidence level**.

We communicate confidence intervals like this:

- There is a 95% probability (or 19 times out of 20) that model slope is between 0.5 and 0.7.
- We cover analytical methods here, but we could also use the more flexible bootstrap!
- In Bayesian methods we refer **credibility intervals**, more on the difference with CI later.



Linear regression model slope confidence interval, from the Linear Regression chapter of Applied Macchine Learning in Python e-book.

Linear Regression Confidence Interval

We Can Calculate the Uncertainty in the Model

Confidence interval for model parameters given the available training data

Standard error in the slope

$$\widehat{b_1} \pm t_{(\alpha/2, n-2)} \times SE_{b_1} \qquad \widehat{b_1} \pm t_{\alpha/2,n-2} \times \boxed{\left(\frac{\sqrt{n}\hat{\sigma}}{\sqrt{n-2}\sqrt{\sum(x_i-\bar{x})^2}}\right)}$$

se1 in Excel

$$\widehat{b_0} \pm t_{(\alpha/2, n-2)} \times SE_{b_0} \qquad \widehat{b_0} \pm t_{\alpha/2,n-2} \times \boxed{\left(\sqrt{\frac{\hat{\sigma}^2}{n-2}}\right)}$$

seb in Excel

Standard error in the intercept

- **Standard Error**: measure of accuracy of an estimate, equal to the standard deviation of a theoretical distribution of a large set of such estimates.

Prediction Interval Definition

The **uncertainty in the next prediction** represented as a range, lower and upper bound, based on a specified probability interval known as the **confidence level**.

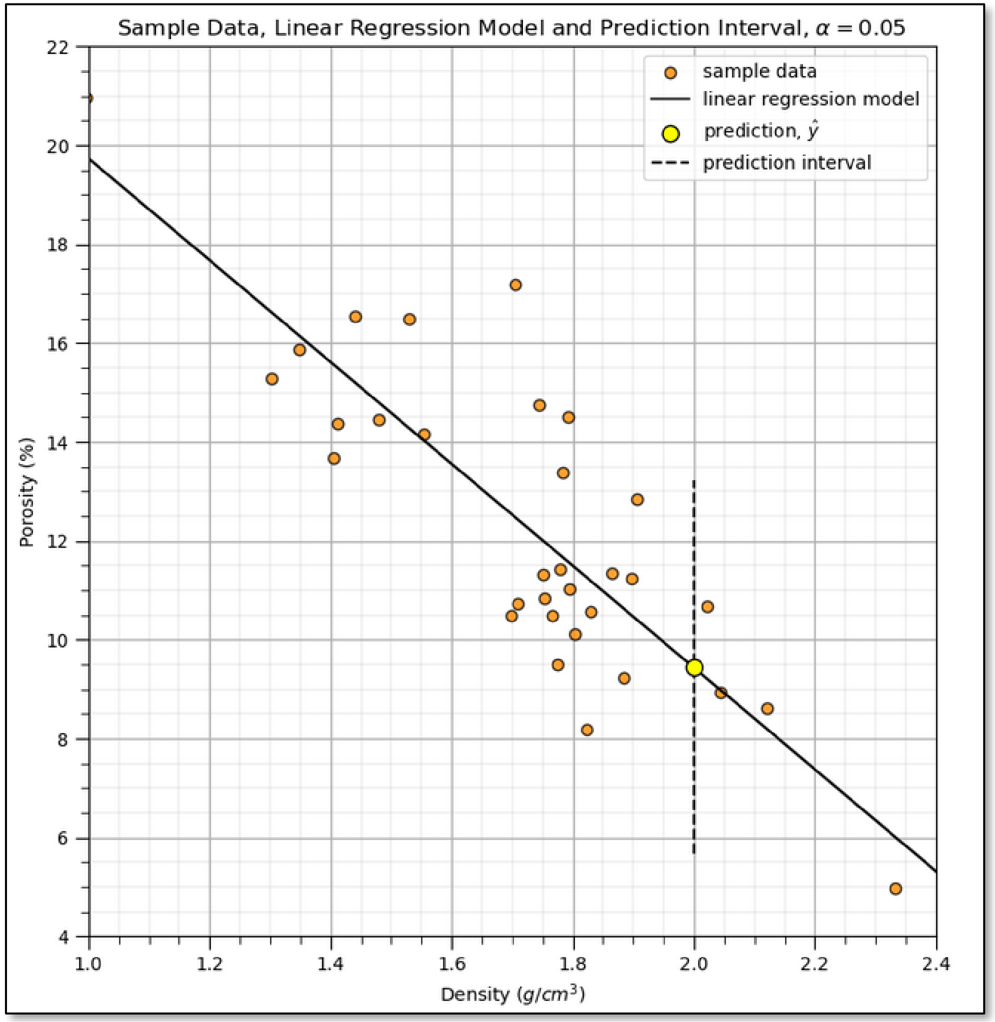
We communicate confidence intervals like this:

- There is a 95% probability (or 19 times out of 20) that the true reservoir NTG is between 13% and 17%, given, $X_1 = x_1, \ldots, X_m = x_m$.

- We are calculating the uncertainty in our prediction (not just the model)

For prediction intervals we integrate:

1. Uncertainty in the model $E\{\hat{Y} | X = x\}$
2. Error in the model, conditional distribution $\hat{Y} | X = x$, (error)



Linear regression model prediction interval, from the Linear Regression chapter of Applied Macchine Learning in Python e-book.

Prediction Interval

Provides an Uncertainty Model for the Predictions

Recall prediction interval are concerned with uncertainty in the next observation

– We answer the question, given I know the porosity, x_{n+1}, what is the interval (e.g.) with 95% probability containing the true value permeability, y_{n+1}? ← next sample

$$\underbrace{\hat{y}_{n+1}}_{\text{model estimate}} \pm \underbrace{t_{\alpha/2,n-2}}_{\text{t-statistic}} \underbrace{\boxed{\sqrt{MSE}\sqrt{1+\frac{1}{n}+\frac{(x_{n+1}-\bar{x})^2}{\sum(x_i-\bar{x})^2}}}}_{\text{standard error of our model estimate}}$$

model estimate

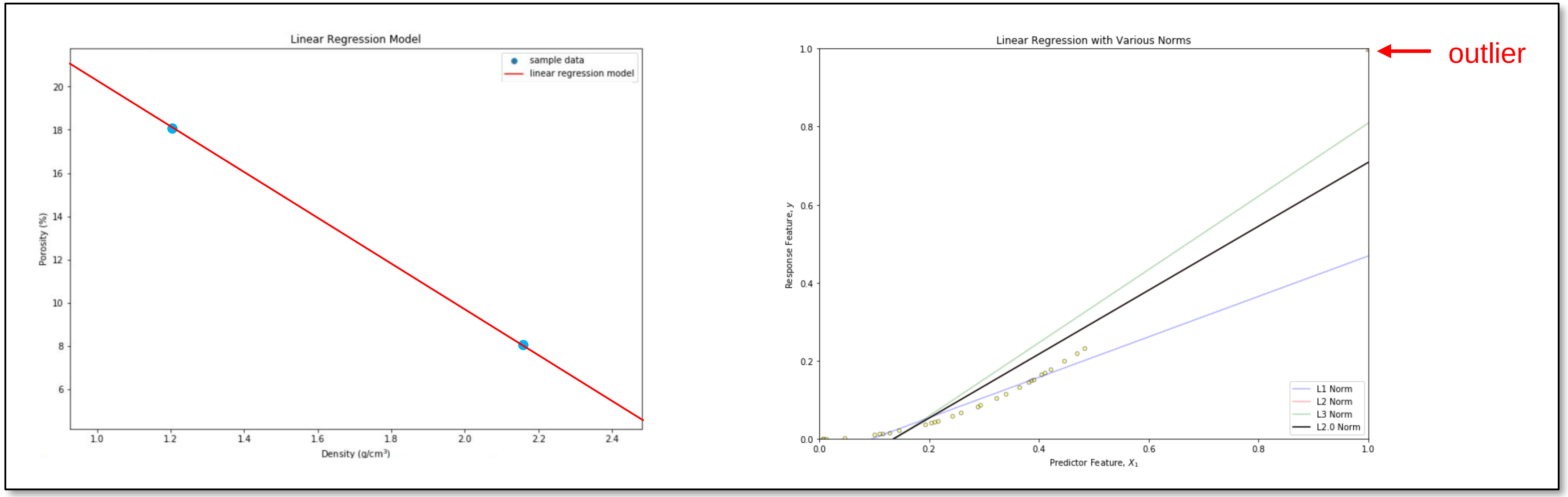
t-statistic

$$MSE = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n-2} = \sum_{i=1}^{n} \frac{(y_i - (b_0 - b_1 x))^2}{n-2}$$

standard error of our model estimate

Our First Machine

Would these be a fair models?



- Does the data support this model? We are **overfitting** the data!
- Is it safe to **extrapolate** with this model away from the data?
- How is our model handling outliers?

Computational Complexity

Time complexity refers to computational time and the scaling of this time to the size of the problem for a given algorithm

Space complexity refers to computer memory required and the scaling of storage to the size of the problem for a given algorithm.

We default to **worst-case complexity**, the worst case for complexity given a specific problem size, where n is large.

- Assumes all steps are required, e.g., data is not presorted etc.
- We will default to time complexity and express it as:

$O(f(n))$, where n represents size

- Within the parenthesis we will include some measures of size of the problem and they way they impact run times.
- We will assume large system, $n \to \infty$, known as asymptotic complexity

Computational Complexity

Time Complexity Examples

Quadratic Time, $\boldsymbol{O(n^2)}$

- Where there is a constant (time per operation), c_u, to provide an upper bound, $c_u n^2$
- e.g., Integer multiplication, bubble sort

Linear time, $\boldsymbol{O(n)}$

- e.g., finding the min or max in an unsorted array

Fractional Power, $\boldsymbol{O(n^c)}$, 0 < c < 1

- e.g., searching in a kd-tree, $O\left(n^{1/2}\right)$

Exponential Time, $\boldsymbol{O(2^n)}$

- e.g., travelling salesman problem with dynamic programing

Computational Complexity of Linear Regression

The computational cost to solve multilinear regression?

Note, we only derived the linear regression analytical solution, but here is it in matrix form for multilinear regression.

$$\beta = (X^T X)^{-1} X^T y$$

Where $X_{n \times m}$ is the predictor feature training data matrix and $y_{n \times 1}$is the vector with the response feature for the training data, β are the model parameters.

For linear regression $O(m^2 n)$ where m is the number of features and n is the number of samples.

- $O(m^2 n)$ to multiply X^T by X
- $O(mn)$ to multiply X^T by Y
- $O(m^3)$ to invert $X^T X$

Given we expect $n >> m$, $O(m^2 n)$ term dominates, complexity is stated as $O(m^2 n)$.

Our First Machine

What did we learn from our simple predictive machine learning model?

1. Flexible to fit the data, learns from the data
2. Minimize error with the training data
3. Important assumptions about the data and model
4. Model can be tested for significance and the proportion of variance explained
5. Includes uncertainty in the model
6. Predict based on new data with uncertainty
7. Issues with overfit and extrapolation

Think of predictive machine learning as advanced linear regression, line fitting to data!

PGE 383 Subsurface Machine Learning

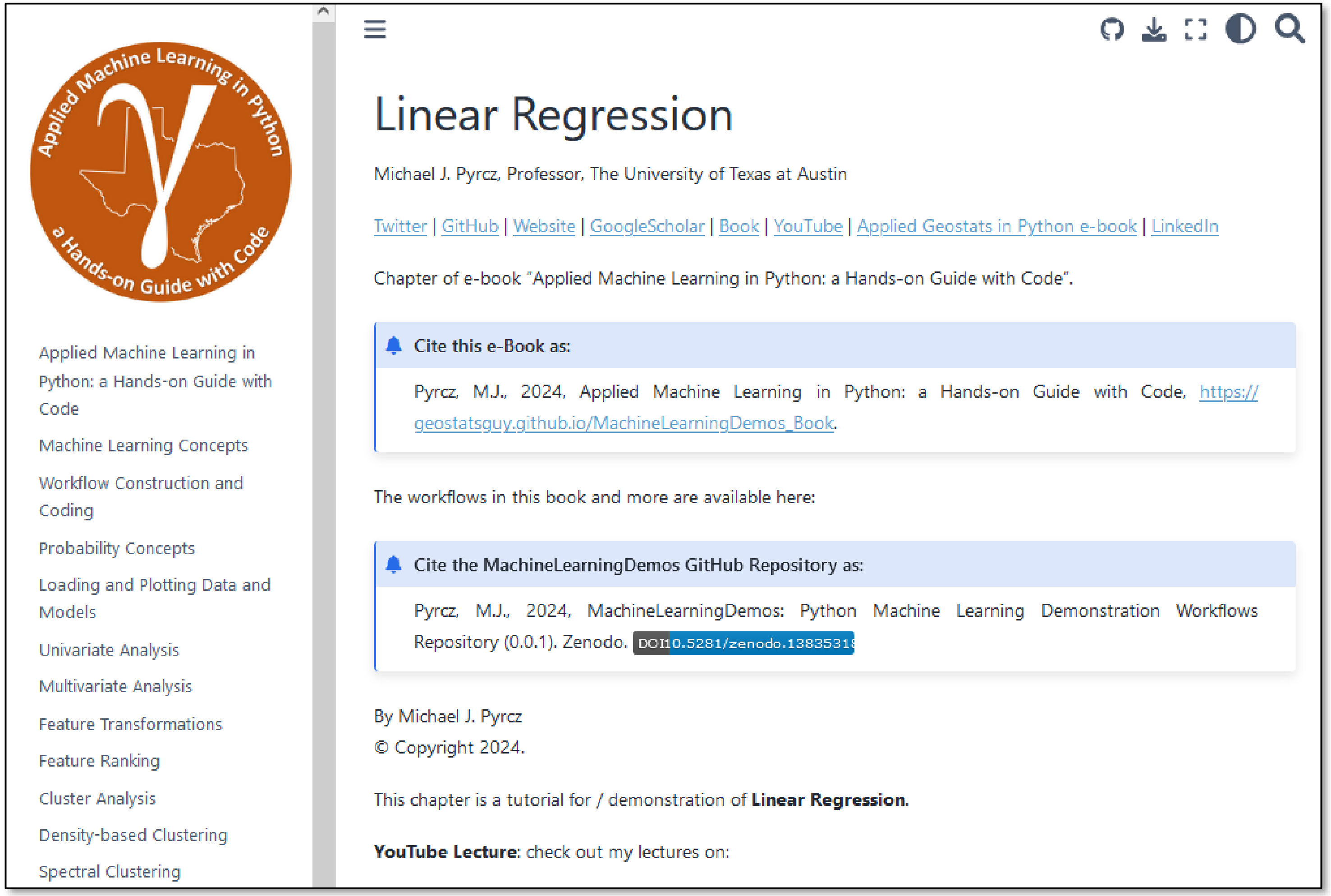
Lecture 9: Linear Regression

Lecture outline:

- **Linear Regression Hands-on**

Linear Regression Demonstration in Python

Demonstration of linear regression with a well-documented workflow.

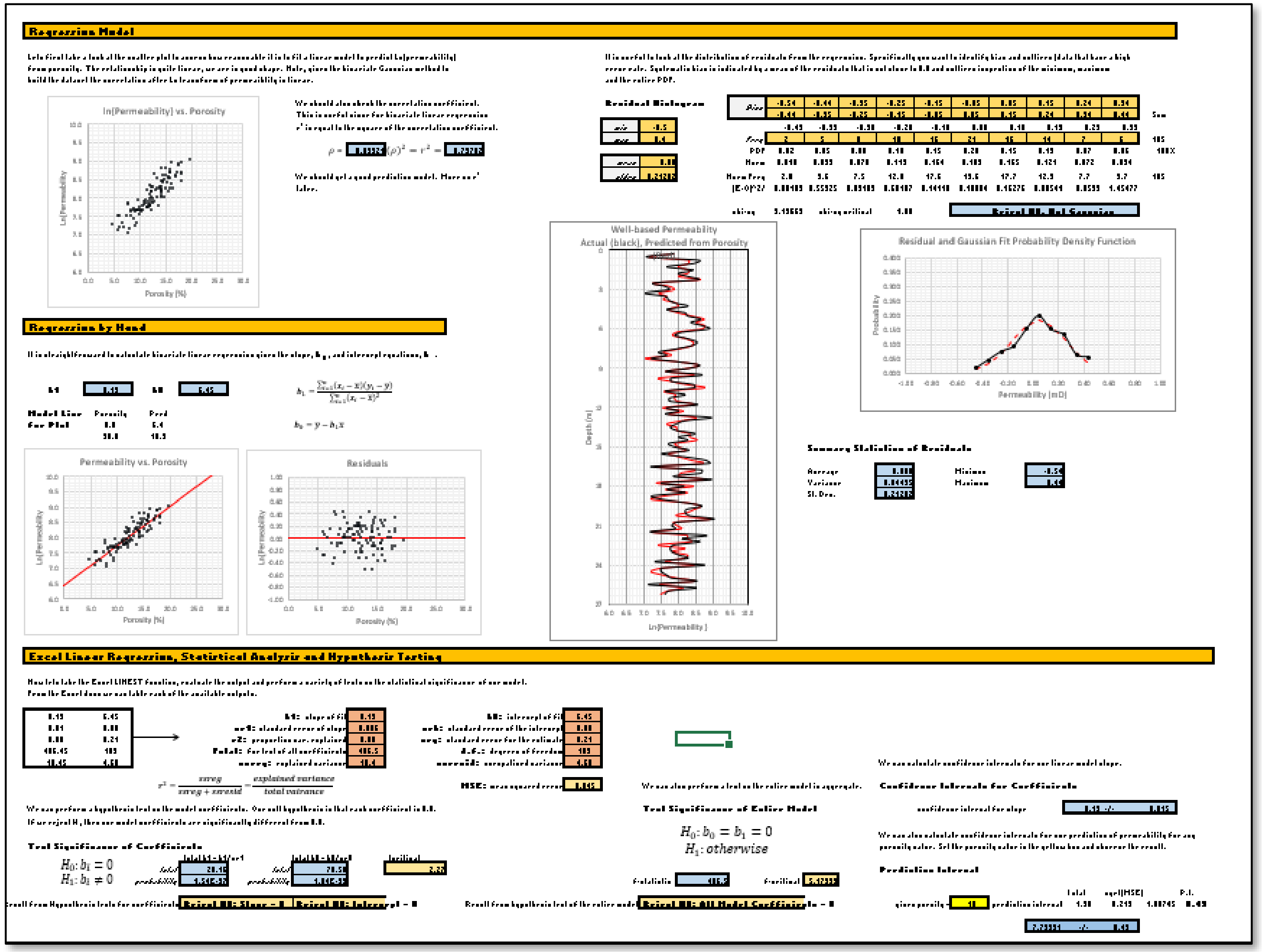


Linear regression chapter of Applied Machine Learning in Python e-book.

Linear Regression Demonstration

Demonstration workflow in for linear regression in Excel.

- Training
- Confidence Intervals
- Prediction Intervals
- Hypothesis Testing
- Model Checking
- Etc.



Linear regression in Excel, file is Linear_Regression_Demo_v2.xls.

PGE 383 Subsurface Machine Learning

Lecture 9: Linear Regression

Lecture outline:

- **Linear Regression**
- L^1 **and** L^2 **Norms**
- **Linear Regression Training**
- **Linear Regression Assumptions and Limitations**
- **Linear Regression Diagnostics**
- **Linear Regression Hands-on**