Using Data Analytics for Gas Turbines: Basics, Potential Pitfalls, and Best Practices

PGU 306 – Christopher Perullo
Senior Research Engineer
Aerospace Engineering
Georgia Institute of Technology

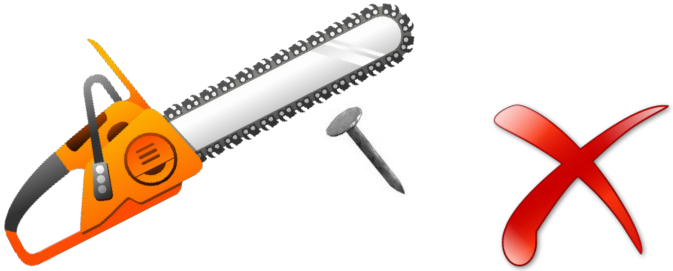
December 3, 2018

Outline

- Some Basic Definitions
- Types of Data Analytic / AI / Machine Learning Models
  - Artificial Neural Networks
  - Clustering Algorithms (Many APR packages)
  - Classification Algorithms
  - Bayesian Learning
- Model Selection – What's appropriate for my problem?
- General Model Creation Process (With Examples)
  - Real or Simulated Data?
  - Identifying a good training data set
  - Evaluating model quality & model validation
- Case Studies
  - Gas turbine performance data – Neural Networks
  - Gas turbine performance data – Clustering
  - Identifying discrete operating modes – Neural Networks

Goals of Course

- You should get two things from this course:
- When to use the right tool (model)…



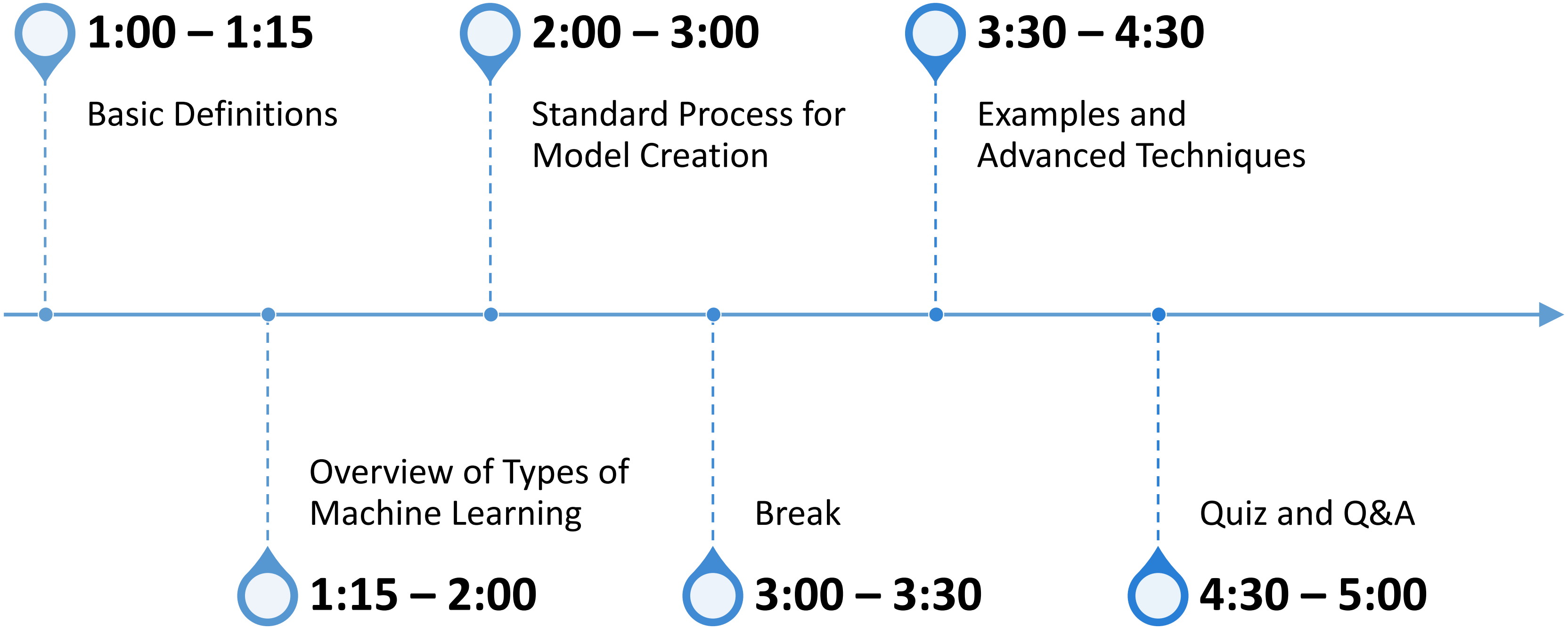
- Basic tools and techniques to evaluate if your results are any good…

What Will Be Avoided Today

- Lots of math or derivations
- Extensive deep-dive into every variant of machine learning
- We will discuss
  - High level categories of models
  - Techniques appropriate to all classes of modeling
- Still important to understand nuances of chosen method
- Unnecessary jargon

A Brief Survey

1. Who here has used PRiSM / Smart Signal / PredictIt! Etc.?
2. Does your company have a structured process in place for model building and validation?
3. How good are your models?
4. Why did you pick the modeling approach you chose?
5. Bonus: What is the modeling approach you chose?



1:00 – 1:15
Basic Definitions

1:15 – 2:00
Overview of Types of Machine Learning

2:00 – 3:00
Standard Process for Model Creation

3:00 – 3:30
Break

3:30 – 4:30
Examples and Advanced Techniques

4:30 – 5:00
Quiz and Q&A

Course Outline

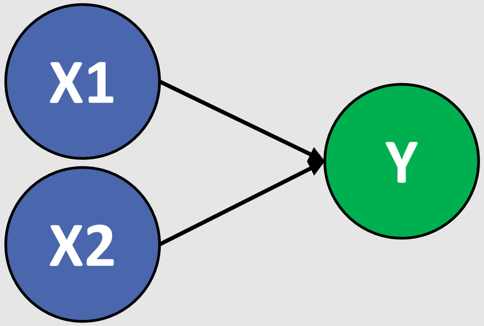
Basic Definitions

Basic Definitions Datasets

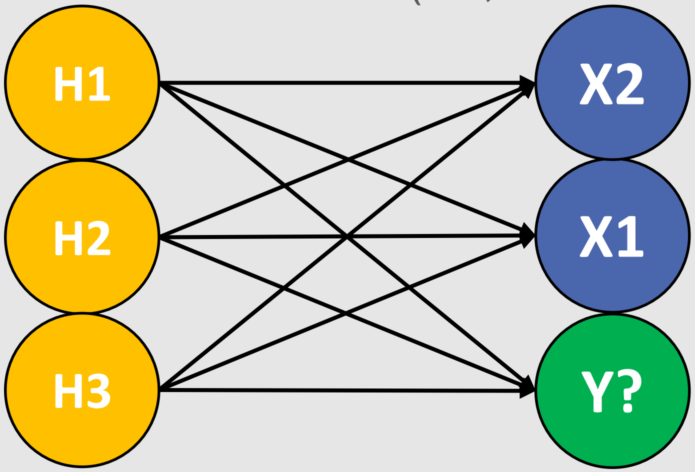
- **Training Data:**
  - Data used to train the model
  - Should be free of errors
  - Should be differentiated into discrete operating modes
  - Should cover entire operational space (i.e., ambient temperature)
- **Verification Data:**
  - Data used to validate model’s goodness of fit
  - Should be selected from same data set as training data
  - Used to identify if model is over-fit
- **Validation Data:**
  - Data used to validate model’s predictive capability
  - Should be selected from a data set outside of the training region
  - Validates whether important factors are missing or if training set was not extensive enough

Basic Definitions Learning

- Supervised Learning
  - Dataset has known inputs and outputs
  - Dataset can be categorized a priori
  - Most engineering problems fall into this category



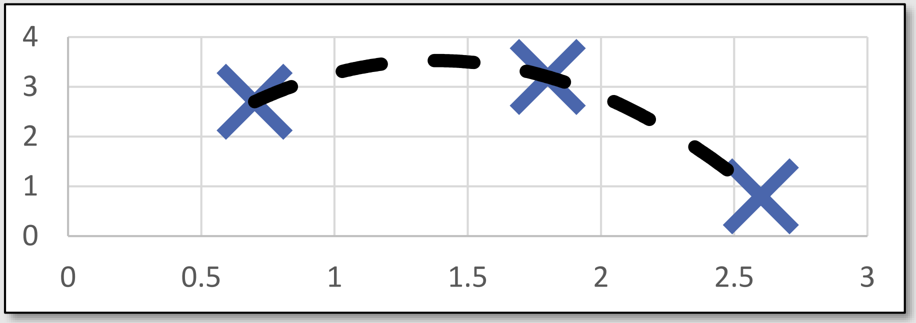
- Unsupervised Learning
  - Inputs and outputs are abstract or cannot be defined
  - Appropriate when relationship between variables is unknown (i.e., social behavior)



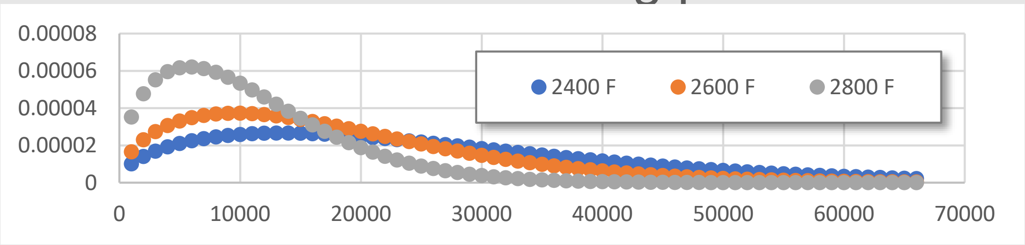
Basic Definitions

Randomness

- **Deterministic Model**
  - Unique inputs provide unique output
  - Most common type of model
  - More sensitive to data quality



- **Stochastic or Probabilistic Model**
  - Inputs and/or outputs represented by probability distribution
  - More common in lifing problems



Overview of Common Types of Analytics and Machine Learning

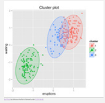
A Note on AI / Machine Learning

- The topics we will be discussing today apply to any type of model regression
- You want to create a ‘black box’ between inputs and outputs
- Can be based on:
  - Measurement data (plant data)
  - Computer generated data (FEM model)
  - Combination
- Conceptually useful to think of machine learning as high fidelity curve fit
- TRUE INTELLIGENCE REQUIRES INSIGHT

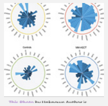
Common Types of Machine Learning

known Author is licensed under CC BY

Artificial Neural Networks

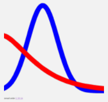


Clustering Algorithms (APR often falls into this category)



This Photo by Unknown Author is

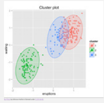
Advanced Pattern Recognition



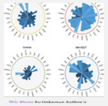
Bayesian Learning

Common Types of Machine Learning

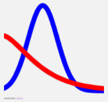
Artificial Neural Networks



Clustering Algorithms (APR often falls into this category)



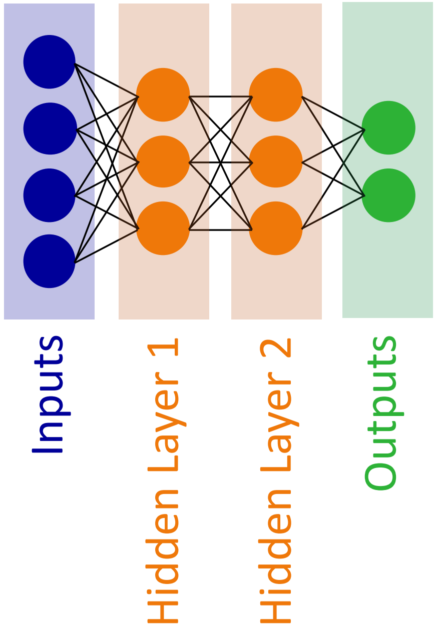
Advanced Pattern Recognition



Bayesian Learning

Artificial Neural Networks – What are they?

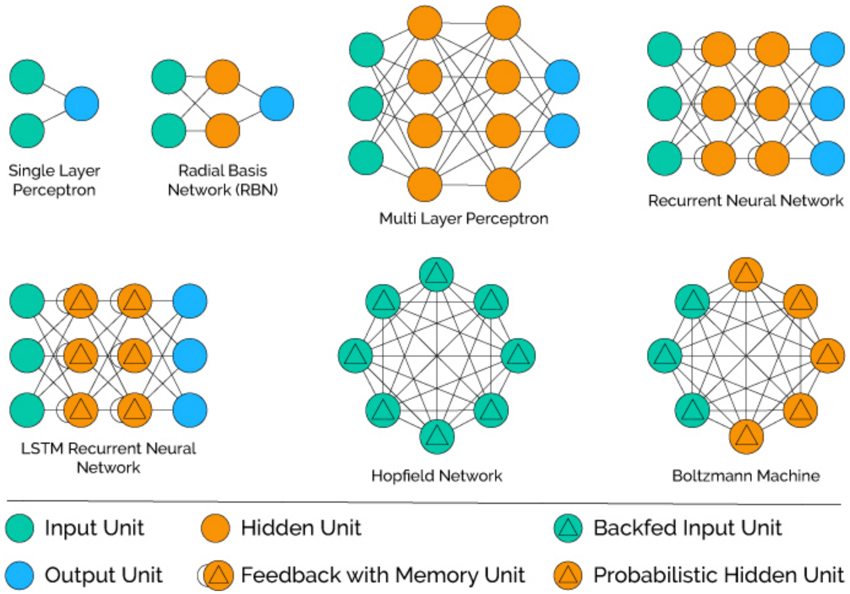
- Designed to mimic the connection of neurons in the human brain
- Nominally consists of 3-4 layers (multi layer perceptron)
  - Input layer
  - One to two neuron layers (hidden nodes)
  - Output layer
- Both deterministic and probabilistic types exist
- Static and 'learning' or updating models exist
- Today we will be discussing deterministic / static models



Artificial Neural Network - Uses

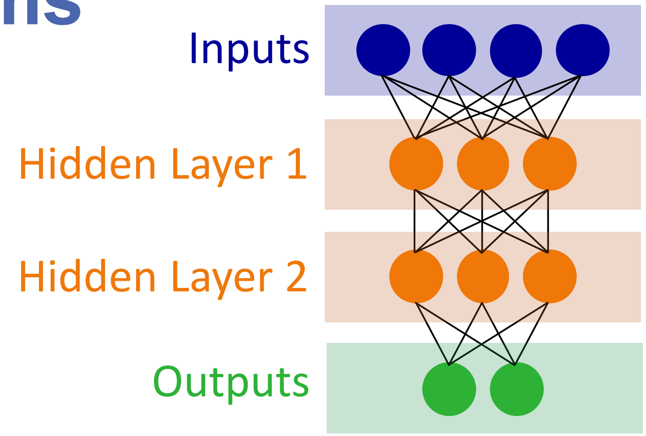
- Uses
  - Fitting models to observed data
  - Fitting models to computer generated data
  - Classification
- Types
  - Shown below
  - For your typical applications will deal with Multi Layer Perceptron (see next slide)

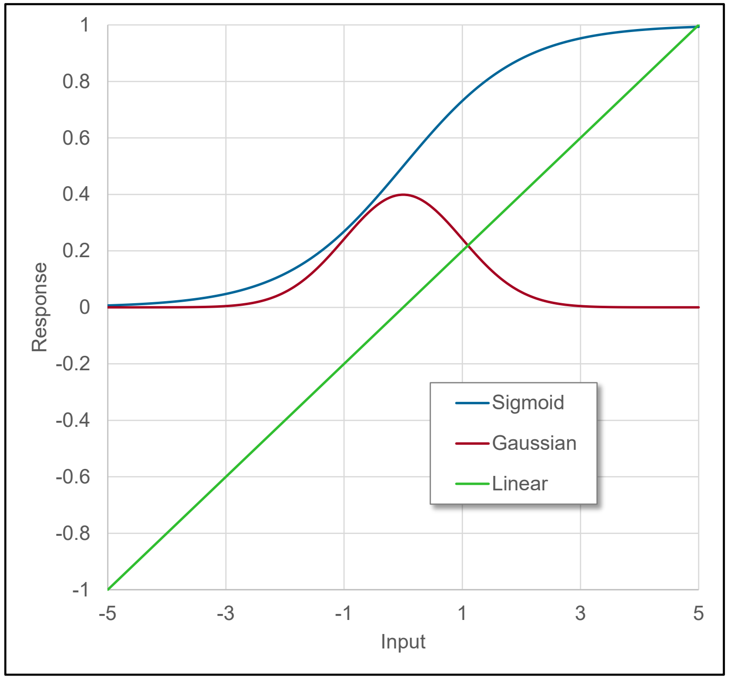
- Pros
  - Can adapt to discrete and non-linear responses
  - Computationally efficient and portable once trained
  - Can handle both discrete and continuous inputs simultaneously
- Cons
  - Easy to over-fit (more on this later)
  - Can require more extensive data set for training
  - Can be guess and check on network structure (number of nodes)



Artificial Neural Networks – Common Functional Forms

- Input layer: Regression variables
- Hidden Layers contain activation functions
- Hidden Layers (commonly one or two)
  - Sigmoid $f(x) = \frac{1}{1+e^{-x}}$
  - Gaussian $f(x) = e^{-x^2}$
  - Linear $f(x) = x$
  - ArcTan $f(x) = \text{atan}(x)$
  - Other variations, but all have similar characteristics shapes
- Output Layer
  - Linear combination of last hidden layer
  - $Y = aH1(bx + c) + eH2(fx + g) + \cdots$
- Backpropagation algorithm solves for coefficients





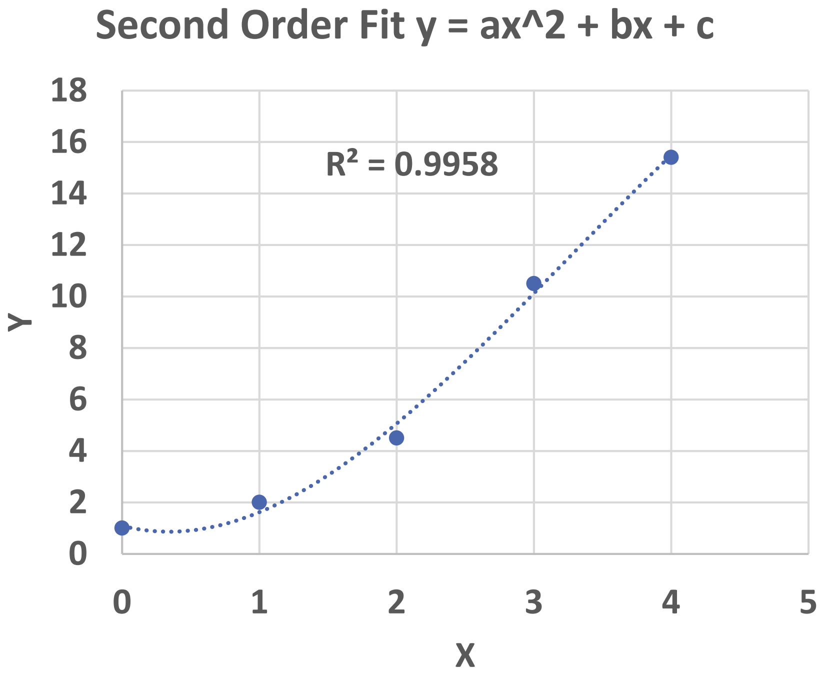
Typical Activation Functions

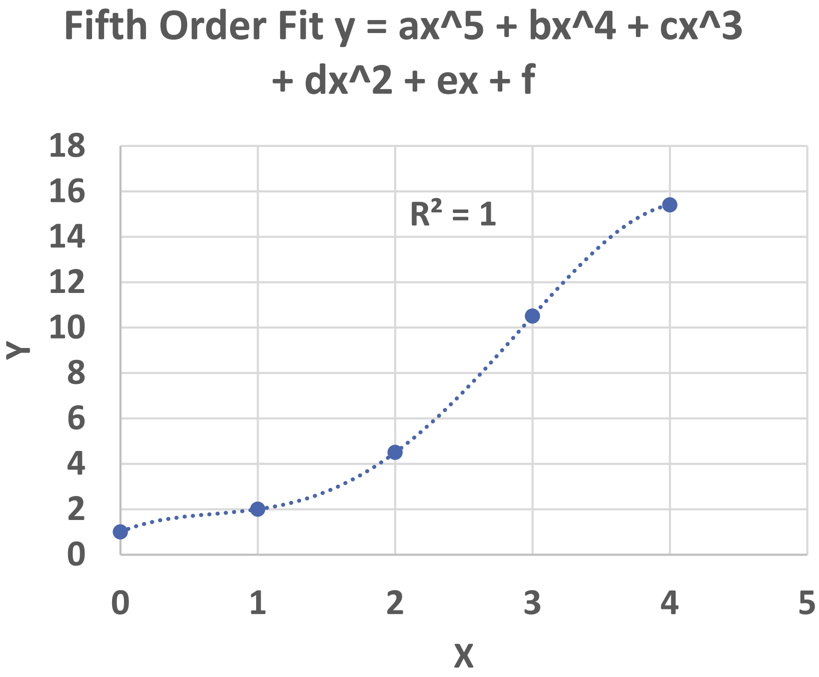
ANN Major Considerations - Overfitting

- Neural networks are more complex
- Overfitting can lead to erratic behavior
- Provides inconsistent predictions away from training points
- Can cause issues if used in numerical simulation (including APR)
  - Most models work better if underlying functions are smooth with slowly changing gradient
  - Fortunately most engineering problems are also 1st or 2nd order
- Another reason training data quality is critical
  - A neural network can fit the data if given enough degrees of freedom

ANN Major Considerations - Overfitting

- Extremely easy to overfit the model
- Take the example of y=x^2 with noise of +/- 0.5
- Plots show second and 5th order fits
- Which one is better?

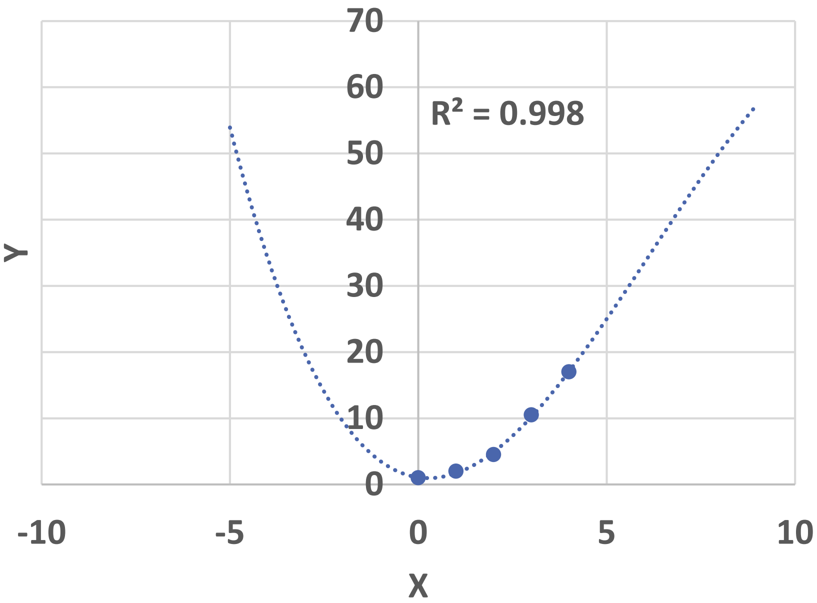




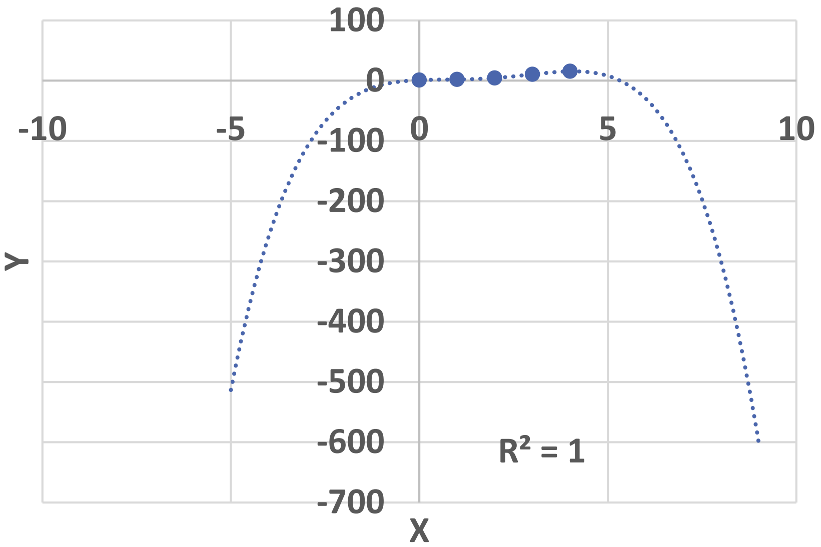
ANN Major Considerations – Overfitting and Extrapolation

What about Extrapolation?

Second Order Fit y = ax^2 + bx + c



Fifth Order Fit y = ax^5 + bx^4 + cx^3 + dx^2 + ex + f



Choosing the Network Structure

- Structure Consists of:
  - Number of nodes
  - Type of activation function in each layer
- Is an iterative process – use below as starting point
- Use error diagnostics (later on) to evaluate and compare multiple regressions

Node Types

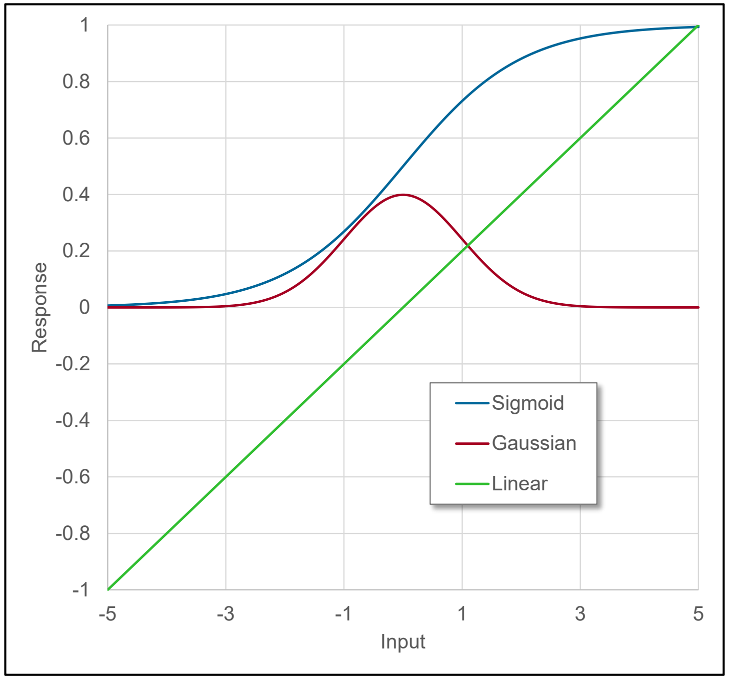
Node Type	# of nodes
Input	Defined by problem (X's)
Hidden 1 (closer to inputs)	~ number of outputs * (number of inputs)
Hidden 2 (closer to outputs)	0 < Number of outputs < number of inputs
Output	Two options: 1. Fit one neural network per output (Y) a) Easier to fit b) Simplifies network structure 2. Fit multiple outputs a) Enables coupling to be observed between Y1 and Y2 b) Often requires additional hidden nodes

Choosing the Network Structure

- Structure Consists of:
  - Number of nodes
  - Type of activation function in each layer
- Is an iterative process – use below as starting point
- Use error diagnostics (later on) to evaluate and compare multiple regressions

Types of Nodes

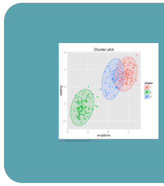
Type of Activation Function	Considerations
Linear	▪ Linear + Linear = Linear ▪ Cannot capture curvature
Gaussian	▪ Threshold function ▪ Useful for classification problems in input layer
Sigmoid Shape	▪ Most generic (and useful) ▪ Should be second layer on classification problems – can represent probability



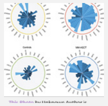
Typical Activation Functions

Common Types of Machine Learning

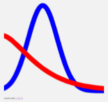
Artificial Neural Networks



Clustering Algorithms (APR often falls into this category)



Advanced Pattern Recognition



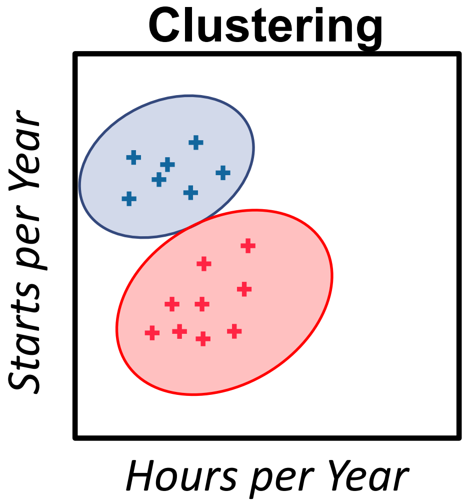
Bayesian Learning

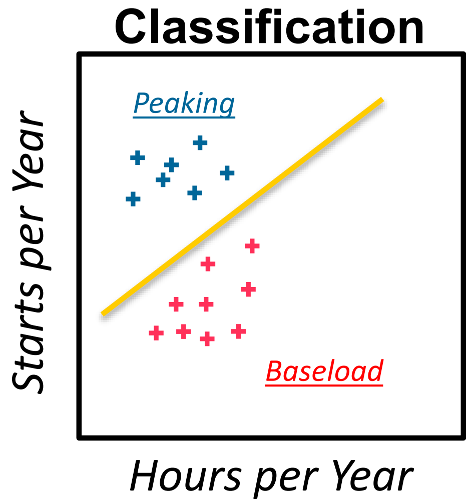
Clustering Algorithms – What Are They?

- Identify clusters of common data points in multidimensional space
- Good for unsupervised learning
- Most are geometrically based
  - Define centroids of commonality
- Balance specificity against generality
  - Could have one cluster point for every point in data set
  - Could have one or two clusters for large number of points

Clustering Algorithms - Uses

- Uses
  - Unsupervised learning
- Common Types
  - K-Means
  - Hierarchical
  - Normal Mixtures
- Pros
  - Useful when functional form of data is not known or hard to define (does not mean it does not exist!)
  - Easy to use and understand
- Cons
  - Lack good ability to extrapolate
  - Choosing the number of clusters can be difficult
  - Geometrically based!
  - Dependent on magnitude of data if data not normalized

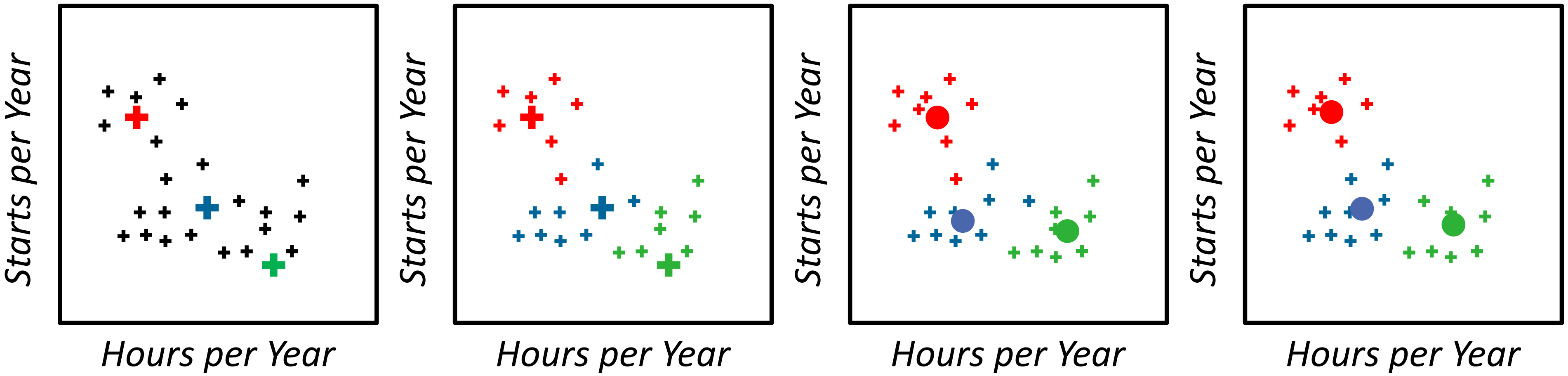




Clustering Types: K- means

- One of the more common types is called k-means clustering
- Forms clusters on k (user selected) means in the dataset
- As an example define boundaries for peaking, cycling, and baseload operation based solely on data

K-Means Clustering Process (3 clusters)



Step 1: Randomly pick three 'means'

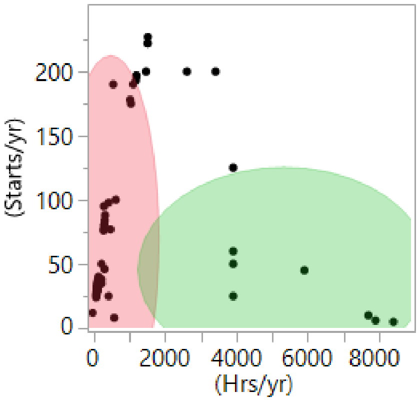
Step 2: Assign membership based on Euclidian distance

Step 3: Calculate centroid (c.g.) of each cluster

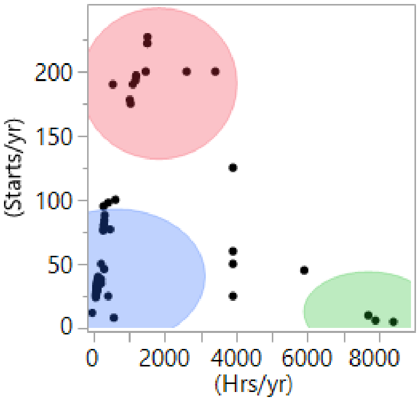
Step 4: Re-cluster on centroid – repeat 2 – 4 until converged

Clustering Types: K- means

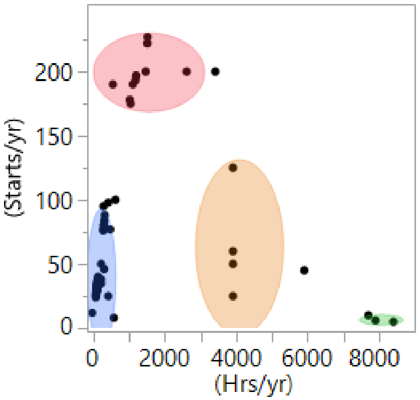
- Example – use *"actual"* data to cluster operational profiles (hours and starts per year)
- Do we get results that match expectations?



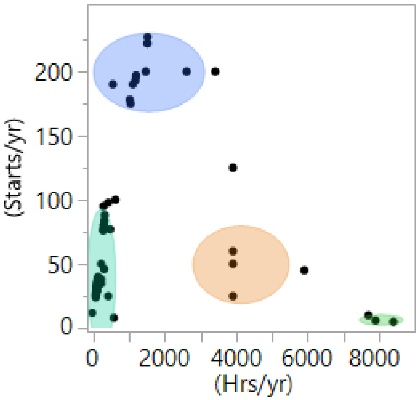
2 Clusters



3 Clusters



4 Clusters

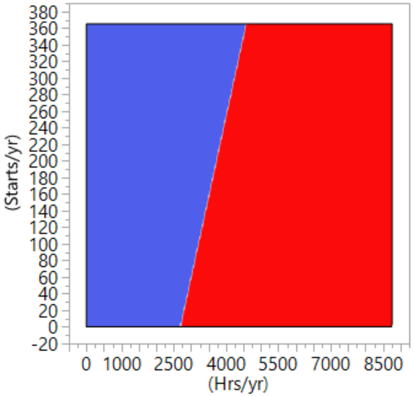


5 Clusters

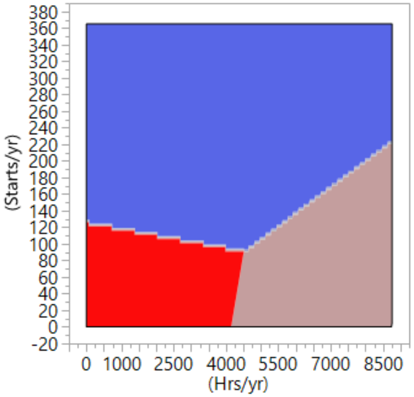
Which One is Best?

Clustering Types: K- means

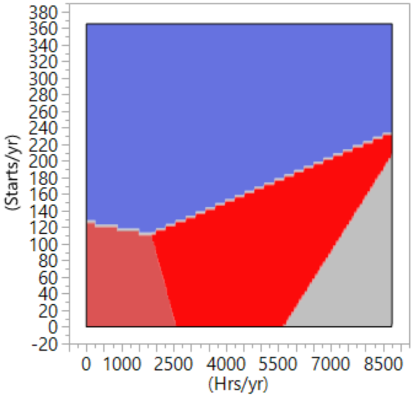
- Example – use *“actual”* data to cluster operational profiles (hours and starts per year)
- Do we get results that match expectations?



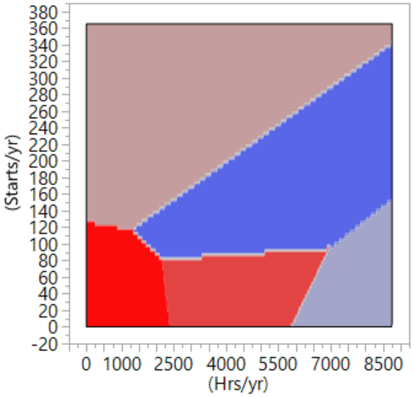
2 Clusters



3 Clusters



4 Clusters



5 Clusters

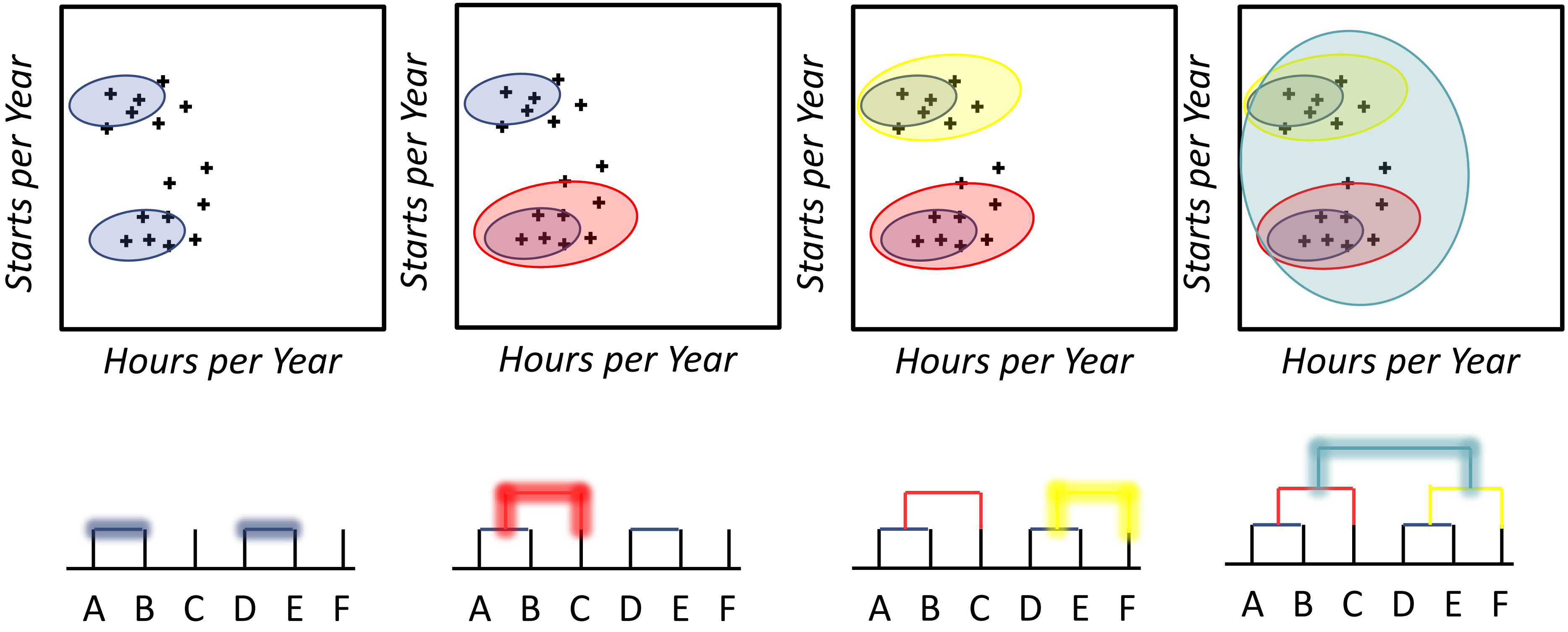
Which One is Best?
Statistically - #4

Clustering Types: Hierarchical

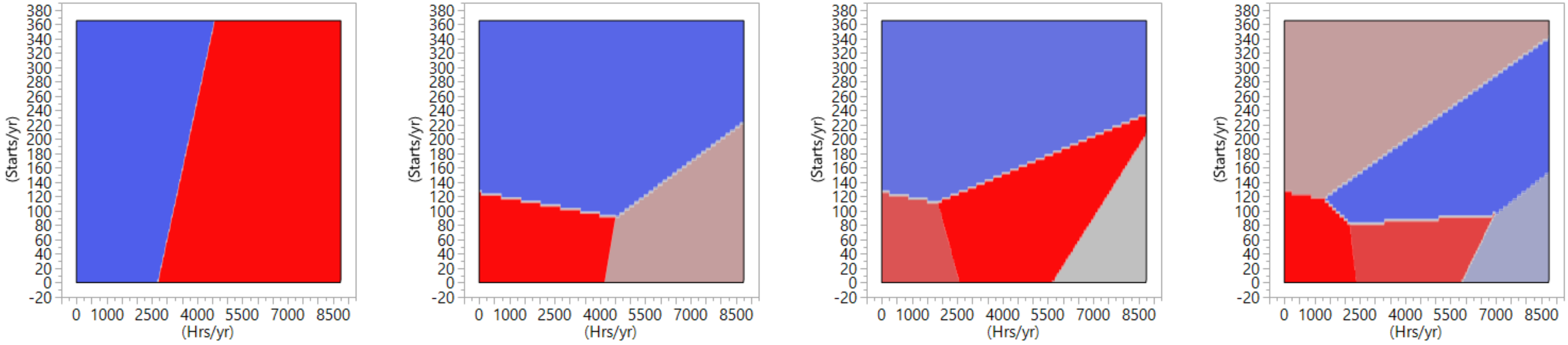
- Builds a hierarchy of clusters
- Hierarchy usually basis splits on some form of geometrically based distance
- Can be computationally more efficient
  - Calculate distances to cluster based on tree
  - Do not have to calculate distance to every cluster

Clustering Types: Hierarchical

- Builds upon flat, k-means with hierarchy of clusters



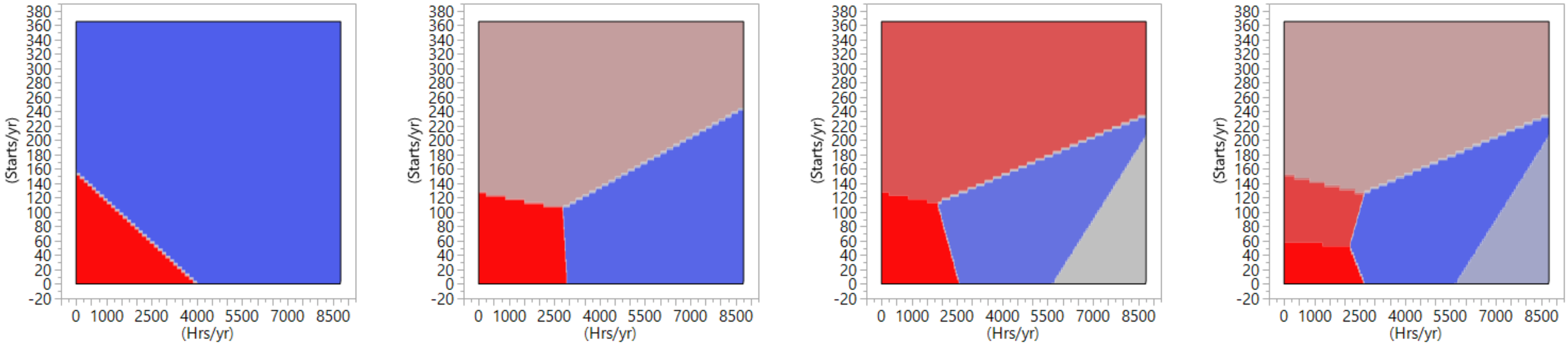
Clustering Types: Hierarchical



2 Clusters 3 Clusters 4 Clusters 5 Clusters

TOP – K-Means

BOTTOM - Hierarchical

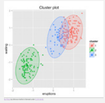


Clustering Major Considerations

- Remember that clusters are based on available data!
- No physics behind the clustering!
- Easy to over or underfit data
- Useful to identifying group membership
- Not as useful as a predictive model

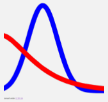
Common Types of Machine Learning

Artificial Neural Networks



Clustering Algorithms (APR often falls into this category)

Advanced Pattern Recognition - Classification



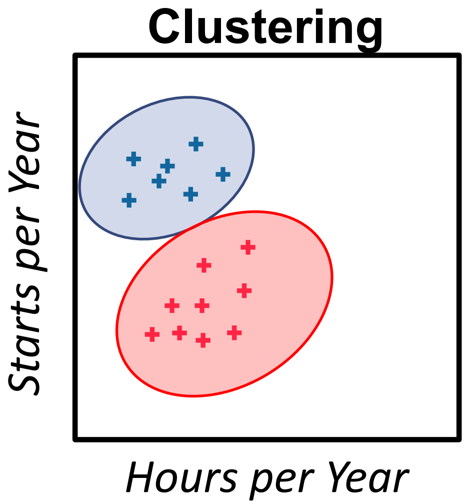
Bayesian Learning

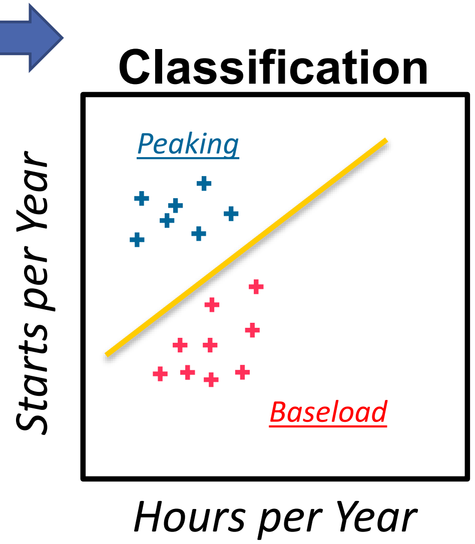
Classification Algorithms – What Are They?

- Predict class membership based on input data
- Conceptually similar to clustering, except groups are tagged in advance
- Several common types
  - Logistic Regression
  - Naïve Bayes Classifier
  - K-Nearest Neighbors
  - Decision Trees
  - Neural Networks
- All basically predict probability that certain set of inputs belongs to specific class
- Will hit highlights of each method
  - Carry operating profile as example
  - This time assume you received list of hours and starts per year with base / peak / cyclic tagged
  - You want to figure out general formula to classify additional units

Classification Algorithms – Uses

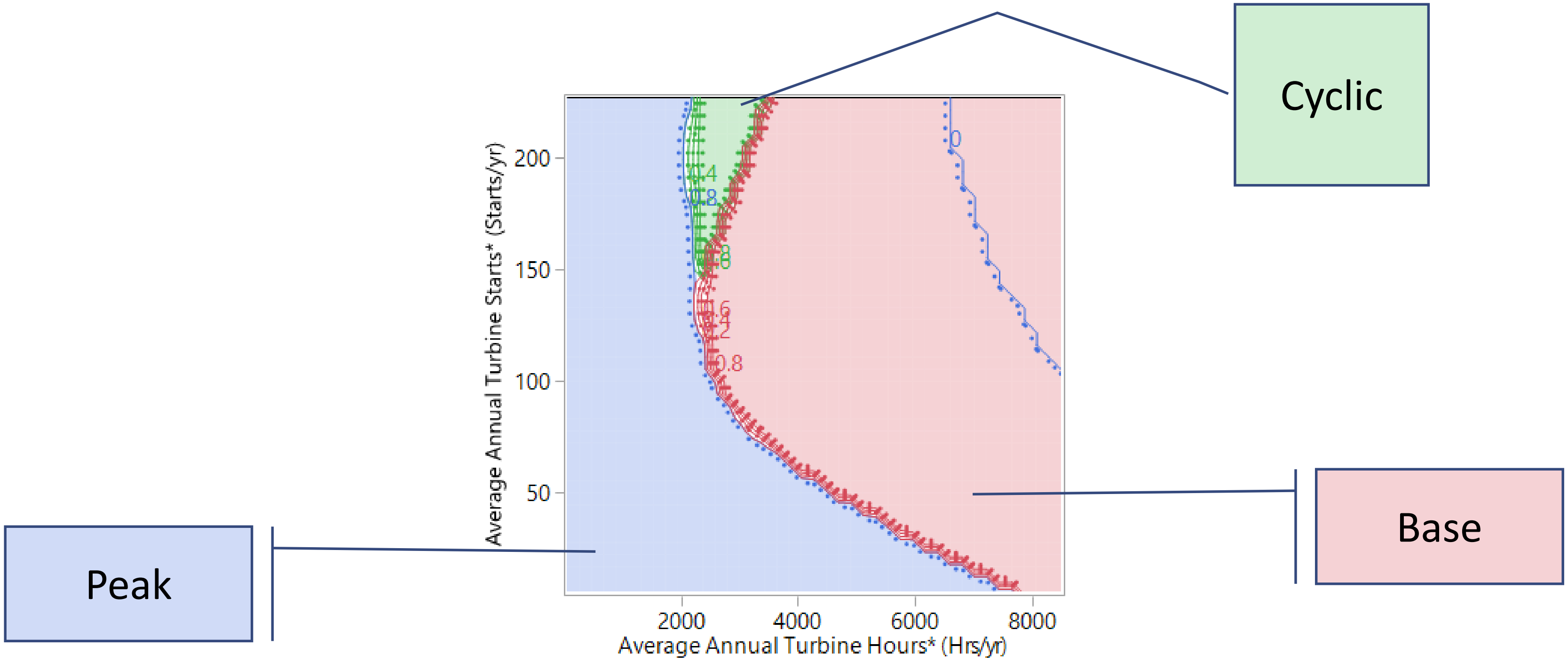
- Uses
  - Appropriate when training dataset already ‘tagged’
- Types
  - Logistic Regression
  - Naïve Bayes Classifier
  - K-Nearest Neighbors
  - Decision Trees
  - Neural Networks
- Pros
  - Several options available
  - Conceptually easy to understand
  - More complex functional forms available
- Cons
  - Relies upon prior knowledge of group membership
  - Some are geometrically based


Clustering


Classification

Classification Example Applications – Logistic Regression

- Predicts probability of something being true based on one or more correlating parameters
- $p(x) = \frac{1}{1+\exp(f(z))}$
- Z is linear transformation of inputs
- For multiple possibilities – create multiple classifiers and choose one with largest probability
  - As shown below regressions are mutually exclusive
  - In this example transition boundaries relatively tight – do not have to be depending on scatter in data



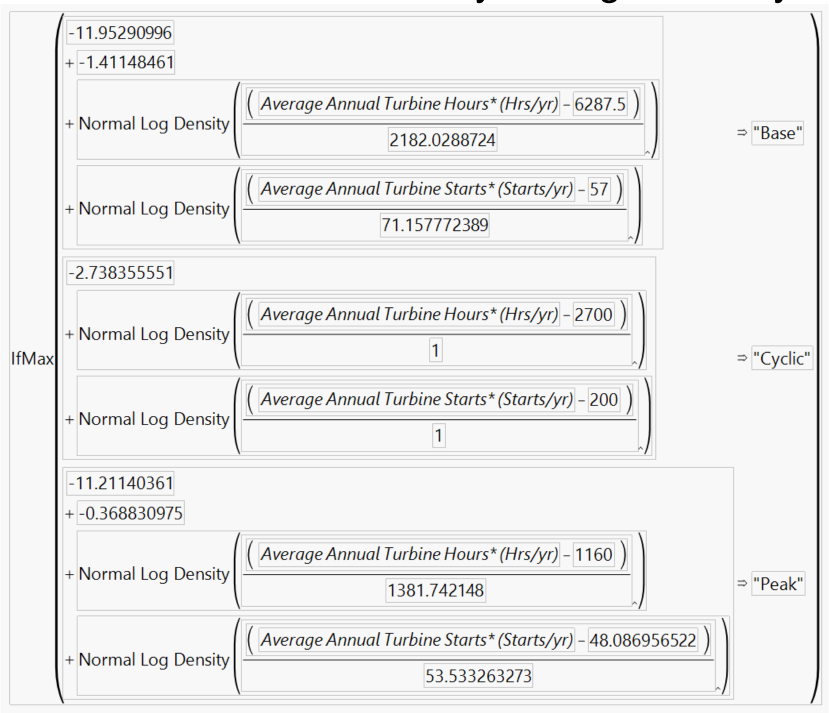
Cyclic

Base

Peak

Classification Example Applications – Naïve Bayes

- Bayes Theorem:
  - $P(Y|F) = \frac{P(F|Y)P(Y)}{P(F)}$
- Bayes Classifier assumes features are conditionally independent
  - $P(needcoffee|Monday, InClass) = P(needcoffee|Monday) * P(needcoffee|InClass)$
- Or in our example using operational profiles with continuous data
  - Combined probability using average and standard deviation of existing dataset
- Note absence of cyclic region – only had one data point

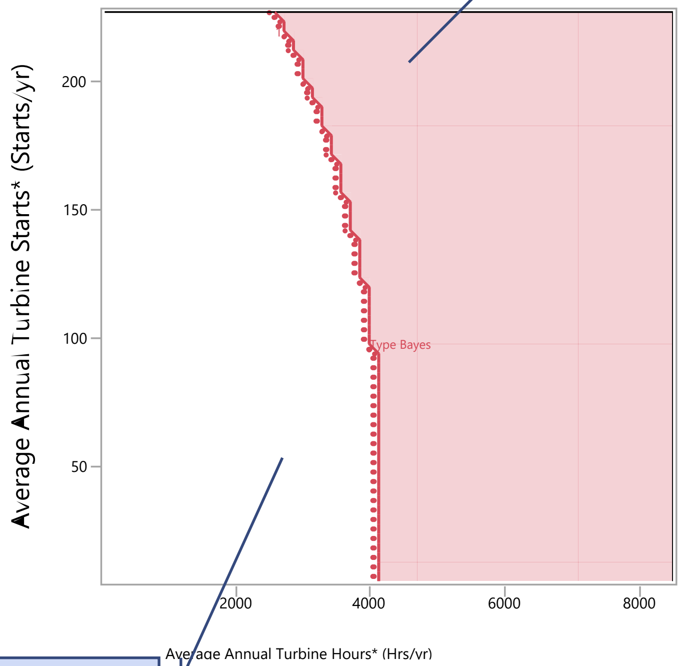


IfMax

-11.95290996
+ -1.41148461
+ Normal Log Density ((Average Annual Turbine Hours* (Hrs/yr) - 6287.5) / 2182.0288724)
+ Normal Log Density ((Average Annual Turbine Starts* (Starts/yr) - 57) / 71.157772389)
⇒ "Base"

-2.738355551
+ Normal Log Density ((Average Annual Turbine Hours* (Hrs/yr) - 2700) / 1)
+ Normal Log Density ((Average Annual Turbine Starts* (Starts/yr) - 200) / 1)
⇒ "Cyclic"

-11.21140361
+ -0.368830975
+ Normal Log Density ((Average Annual Turbine Hours* (Hrs/yr) - 1160) / 1381.742148)
+ Normal Log Density ((Average Annual Turbine Starts* (Starts/yr) - 48.086956522) / 53.533263273)
⇒ "Peak"

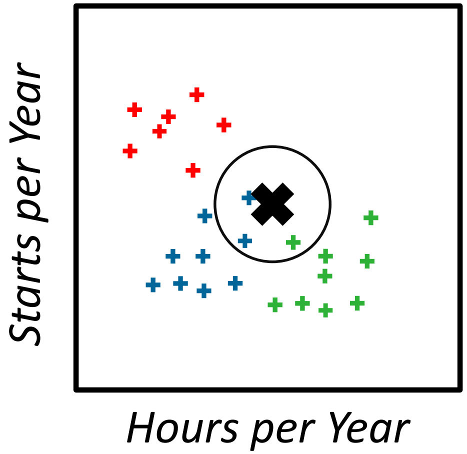


Base

Peak

Classification Example Applications – K Nearest Neighbors

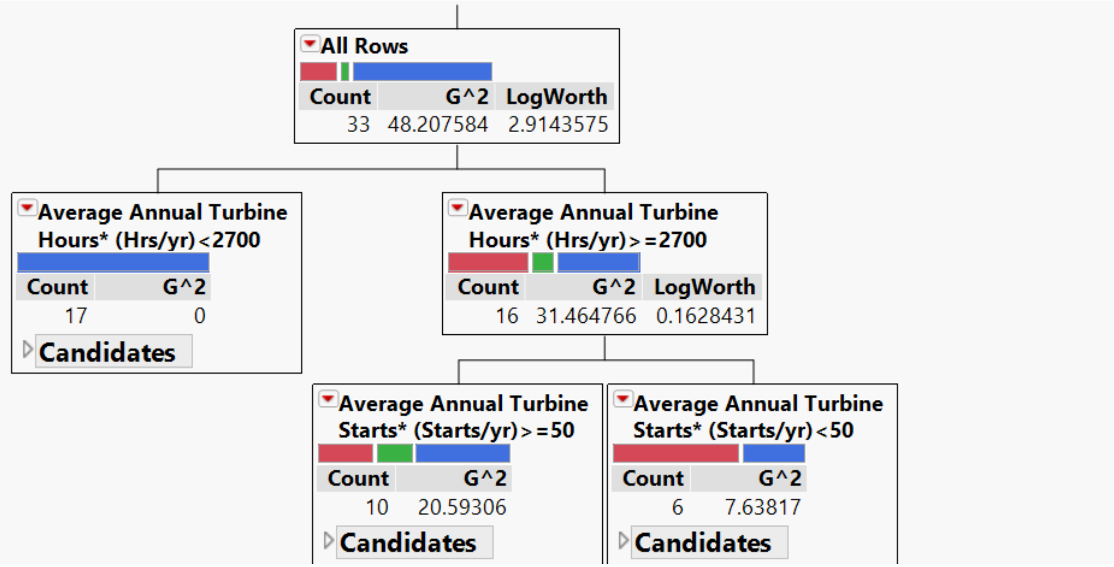
- Similar to clustering approach, except response is the average of the k-nearest neighbors
- For a new point – finds *k* nearest neighbors
- Largest number of matches yields class association
- Choosing the right *k* is trial and error



- Assume k set to three
- New point at **X**
- 3 nearest neighbors are two blue and one green
- Membership is blue

Classification Example Applications – Decision Trees

- Recursively subdivide the dataset into a decision tree
- Results in square spaces
- May not be useful if boundaries are correlated or non-linear



All Rows

Count	G^2	LogWorth
33	48.207584	2.9143575

Average Annual Turbine Hours* (Hrs/yr)<2700

Count	G^2
17	0

Candidates

Average Annual Turbine Hours* (Hrs/yr)>=2700

Count	G^2	LogWorth
16	31.464766	0.1628431

Average Annual Turbine Starts* (Starts/yr)>=50

Count	G^2
10	20.59306

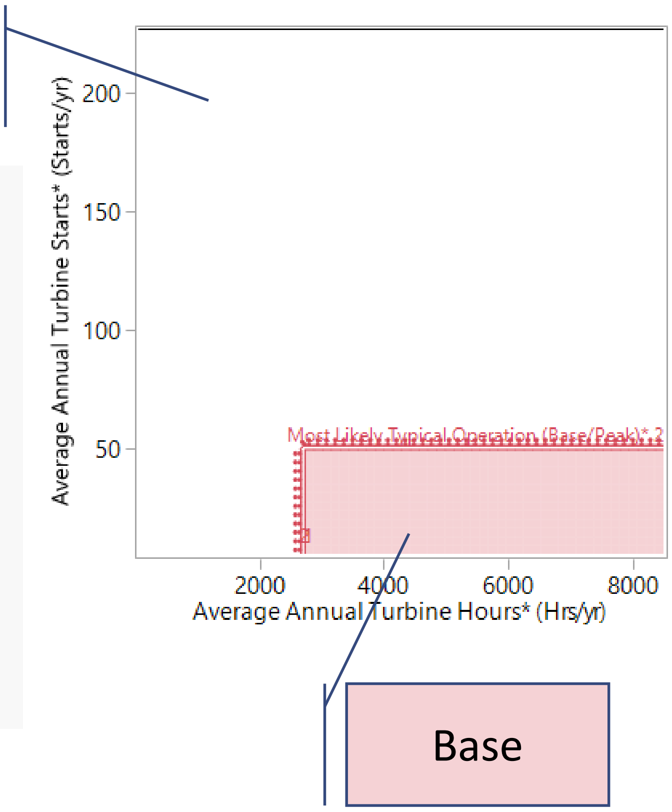
Candidates

Average Annual Turbine Starts* (Starts/yr)<50

Count	G^2
6	7.63817

Candidates

Peak



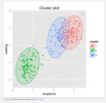
Base

Classification Major Considerations

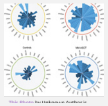
- Need to consider the shape of the space and what defines membership
  - Either / or relationship? – Decision tree
  - Closeness to existing metrics which are independent of each other – Bayes
  - Non-linear or correlated boundaries – Logistic Regression
  - Geometric similarity to existing parameters – K-Nearest Neighbors
- More advanced techniques exist, these are some of the more common ones you'll come across
- The software you use may not describe method – why it's critical to plot the results!

Common Types of Machine Learning

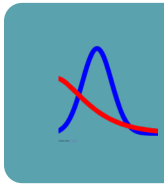
Artificial Neural Networks



Clustering Algorithms (APR often falls into this category)



Advanced Pattern Recognition

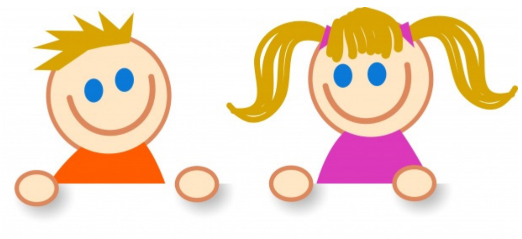


Bayesian Learning

Bayesian Networks – What Are They?

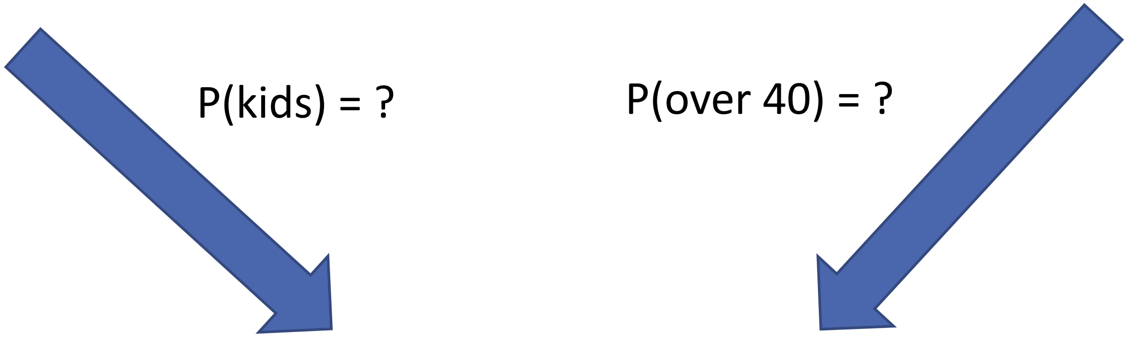
- Follows Bayes Theorem:
  - $P(Y|F) = \frac{P(F|Y)P(Y)}{P(F)}$
- The power behind Bayesian Networks lie in the fact that:
  - Prior beliefs can influence posterior (future) thinking based on new observations
  - Allow for model to learn over time as new data becomes available
  - Probabilistic

Bayesian Networks – A Simple Example



Have Kids?	Yes:	No:

Over 40?	Yes:	No:



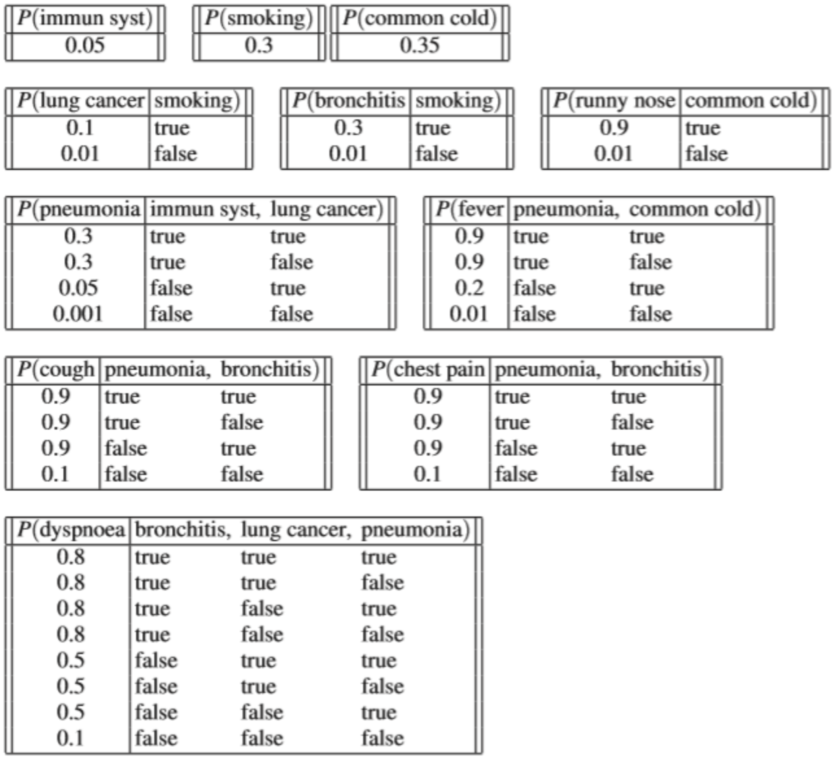
P(kids) = ?

P(over 40) = ?

$$P(Over\ 40|Have\ Kids) = \frac{P(Have\ Kids|Over\ 40)P(Over\ 40)}{P(K)}$$

Will do exercise in class

Bayesian Networks – A More “Real World” Example



P(immun syst)
0.05

P(smoking)
0.3

P(common cold)
0.35

P(lung cancer	smoking)
0.1	true
0.01	false

P(bronchitis	smoking)
0.3	true
0.01	false

P(runny nose	common cold)
0.9	true
0.01	false

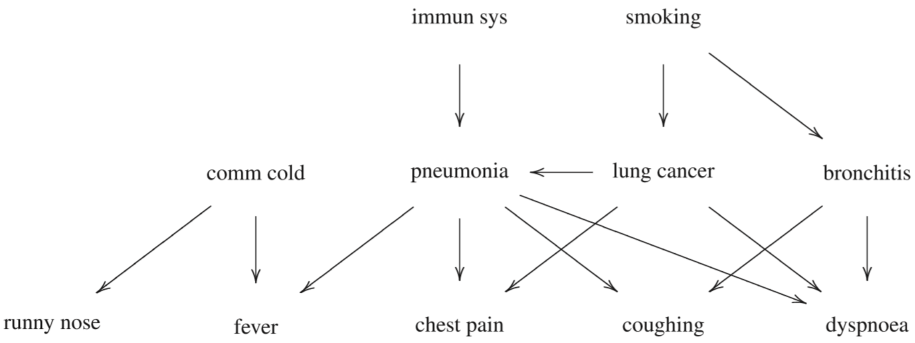
P(pneumonia	immun syst,	lung cancer)
0.3	true	true
0.3	true	false
0.05	false	true
0.001	false	false

P(fever	pneumonia,	common cold)
0.9	true	true
0.9	true	false
0.2	false	true
0.01	false	false

P(cough	pneumonia,	bronchitis)
0.9	true	true
0.9	true	false
0.9	false	true
0.1	false	false

P(chest pain	pneumonia,	bronchitis)
0.9	true	true
0.9	true	false
0.9	false	true
0.1	false	false

P(dyspnoea	bronchitis,	lung cancer,	pneumonia)
0.8	true	true	true
0.8	true	true	false
0.8	true	false	true
0.8	true	false	false
0.5	false	true	true
0.5	false	true	false
0.5	false	false	true
0.1	false	false	false



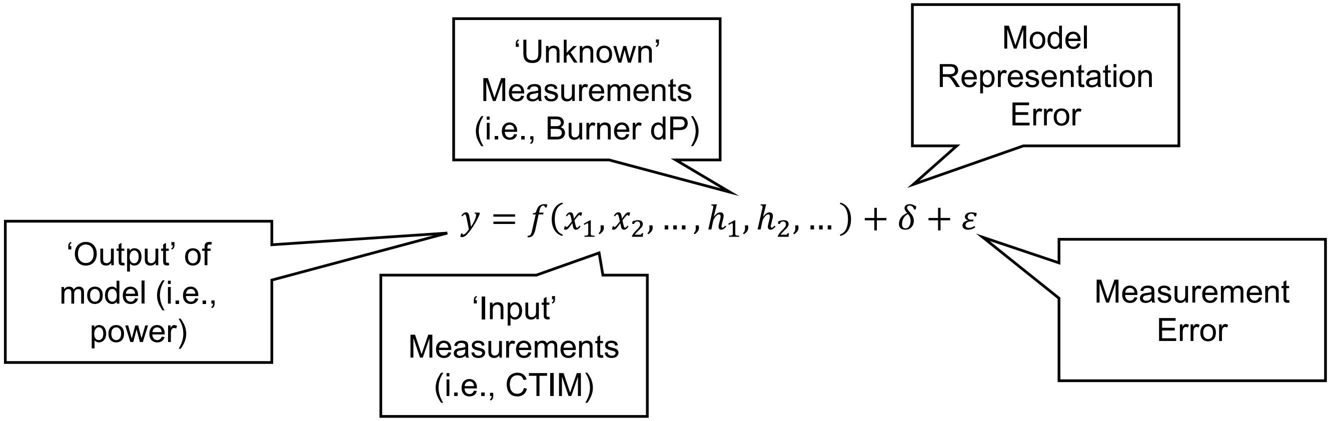
*Wiegerinick, W., Burgers, W., Kappen, B., “Bayesian Networks, Introduction and Practical Applications”

Bayesian Learning – Uses

- Uses
  - Model calibration
  - Diagnostics
  - Model Updating
- Pros
  - Flexible
  - Can learn over time
  - Suitable for discrete and continuous data
  - Good for mixed data sets
- Cons
  - Often difficult to setup
  - Validation tricky
  - Often requires coupling with additional modeling (i.e., neural networks)

Idea Behind Bayesian Calibration

- Use assumed prior belief coupled with observations to update your prior belief
- Also takes into account measurement and model representation error
  - Model representation error known from regression (prior slide)
  - Measurement error can be assumed based on sensor types
- All values are really distributions
  - Conceptually think of every measurement & prediction as having a +/- intrinsically associated with it



'Unknown' Measurements (i.e., Burner dP)

Model Representation Error

$$y = f(x_1, x_2, \dots, h_1, h_2, \dots) + \delta + \varepsilon$$

'Output' of model (i.e., power)

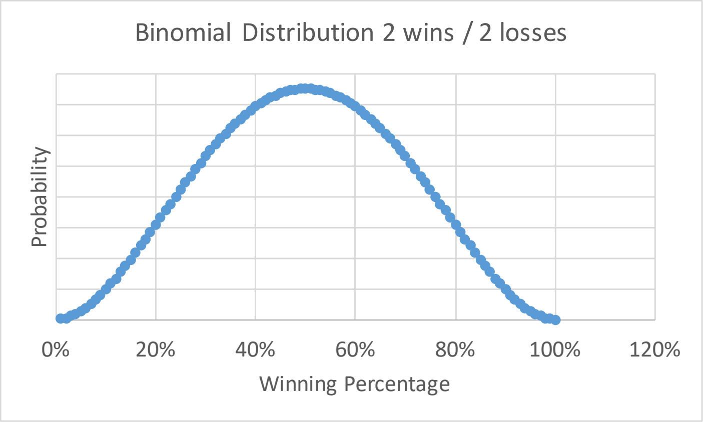
'Input' Measurements (i.e., CTIM)

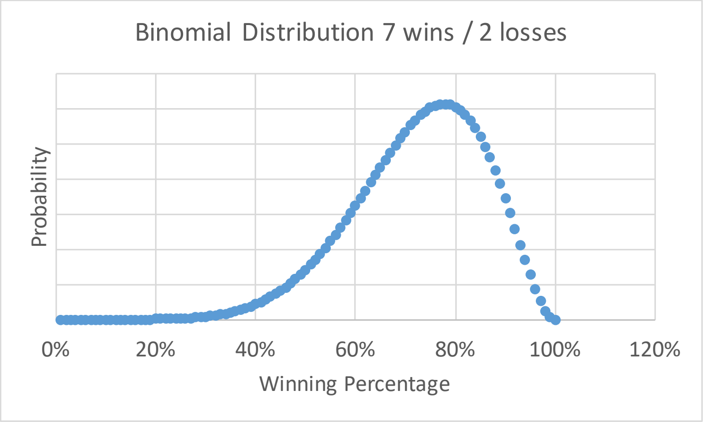
Measurement Error

Copyright 2018 by Christopher Perullo

A Simple Example – Winning Percentage

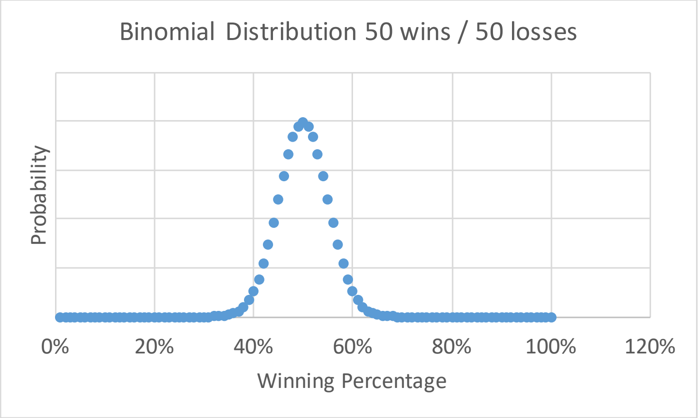
- A binomial distribution shows expected win rate
  - Useful for example since it is a 'closed form' update
- Example 1: Little prior knowledge
  - Let's assume I know my favorite team has 2 wins and 2 losses
  - The winning percentage is 50%, but how sure am I that is the true value?
  - This curve represents my prior belief
  - Looking at the spread it says I'm open to changing my opinion
- Let's say my team goes on to win 5 in a row (so they are now 7 and 2)
  - Now I'm fairly convinced they are an above 50% team
  - Still some uncertainty as to how much better

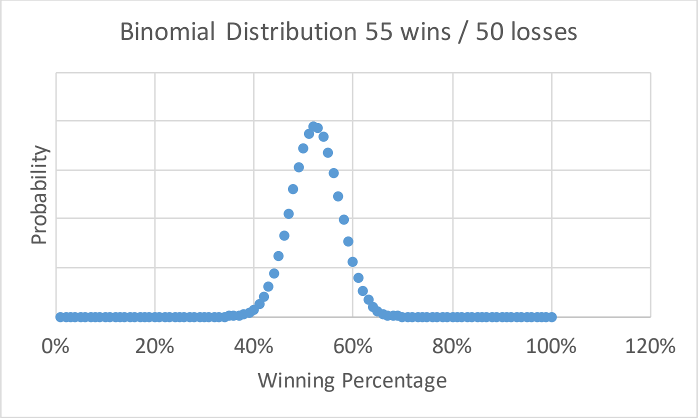




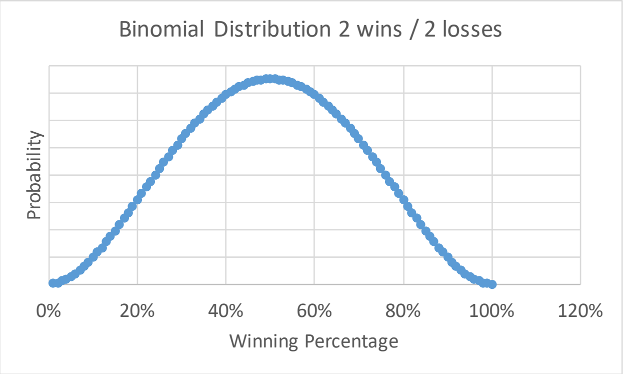
A Simple Example – Winning Percentage - Continued

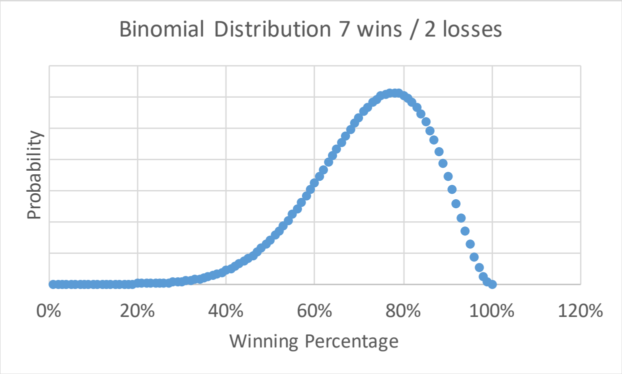
- Now let’s assume my prior knowledge is that the team has 50 wins and 50 losses
  - Same winning percentage (50%) as prior example
  - More evidence, so I’m more certain
- Assume the team wins the next 5 games, same as before
  - Now 55 wins and 50 losses
  - Still shits my opinion, but the meat of my opinion is that they’re still close to a .500 team





Winning Percentage – Putting into Bayesian Speak





5 game winning streak

Prior Belief + **New Observations** = **Posterior Knowledge**

Gas turbine model more complex, but same basic idea:

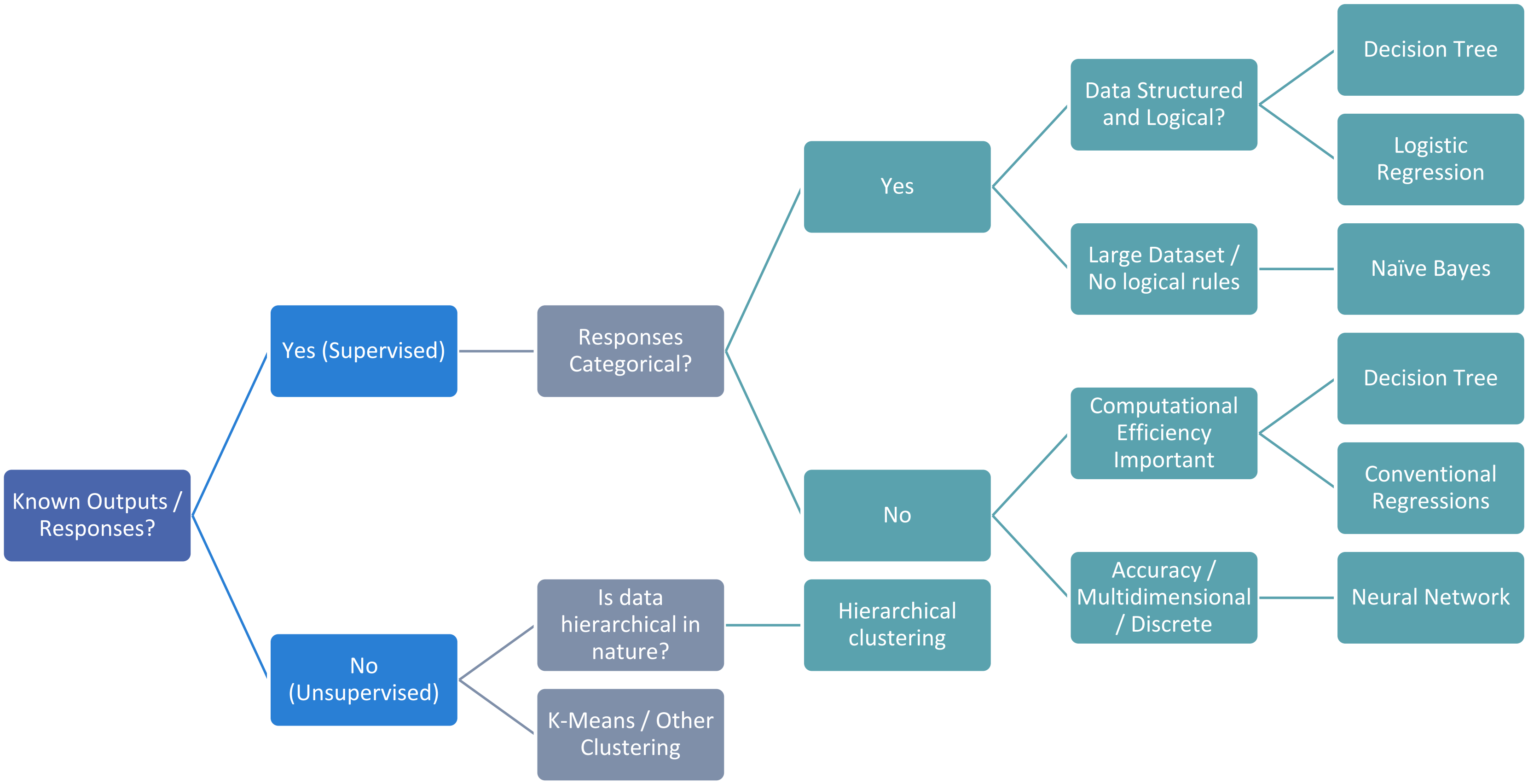
There are health and performance parameters which influence the performance of the machine – we want to estimate them based on our working knowledge of the hardware

Selecting the Right Modeling Approach

Characteristics to Consider

- Do I have specific responses (outputs)?
- Are my responses:
  - Continuous?
  - Discrete Numerical?
  - Categorical?
  - Mixed?
- Is the training data synthetic or measured?
  - How much noise in your dataset?
  - Can you denoise the data through signal analysis?
  - Reconcile with a model?
- How noisy is your dataset?

Model Type Selection Cheat Sheet


Known Outputs / Responses?
Yes (Supervised)
No (Unsupervised)
Responses Categorical?
Yes
No
Data Structured and Logical?
Decision Tree
Logistic Regression
Large Dataset / No logical rules
Naïve Bayes
Computational Efficiency Important
Decision Tree
Conventional Regressions
Accuracy / Multidimensional / Discrete
Neural Network
Is data hierarchical in nature?
Hierarchical clustering
K-Means / Other Clustering

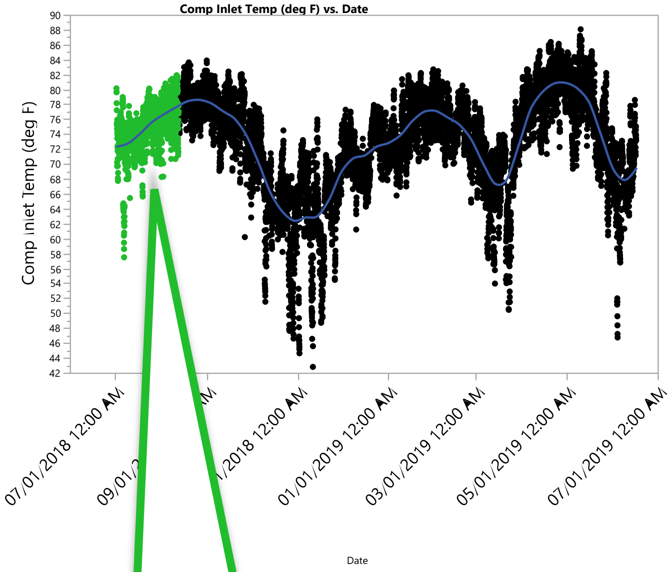
Model Creation Process

General Model Creation Process

- Constructing a training set
  - Identifying a good training data set
  - Real or Simulated Data?
  - Identifying good training regions
  - Outliers vs. ‘bad’ data
- Identifying Responses
  - Continuous
  - Discrete (classification)
  - *Probabilistic
- Train the Model
- Evaluating Model Accuracy
  - Actual vs. Predicted
  - Residual vs. Predicted
  - Model Fit and Representation Error
  - Diagnosing bad model fits

Identifying Training Set

- Need to consider applicability of model
  - Do you have a good coverage of operating conditions?
  - Will the resulting model need to extrapolate?
- Critical continuous measurements to consider:
  - Compressor inlet temperature
  - Inlet pressure drop
  - Exhaust pressure drop
  - Inlet guide vane angle
  - RPM
  - Fuel heating value
  - Ambient Pressure
- Less critical, still important:
  - Fuel temperature
- Consider if you want to track only base load conditions



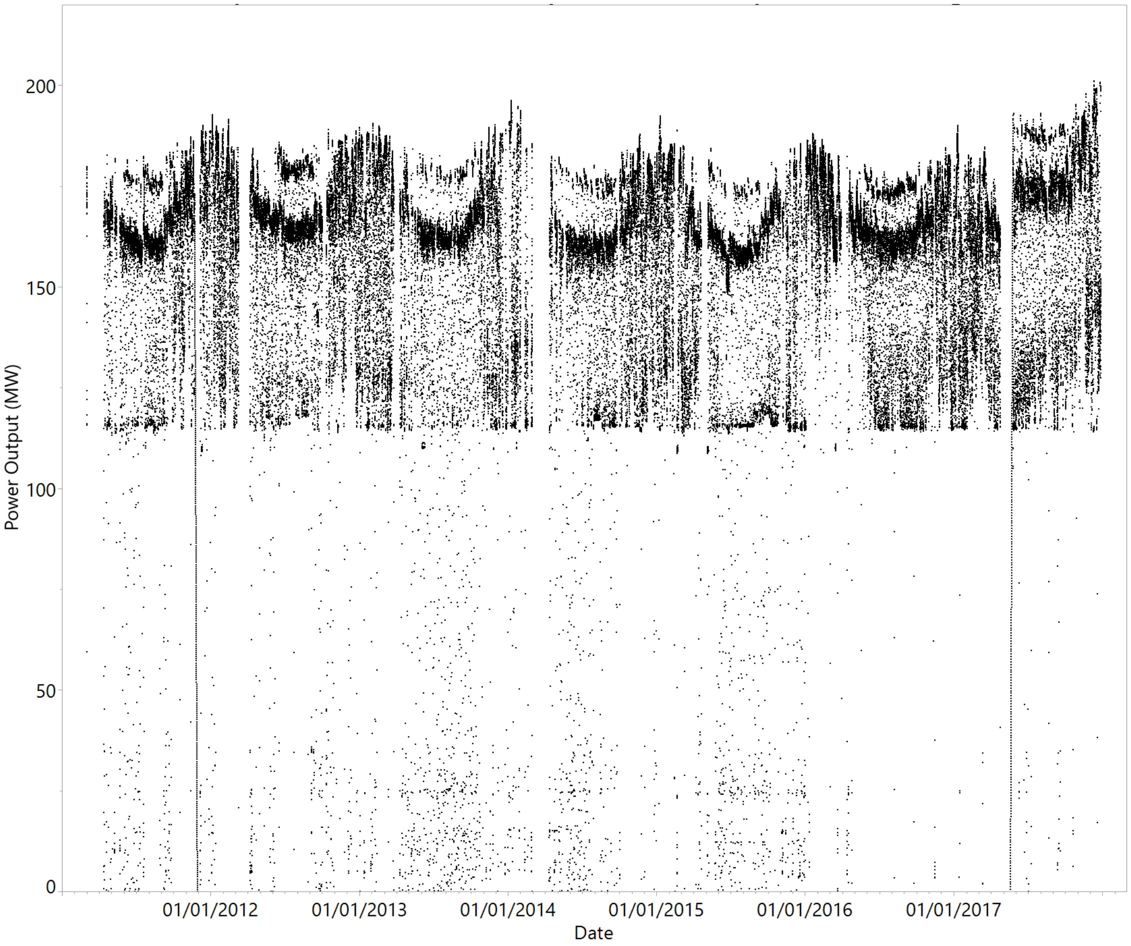
Using this training data will yield bad results later in the year

Cleaning GT Data - Transients

- Want to remove load swings from data set
- Thermal heat soak takes time
- Recommend removing data ~15 minutes before and after load change
- Cannot use MW to determine this as it changes with operating conditions
- In combined cycle operations
  - Use inlet guide vane angle to determine load changes
- In simple cycle operations
  - Depends on control curve – if combined cycle control curve use inlet guide vane angle
  - If simple cycle control curve – use estimated firing temperature – if not available could use exhaust gas temperature
- AGP / Model Based control can make this more difficult
  - Control curve could be dynamic – more on this in a minute

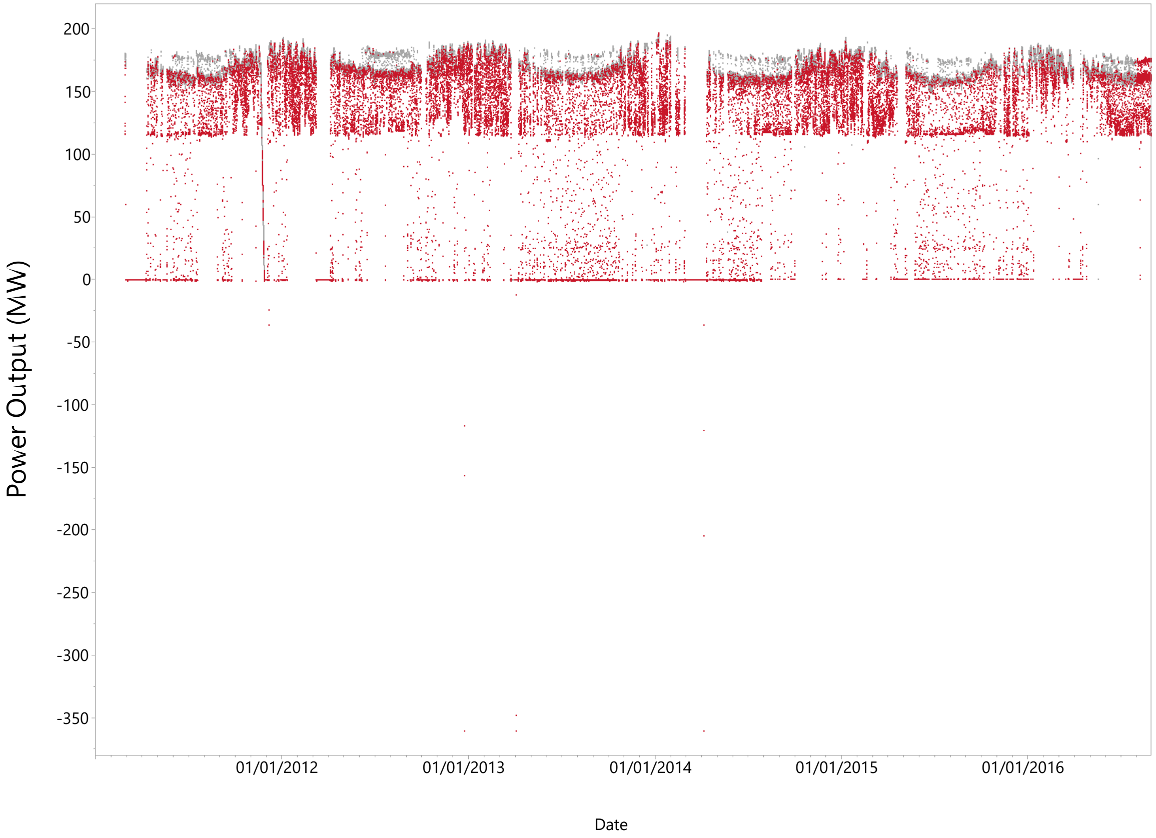
Cleaning GT Data – Identifying Baseload Conditions

How do you identify baseload conditions?



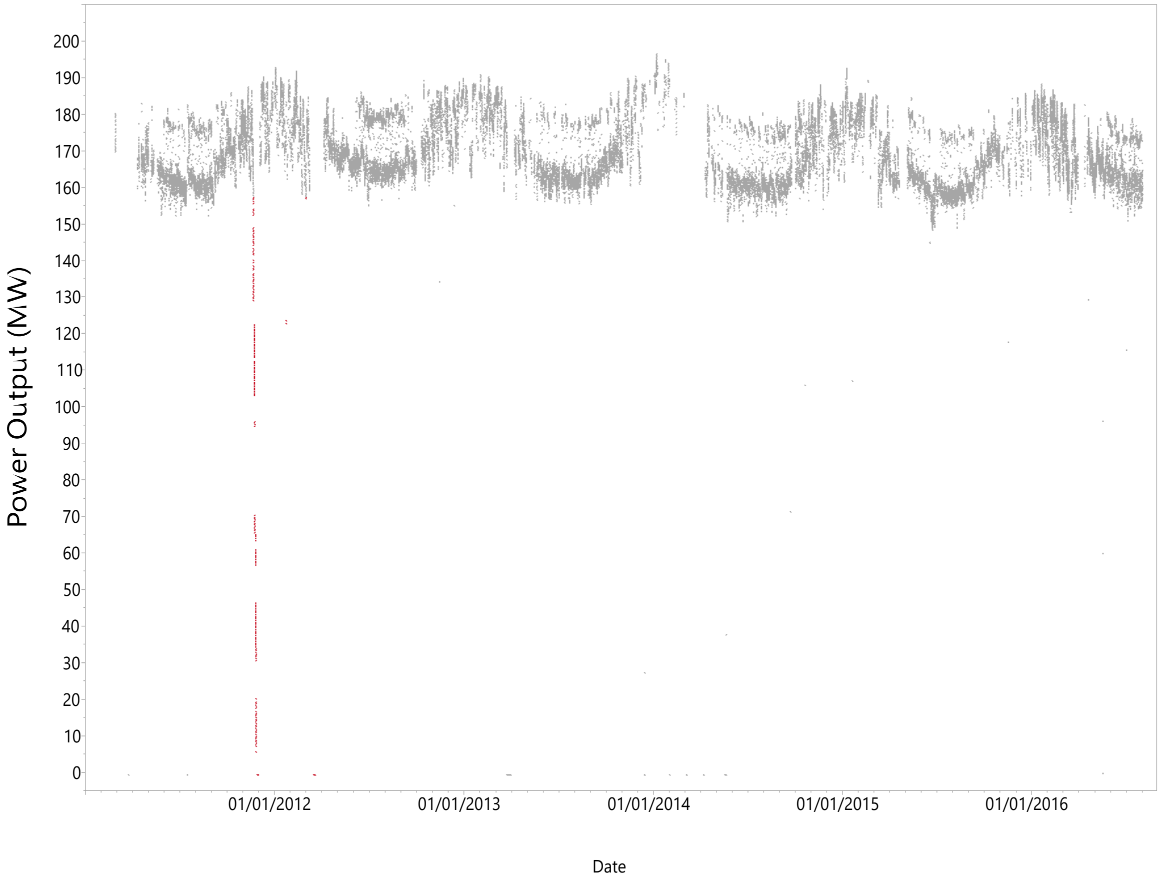
Cleaning GT Data – Identifying Baseload Conditions

1. Constrain RPM >= 3,600 (or 3,000)
2. Constrain IGV to full open



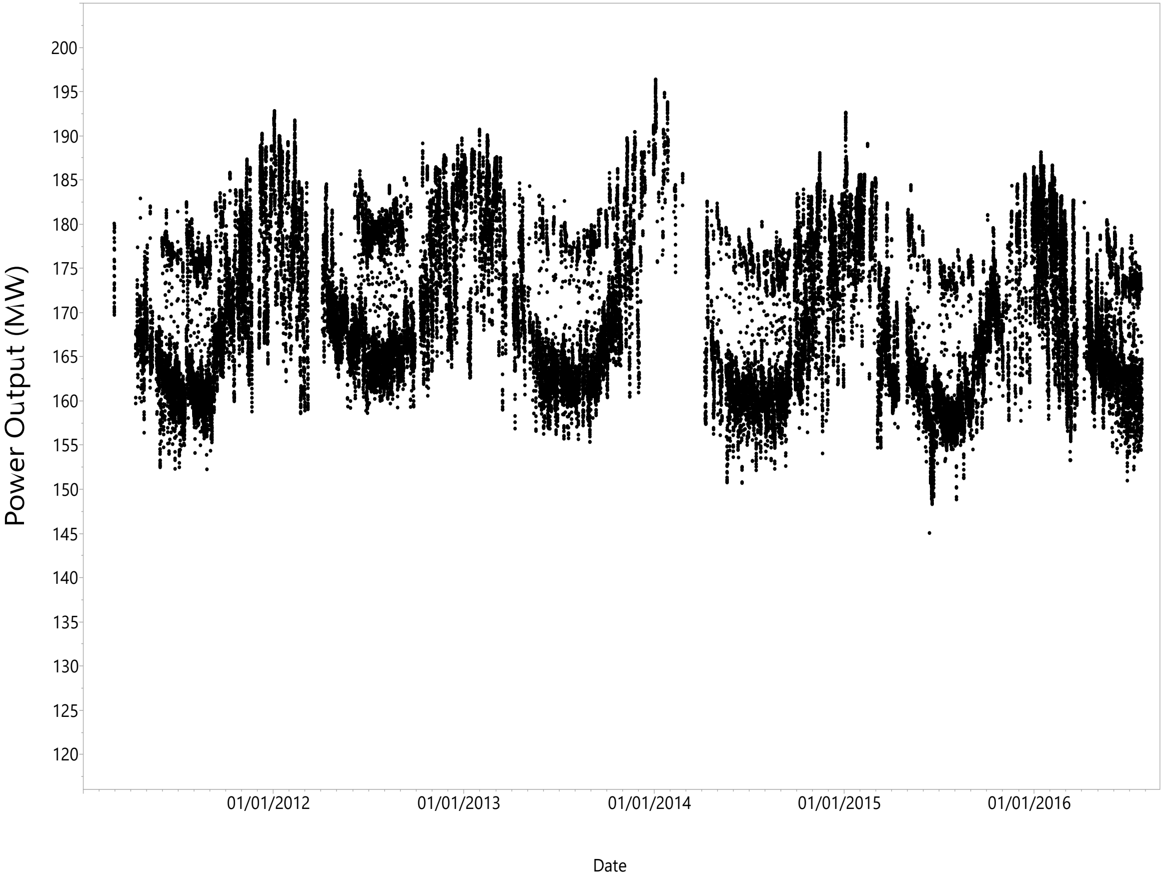
Cleaning GT Data – Identifying Baseload Conditions

1. Constrain RPM >= 3,600 (or 3,000)
2. Constrain IGV to full open
3. Visually remove remaining outliers (or pre-process to remove those without tags)



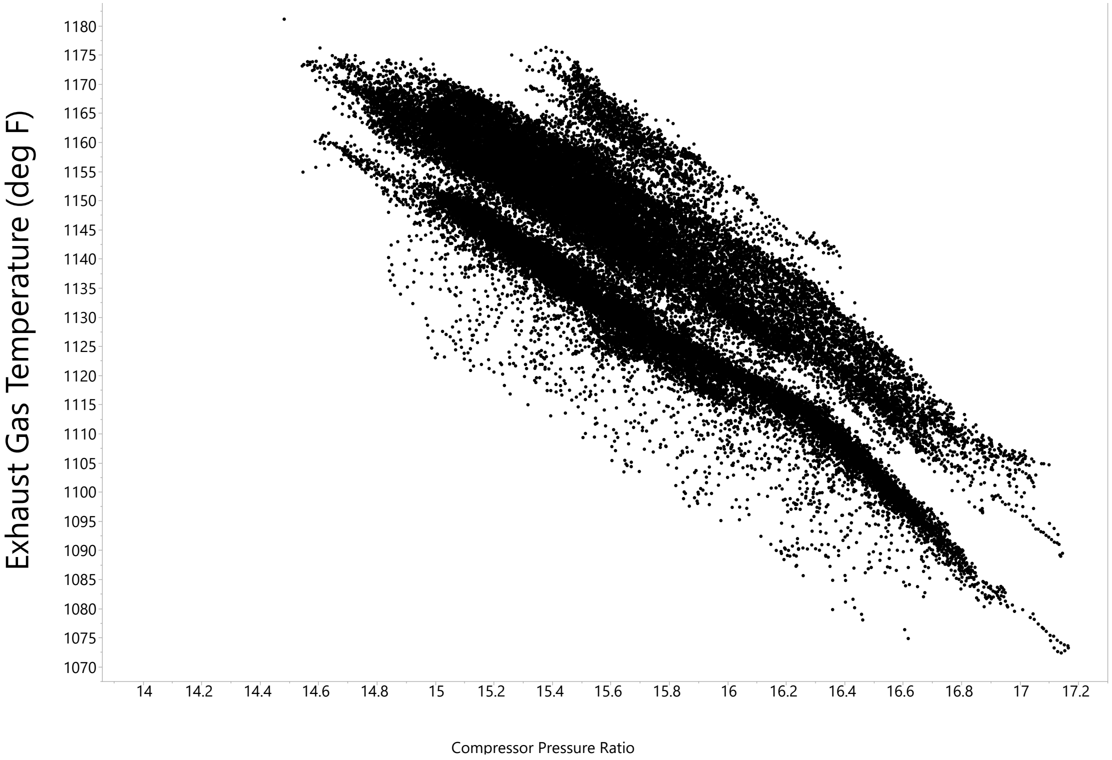
Cleaning GT Data – Identifying Baseload Conditions

1. Constrain RPM >= 3,600 (or 3,000)
2. Constrain IGV to full open
3. Visually remove remaining outliers (or pre-process to remove those without tags)
4. NOW WHAT?



Cleaning GT Data – Identifying Baseload Conditions

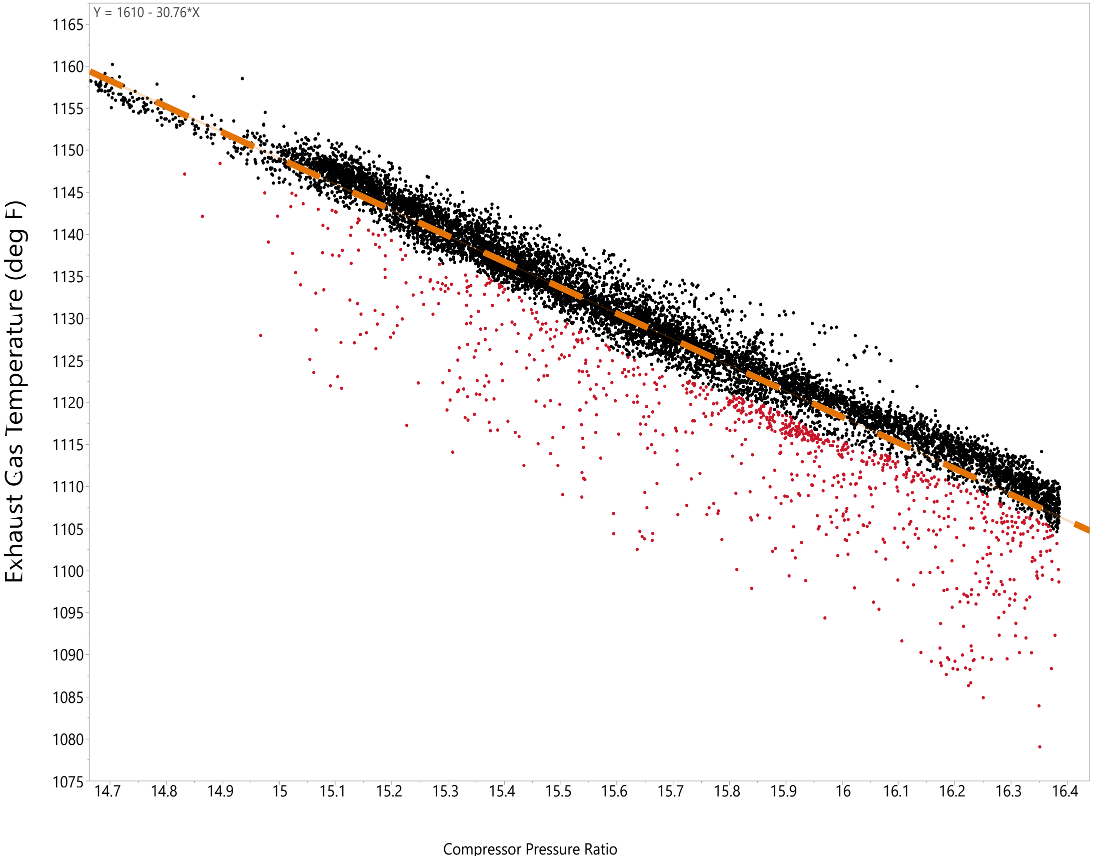
1. Plot Exhaust gas temperature vs. CDT or CPR (for DLN)
2. Should be a single line
3. Multiple lines indicate hardware or control changes
4. If control changes, must have control curve represented in model



For this example, assume single control curve

Cleaning GT Data – Identifying Baseload Conditions

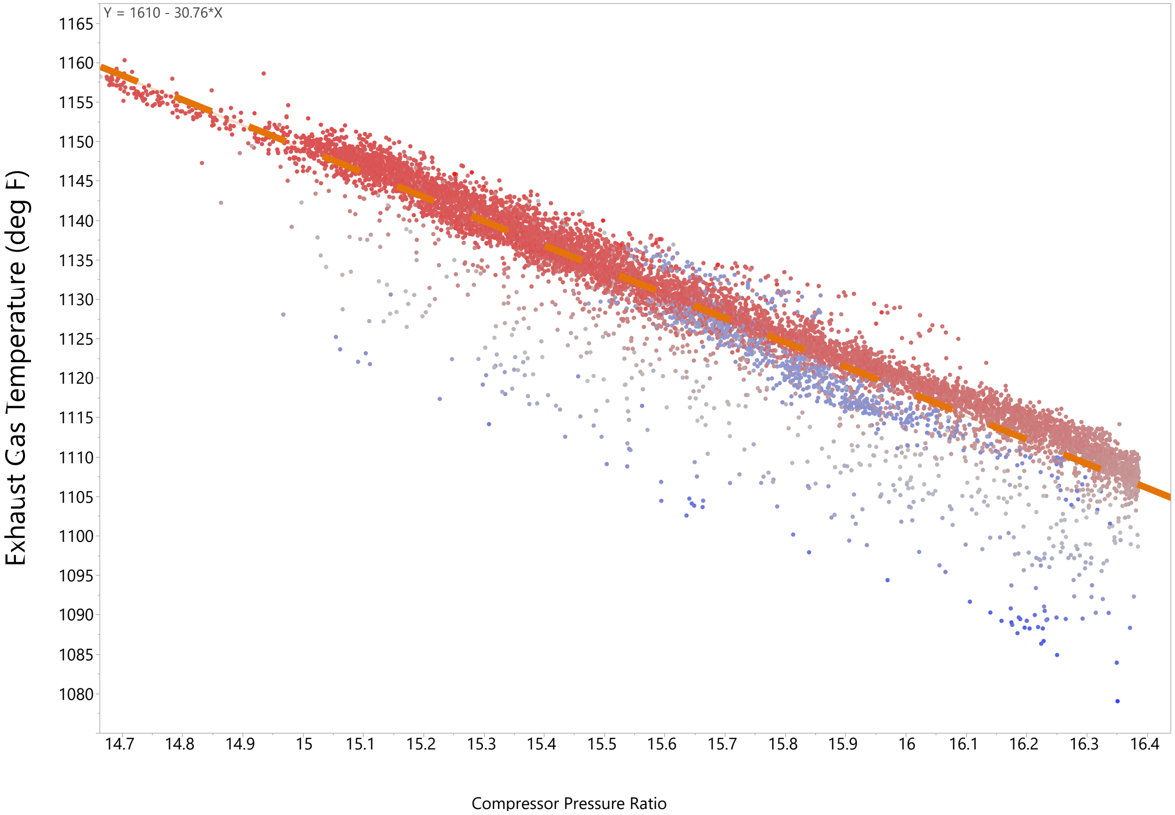
1. Fit line to data
2. Remove points outside of +/- 10 degrees
3. WHAT IF YOU HAVE AGP / MODEL BASED CONTROL?



Control curve not static if MBC present

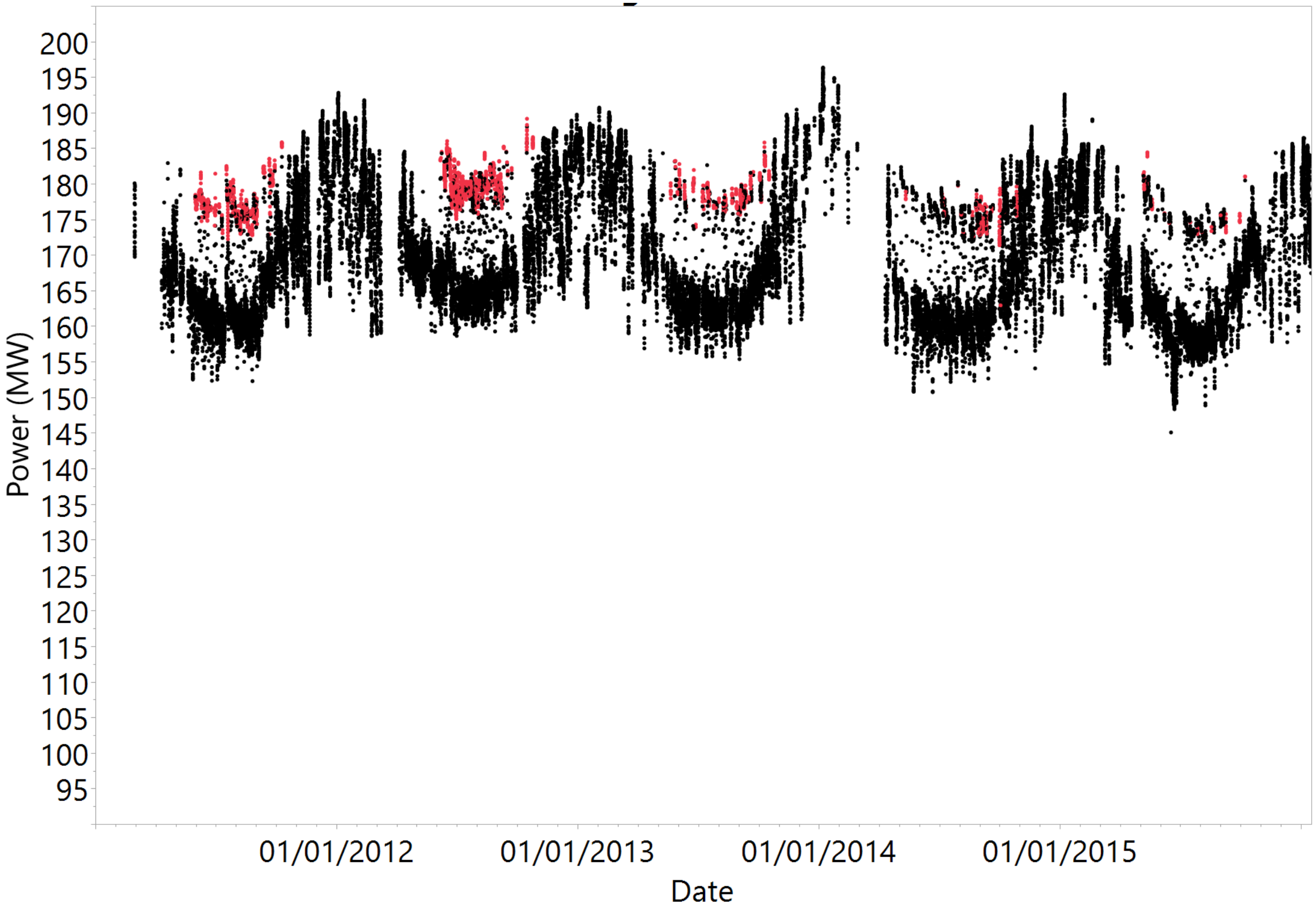
Cleaning GT Data – Identifying Baseload Conditions

1. Fit line to data
2. Remove points outside of +/- 10 degrees
3. WHAT IF YOU HAVE AGP / MODEL BASED CONTROL?
4. Can use Tfire calculation as well if trusted



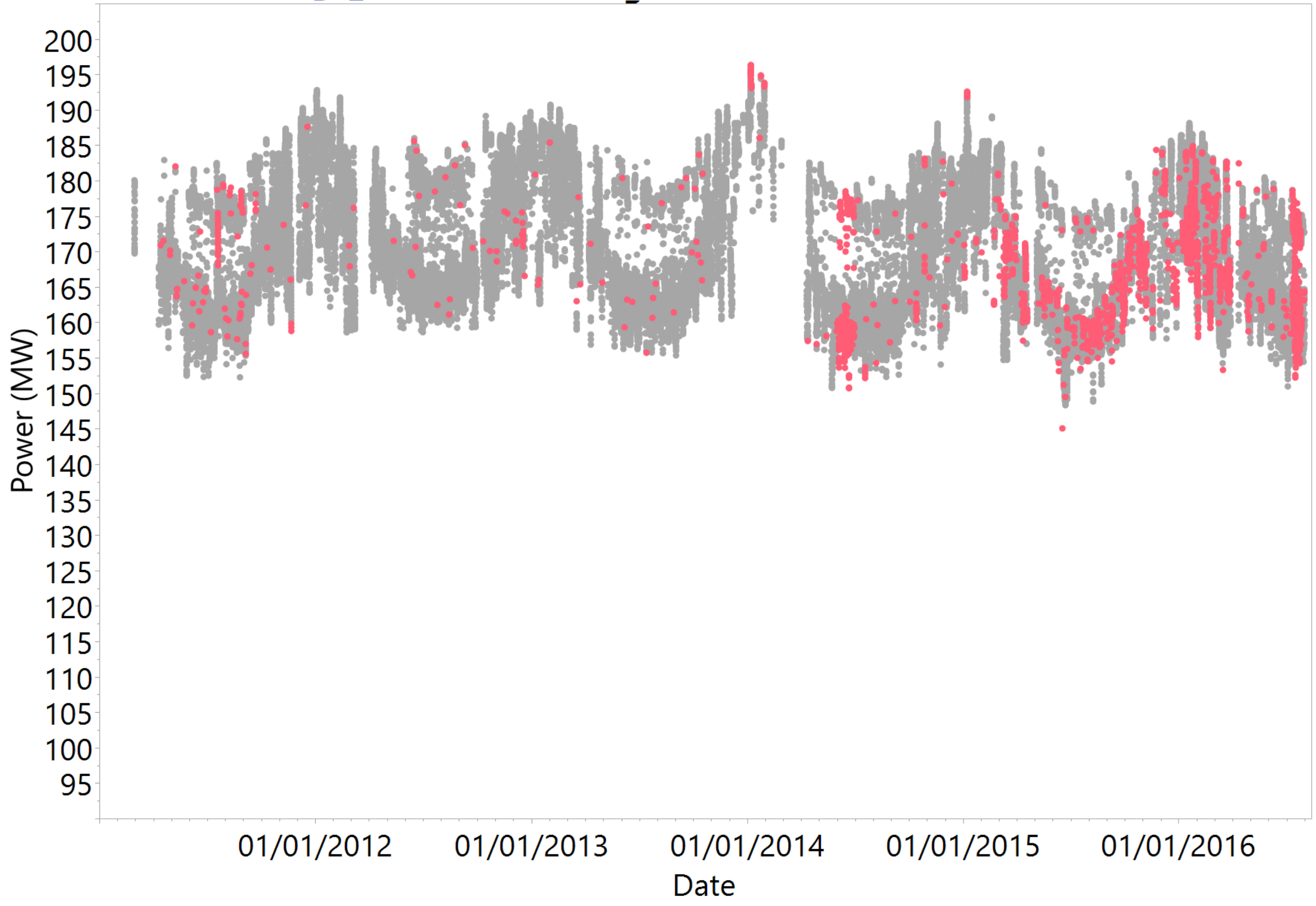
Control curve not static if MBC present

Cleaning GT Data
Discrete Operating Modes



What are the red points?

Cleaning GT Data – Discrete Operating Modes



They are not statistical outliers – shown highlighted

Cleaning GT Data
Discrete Operating Modes

- Red points on prior slide are steam injection
- Other discrete modes to screen for:
    - Inlet bleed heat
    - Steam / Water injection
    - Peak firing
    - Fuel type (liquid / gas)
    - Evaporative cooling / inlet chilling
        - Usually can be lumped as continuous parameters if compressor inlet temperature is tracked
        - Sometimes cause non-uniform flow which leads to erroneous sensor measurements
        - Suggest using if large scatter in measurements correlates with inlet cooling use

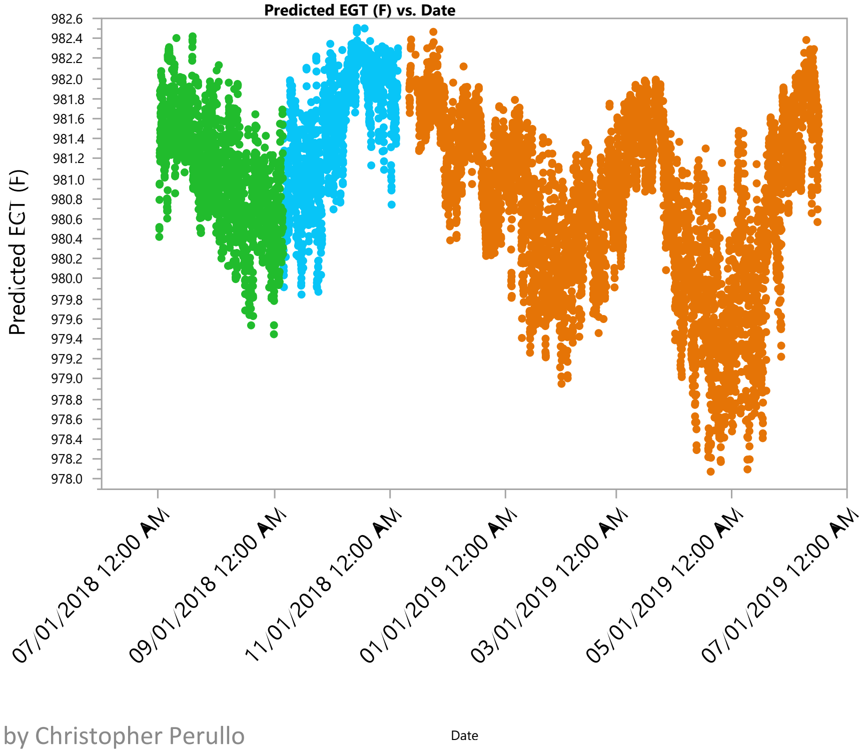
Training the Model

- Specifics depend on the software you are using
- Before clicking the ‘go’ button
  - Make sure data is properly segmented into
    - Training
    - Verification
    - Validation
  - Some software does this automatically (most don’t)
  - Understand modeling options
- It’s ok if you don’t understand math behind every option – try them all!
- Process on next few slides will allow you to objectively compare the options

Training vs. Validation Data

- Recommend splitting cleaned training data into two regions
  - Base training and verification set:
    - Training data: 75%
    - Verification data: 25%
  - Validation set:
    - Should contain full coverage over training region and small scale extrapolation if available

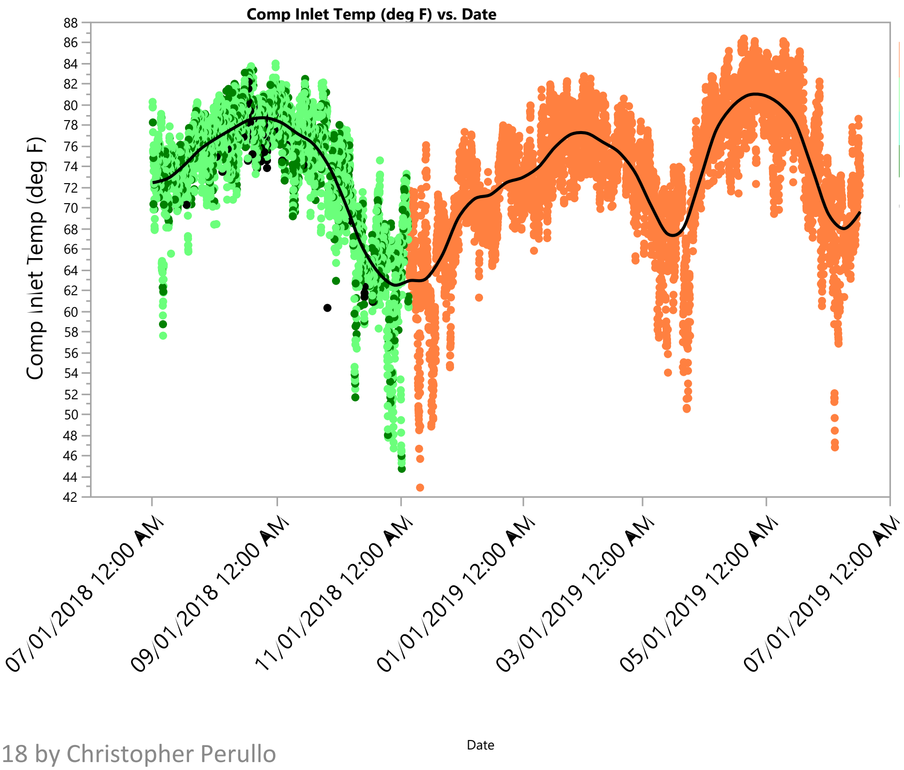
What’s wrong with this selection of training and verification data?



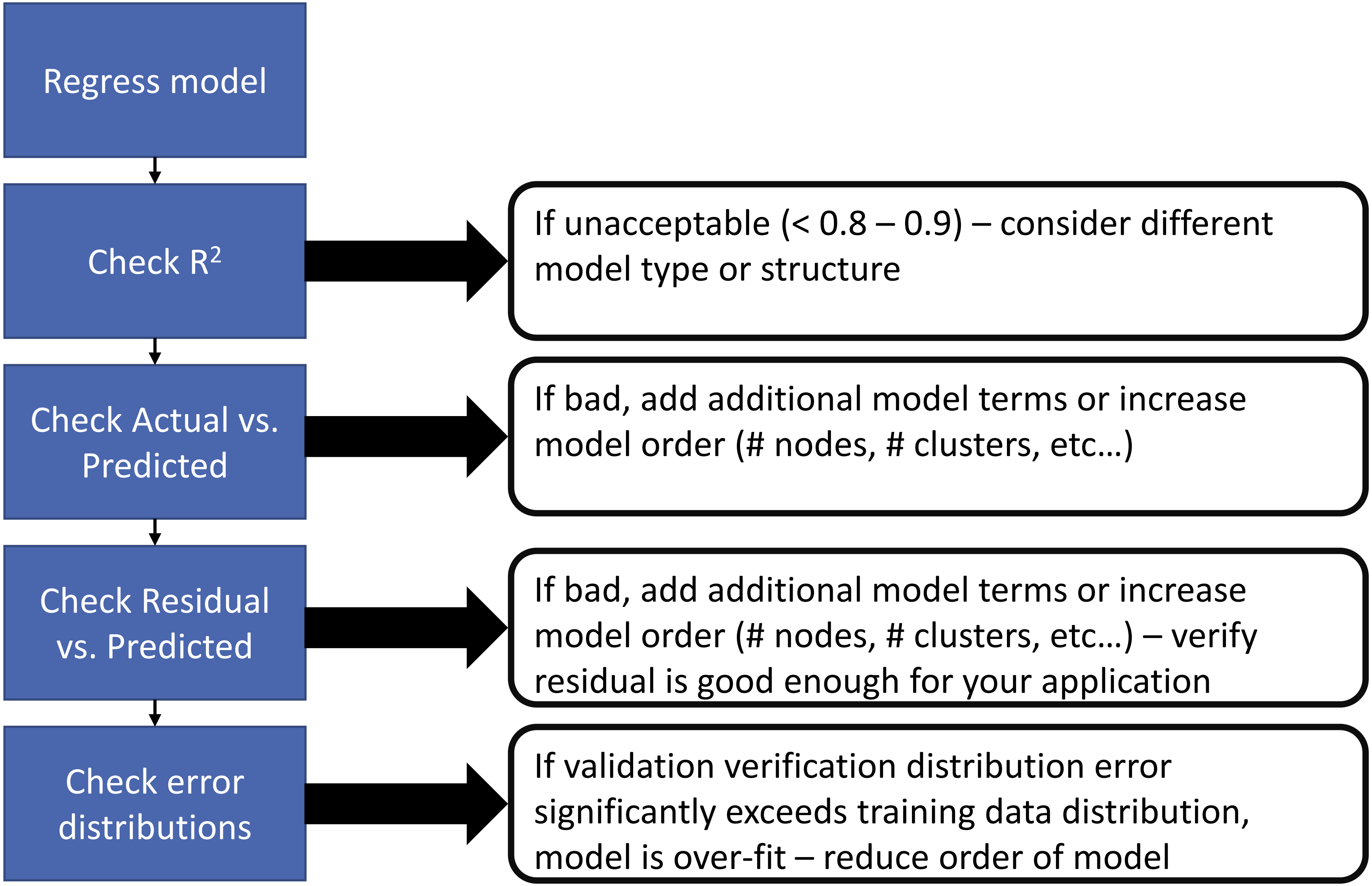
Training vs. Validation Data

- Recommend splitting cleaned training data into two regions
  - Base training and verification set:
    - Training data: 75%
    - Verification data: 25%
  - Validation set:
    - Should contain full coverage over training region and small scale extrapolation if available

Training and verification data should be randomly chosen from region with full coverage – Ensures you do not bias model on any inputs or unmeasured parameters



Process for Checking Model Quality



Regress model

Check R^2

If unacceptable (< 0.8 – 0.9) – consider different model type or structure

Check Actual vs. Predicted

If bad, add additional model terms or increase model order (# nodes, # clusters, etc…)

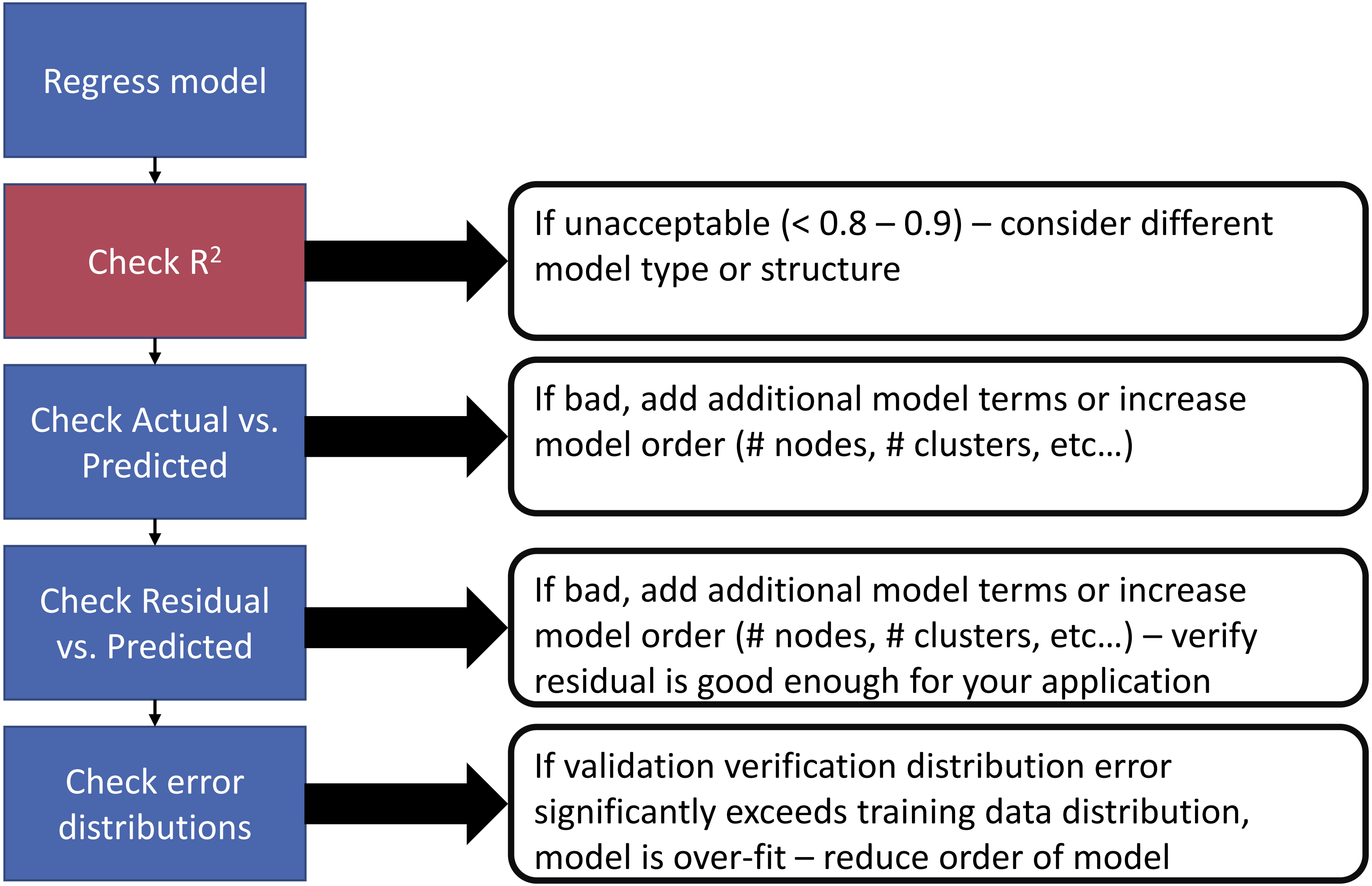
Check Residual vs. Predicted

If bad, add additional model terms or increase model order (# nodes, # clusters, etc…) – verify residual is good enough for your application

Check error distributions

If validation verification distribution error significantly exceeds training data distribution, model is over-fit – reduce order of model

Process for Checking Model Quality



Regress model

Check R^2

If unacceptable (< 0.8 – 0.9) – consider different model type or structure

Check Actual vs. Predicted

If bad, add additional model terms or increase model order (# nodes, # clusters, etc…)

Check Residual vs. Predicted

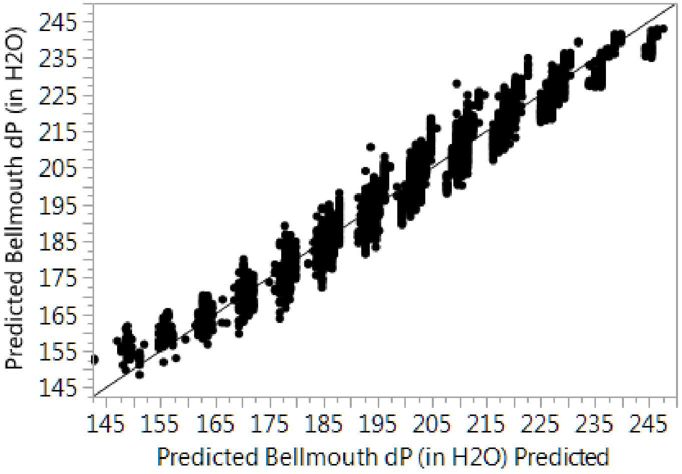
If bad, add additional model terms or increase model order (# nodes, # clusters, etc…) – verify residual is good enough for your application

Check error distributions

If validation verification distribution error significantly exceeds training data distribution, model is over-fit – reduce order of model

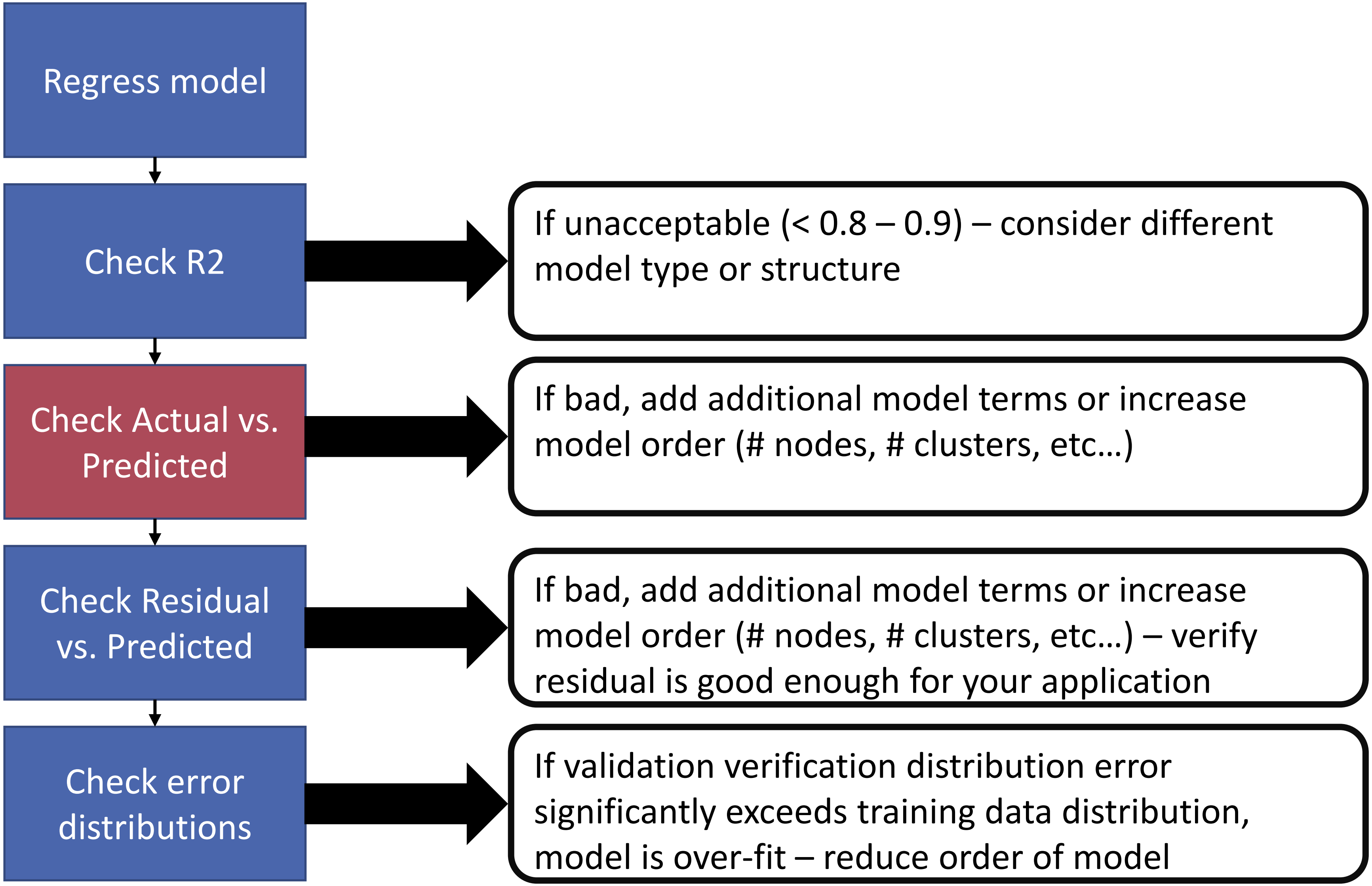
Model Fitting – Evaluating Quality

- R^2: Proportion of the variance in the dependent variable that is predictable from the independent variables
    - $R^2 = 1 - \frac{SS_{residuals}}{SS_{total}}$
    - $SS_{total} = \sum_i (x_i - \bar{x})^2$
    - $SS_{residuals} = \sum_i (x_i - y_i)^2$
- A good initial screening tool
    - Low values (<0.8 to 0.9) indicate poor accuracy
    - High values **do not** indicate a good model
- Acceptable values tell you that functional form of the model you have chosen is acceptable
    - Type of model (neural, clustering)
    - Model parameters (degrees of freedom, number of nodes)
- Does not evaluate predictive capability of model!



R^2 = 0.95!

Process for Checking Model Quality



Regress model

Check R2

If unacceptable (< 0.8 – 0.9) – consider different model type or structure

Check Actual vs. Predicted

If bad, add additional model terms or increase model order (# nodes, # clusters, etc…)

Check Residual vs. Predicted

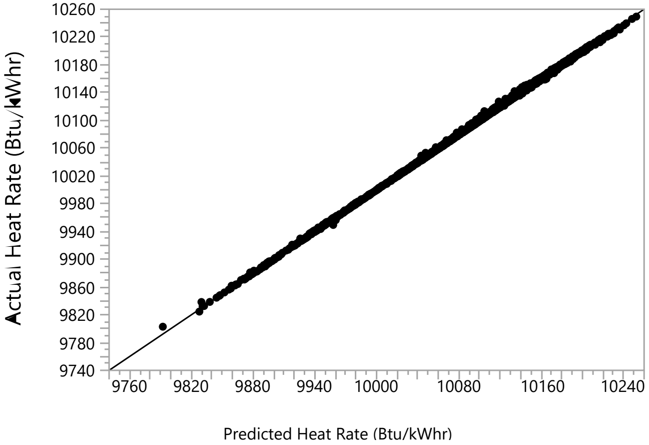
If bad, add additional model terms or increase model order (# nodes, # clusters, etc…) – verify residual is good enough for your application

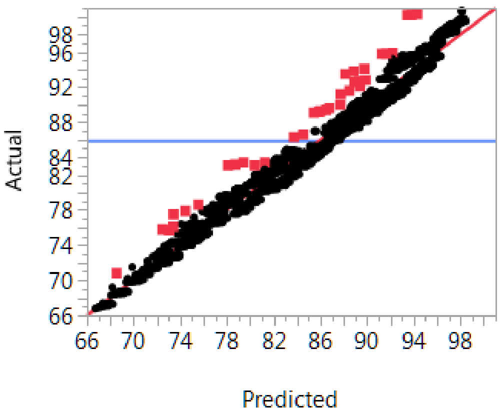
Check error distributions

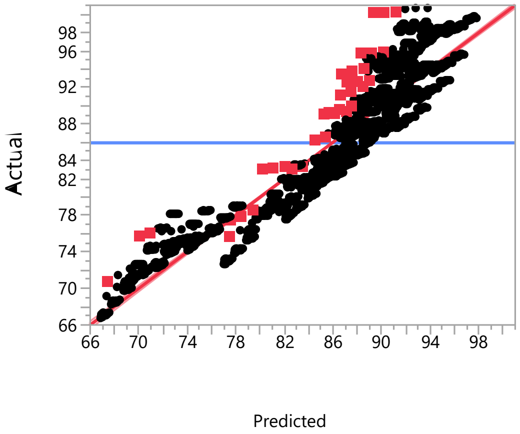
If validation verification distribution error significantly exceeds training data distribution, model is over-fit – reduce order of model

Model Fitting – Actual vs. Predicted

- Cross plot of training data vs. model prediction for same inputs







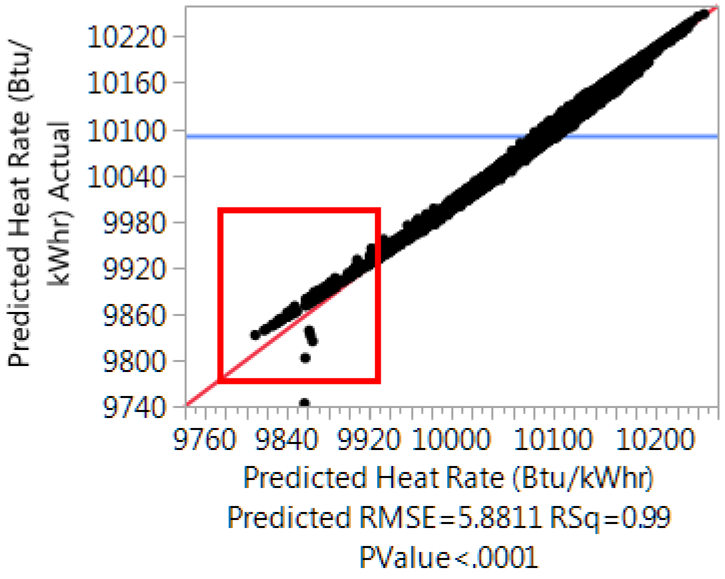
Greatest Fit Ever!

Not bad:
Might want to check outliers

Unacceptable
(R2 = 0.88)

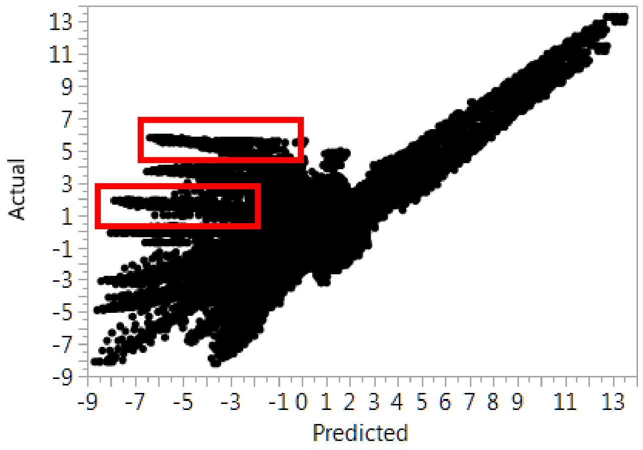
Model Fitting – Actual vs. Predicted Diagnostic Plots

Curvature could indicate higher order model needed



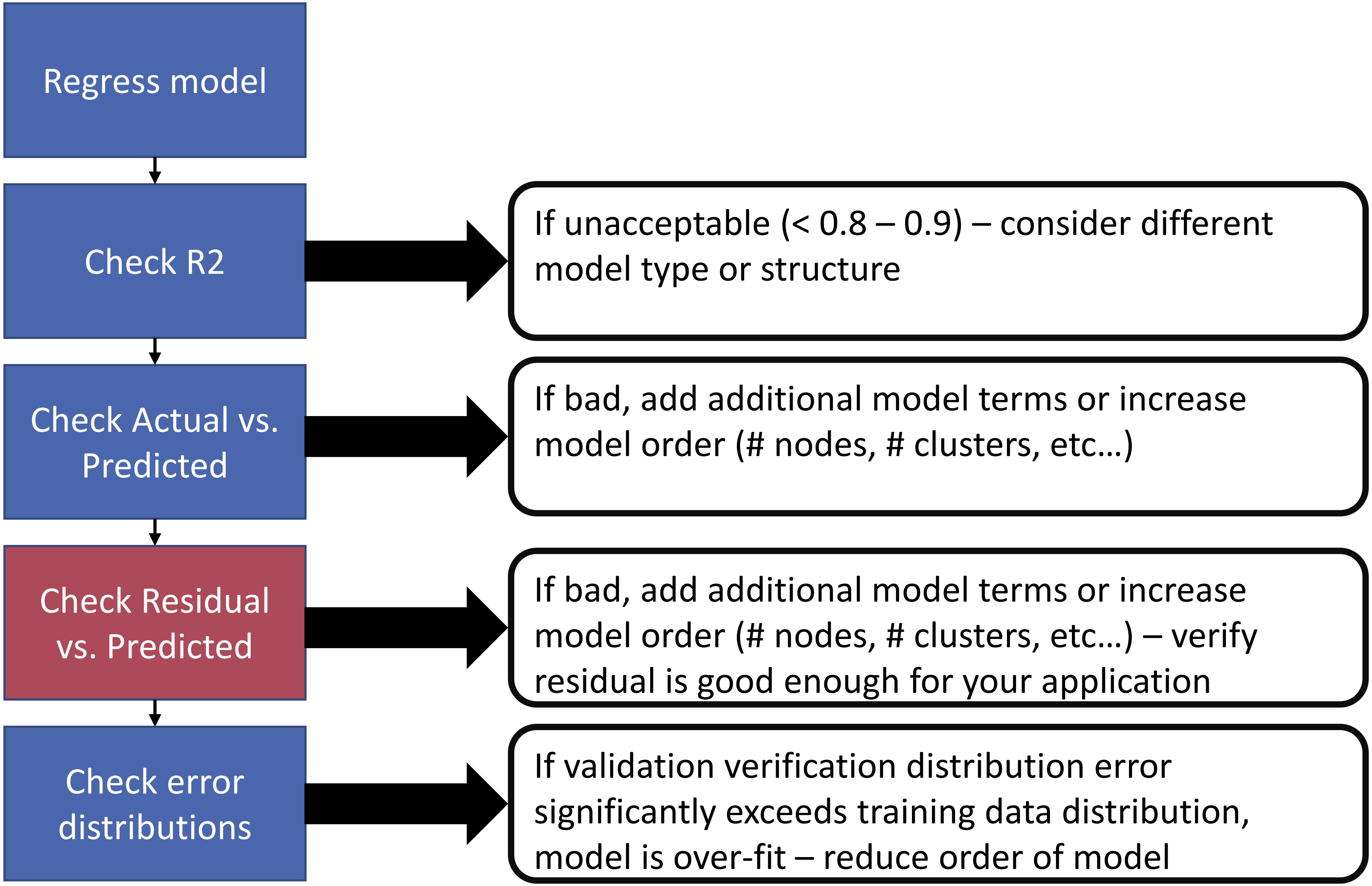
Increase degree, # of nodes or clusters

Banding or multiple series could indicate important input was neglected



Check for missing correlating parameter or 'clumped' input data

Process for Checking Model Quality



Regress model

Check R2

If unacceptable (< 0.8 – 0.9) – consider different model type or structure

Check Actual vs. Predicted

If bad, add additional model terms or increase model order (# nodes, # clusters, etc…)

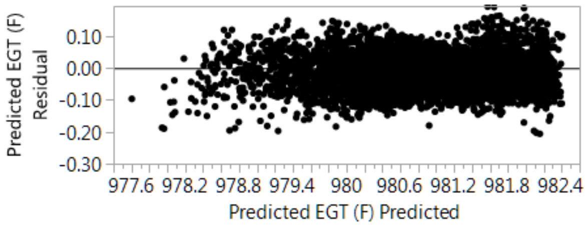
Check Residual vs. Predicted

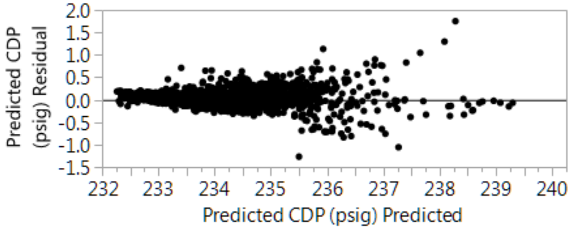
If bad, add additional model terms or increase model order (# nodes, # clusters, etc…) – verify residual is good enough for your application

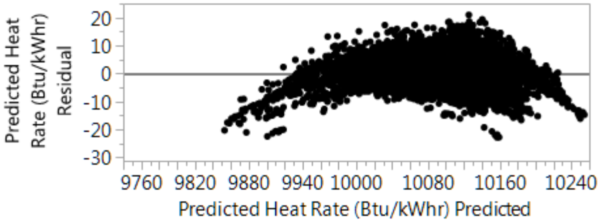
Check error distributions

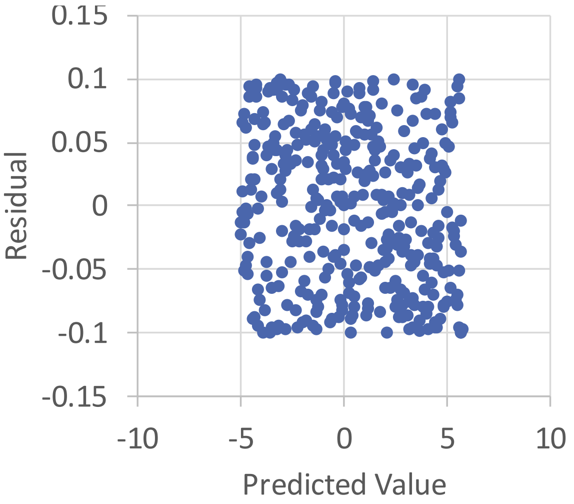
If validation verification distribution error significantly exceeds training data distribution, model is over-fit – reduce order of model

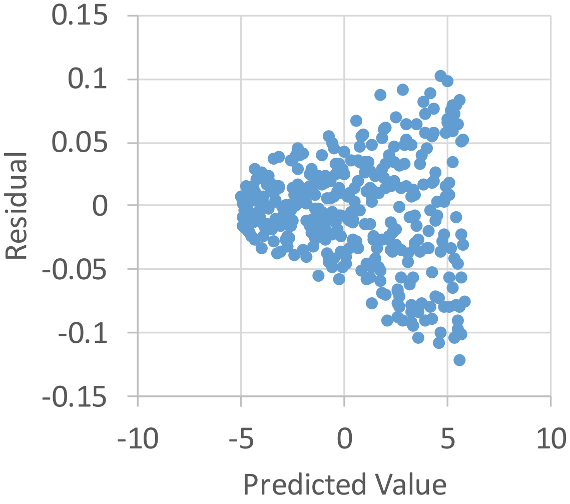
Model Fitting – Residuals vs. Predicted

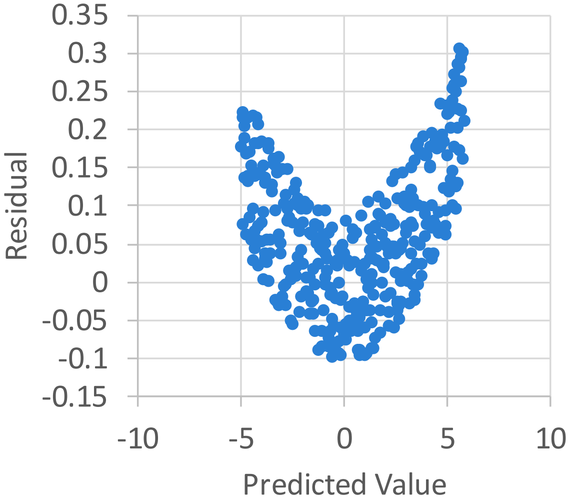










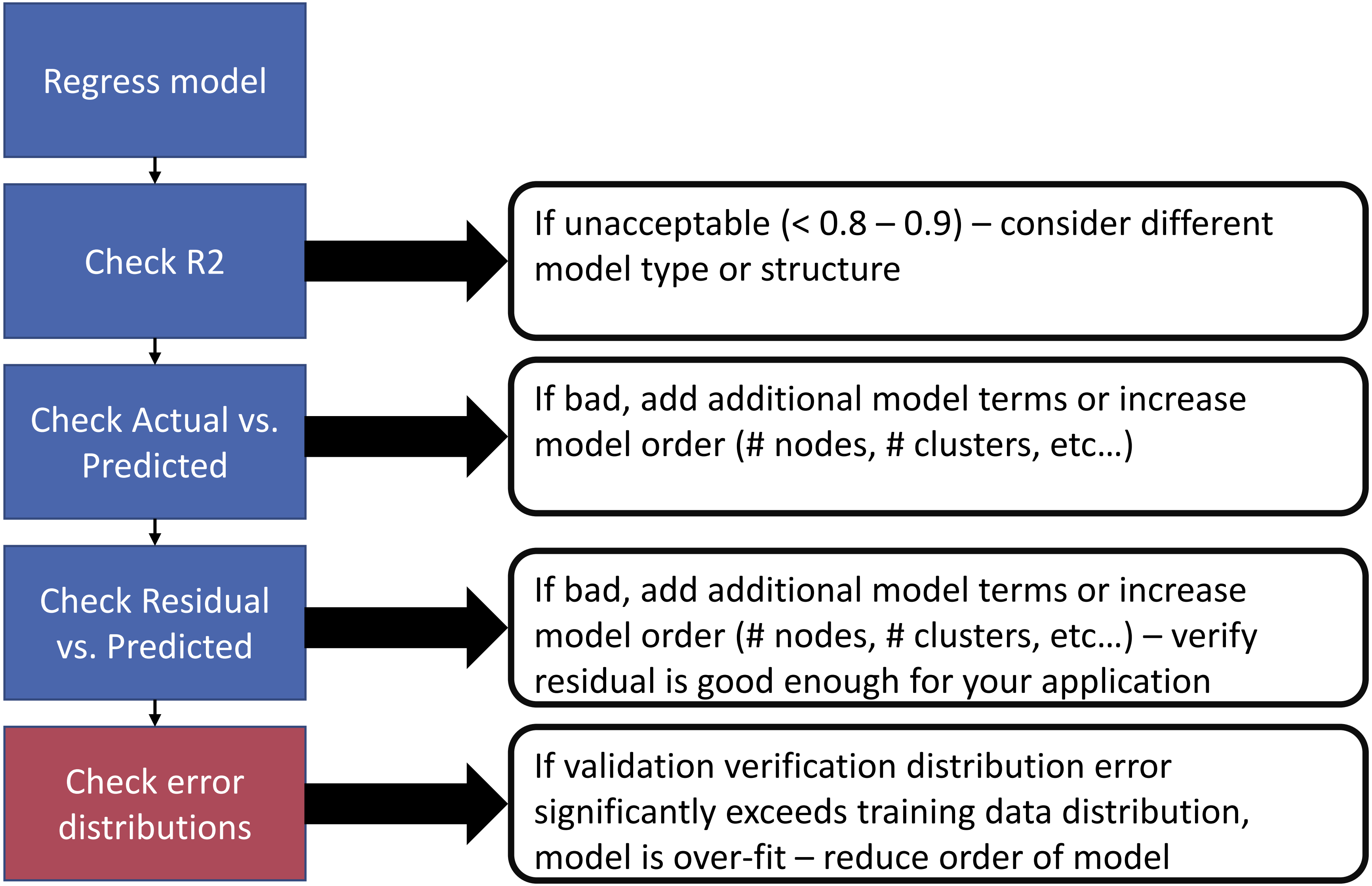


Good!
Shows random spread

Indicates one variable driving response or missing effects
Check magnitude
(maybe you do not care)

Unacceptable
Model should be higher order

Process for Checking Model Quality



Regress model

Check R2

If unacceptable (< 0.8 – 0.9) – consider different model type or structure

Check Actual vs. Predicted

If bad, add additional model terms or increase model order (# nodes, # clusters, etc…)

Check Residual vs. Predicted

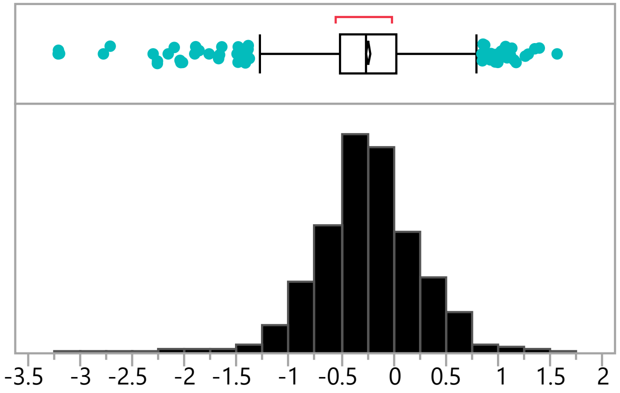
If bad, add additional model terms or increase model order (# nodes, # clusters, etc…) – verify residual is good enough for your application

Check error distributions

If validation verification distribution error significantly exceeds training data distribution, model is over-fit – reduce order of model

Model Fitting – Error Distributions

- Can calculate as percent error or residual
- Useful for two diagnostics
  - Error is normally distributed
  - Model is not over-fit
- Should be centered around zero and normally distributed



Percent Error or Absolute Residual

BREAK TIME!

Additional Statistics for Categorical Models

Confusion Matrix

- Provides quick scan of accuracy of discrete predictions
- Essentially a discrete version of the actual vs. predicted plot
- Suitable for categorical or ordinal data
  - Categorical = red, green, blue
  - Ordinal = 1, 2, 3, 4 or first, second, third

		Actual		
		Red	Green	Blue
Predicted	Red	45	4	3
	Green	1	72	0
	Blue	3	2	55

Want zeroes off-diagonal – indicates good predictive capability

Make sure to examine for training and validation data sets!

This form provides good quick visual – is there a better way to examine?

Confusion Matrix – Other views

- Constructs a table of confusion for each category
- Is basis for constructing graphical diagnostic (next slide)

		Actual		
		Red	Green	Blue
Predicted	Red	45	4	3
	Green	1	72	0
	Blue	3	2	55

		Actual	
		Category	Not-Category
Predicted	Category	True Positive	False Positive
	Not-Category	False Negative	True Negative

		Actual	
		Red	Not Red
Predicted	Red	**45**	
	Not Red		

Confusion Matrix – Other views

- Constructs a table of confusion for each category
- Is basis for constructing graphical diagnostic (next slide)

		Actual		
		Red	Green	Blue
Predicted	Red	45	4	3
	Green	1	72	0
	Blue	3	2	55

		Actual	
		Category	Not-Category
Predicted	Category	True Positive	False Positive
	Not-Category	False Negative	True Negative

		Actual	
		Red	Not Red
Predicted	Red	**45**	7
	Not Red		

Confusion Matrix – Other views

- Constructs a table of confusion for each category
- Is basis for constructing graphical diagnostic (next slide)

		Actual		
		Red	Green	Blue
Predicted	Red	45	4	3
	Green	1	72	0
	Blue	3	2	55

		Actual	
		Category	Not-Category
Predicted	Category	True Positive	False Positive
	Not-Category	False Negative	True Negative

		Actual	
		Red	Not Red
Predicted	Red	**45**	**7**
	Not Red	**4**	

Confusion Matrix – Other views

- Constructs a table of confusion for each category
- Is basis for constructing graphical diagnostic (next slide)

		Actual		
		Red	Green	Blue
Predicted	Red	45	4	3
	Green	1	72	0
	Blue	3	2	55

		Actual	
		Category	Not-Category
Predicted	Category	True Positive	False Positive
	Not-Category	False Negative	True Negative

		Actual	
		Red	Not Red
Predicted	Red	**45**	**7**
	Not Red	**4**	**129**

Receiver Operating Characteristic (ROC) Curve

- Plots true positive rate against false positive rate
- Neural networks actually predict probability of classification – lead to multiple tables – can be used to generate curve

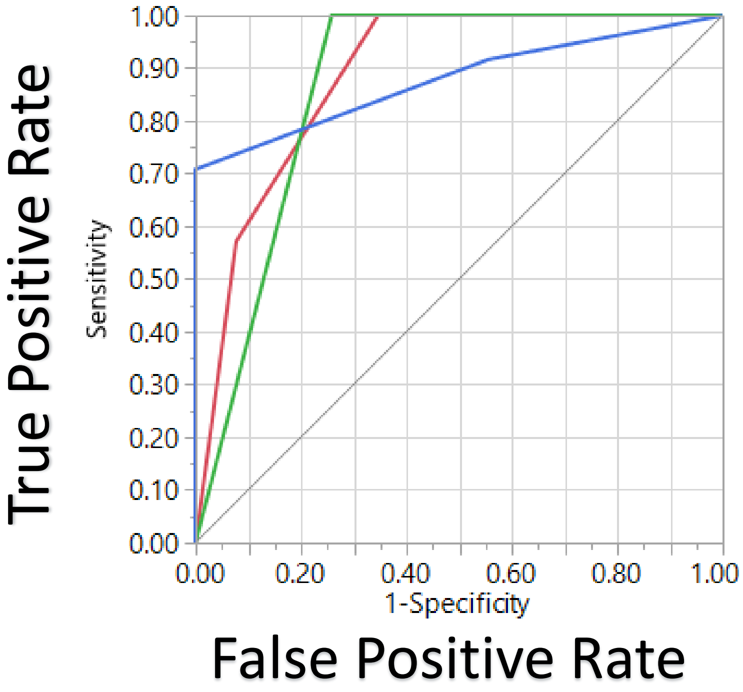
		Actual	
		Red	Not Red
Predicted	Red	**45**	**7**
	Not Red	**4**	**129**
		$\sum = \mathbf{49}$	$\sum = \mathbf{136}$

$$TRUE\ POSITIVE\ RATE\ (TPR) = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{45}{49} = 0.918$$

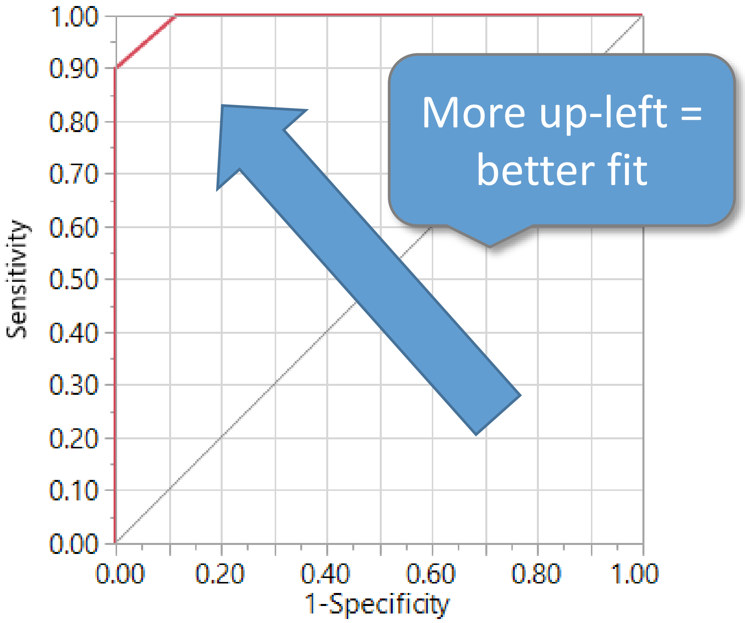
$$FALSE\ POSITIVE\ RATE\ (FPR) = \frac{False\ Positive}{True\ Negative + False\ Positive} = \frac{7}{129} = 0.054$$

Receiver Operating Characteristic (ROC) Curve

Receiver Operating Characteristic – Decision Tree



Receiver Operating Characteristic – Logistic Regression



More up-left = better fit

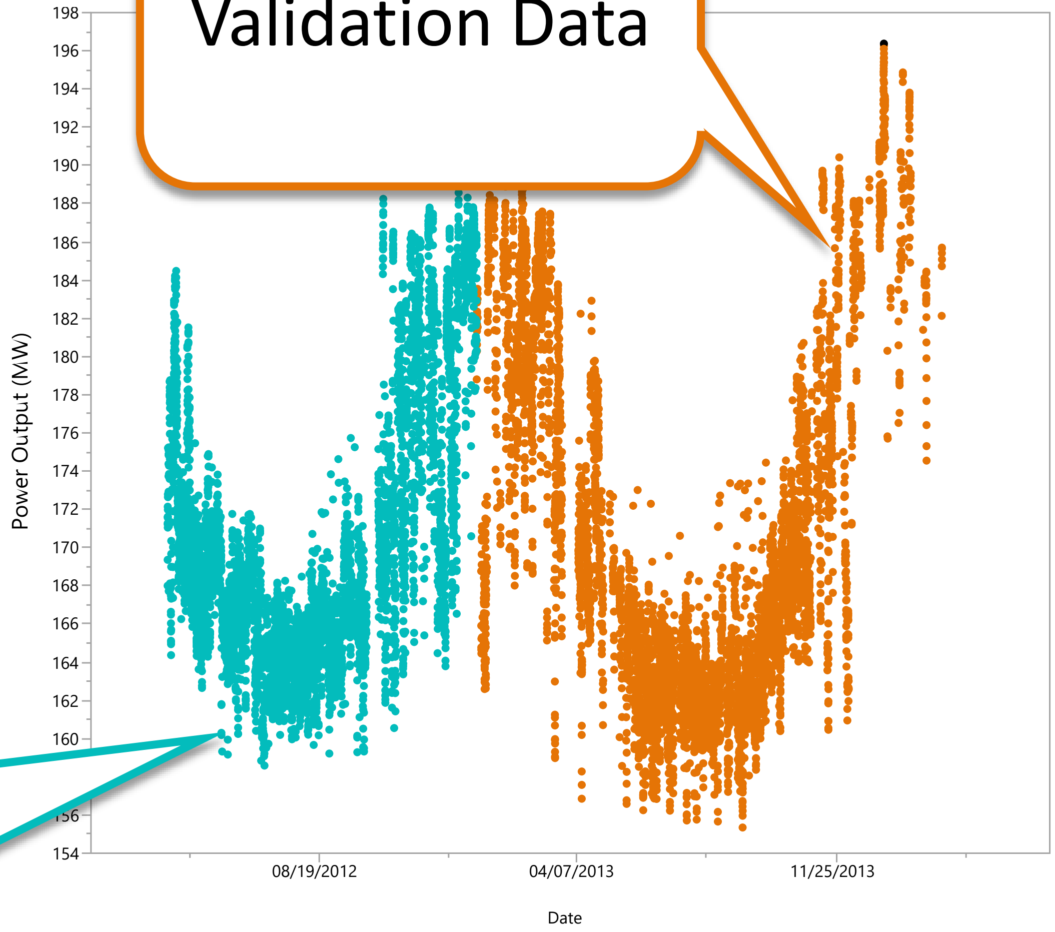
Use Cases

Example use Cases

- Performance Examples – Which model is better at predicting expected power?
  - Neural Network of Performance
  - Clustering (k-means) model of performance
- Neural Network Classifier
  - Can we predict when steam injection is running ? (using prior example)

Fitting Neural Network to Performance Data – Use Case

- Attempt to use one year of data to predict the next year's power output?
- Let' use a neural network



Validation Data

Training / Validation Data

Neural Network – Step 1 – Select Input List

- For gas turbine, typical inputs list
  - In order of importance for power
  - List or ordering may change for different metrics
- Input list for this use case
  - Compressor Inlet Temperature
  - Compressor Inlet Pressure Drop
  - Exhaust Pressure Drop
  - Barometric Pressure
  - Natural Gas (or fuel) Temperature
  - Relative Humidity
- Six inputs – one output – let’s try it!

Neural Network Use Case – Selecting the Structure

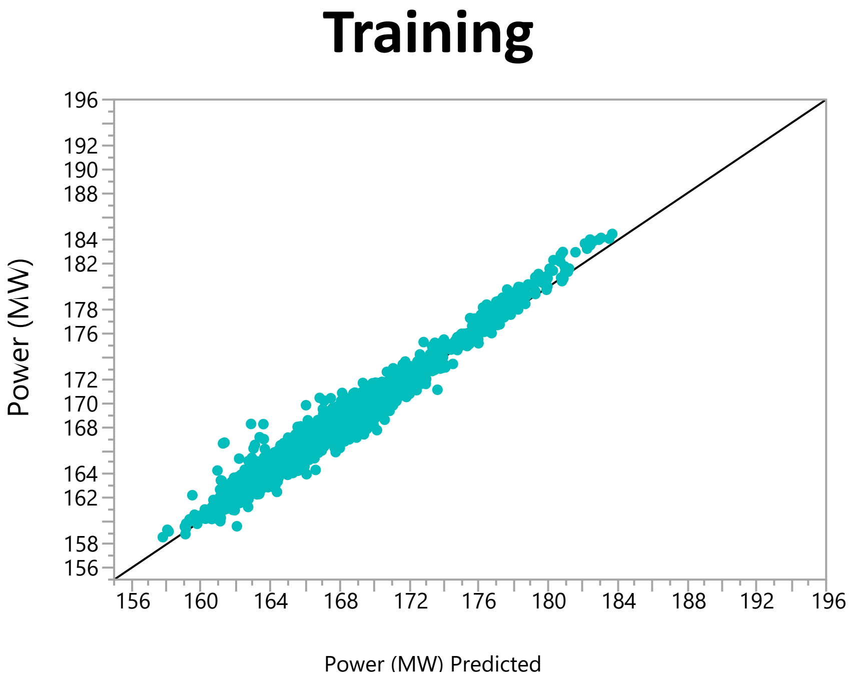
- 6 inputs, 1 output
- Try one hidden layer with 6 nodes first
- Use TanH activation function

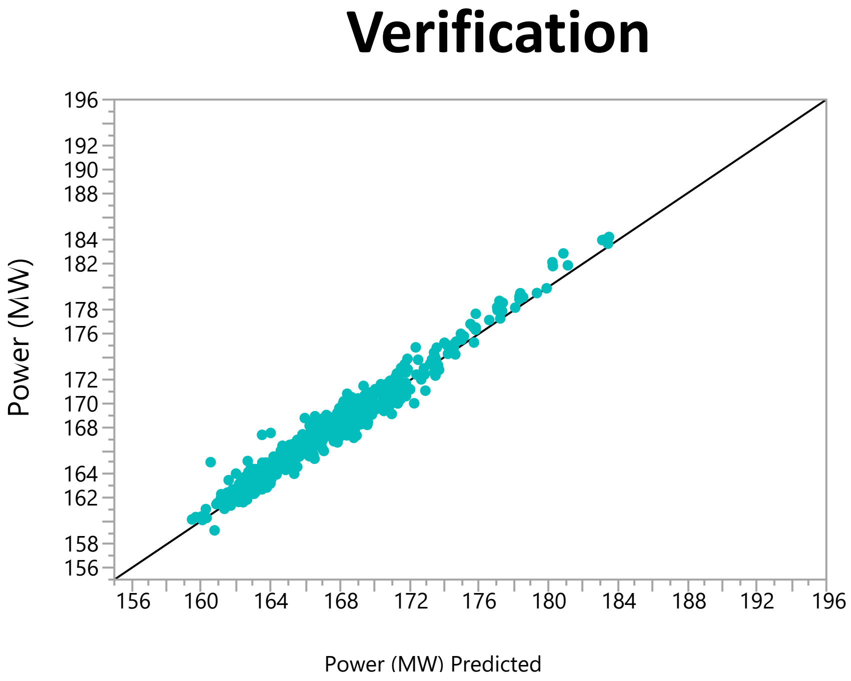
Node Type	# of nodes
Input	Defined by problem (X’s)
Hidden 1 (closer to inputs)	~ number of outputs * (number of inputs)
Hidden 2 (closer to outputs)	0 < Number of outputs < number of inputs
Output	Two options: 1. Fit one neural network per output (Y) a) Easier to fit b) Simplifies network structure 2. Fit multiple outputs a) Enables coupling to be observed between Y1 and Y2 b) Often requires additional hidden nodes

Neural Network Use Case (6 node single layer)

- Check Diagnostics
  - R^2 Training = 0.987
  - R^2 Verification = 0.988

- Actual by predicted plots:

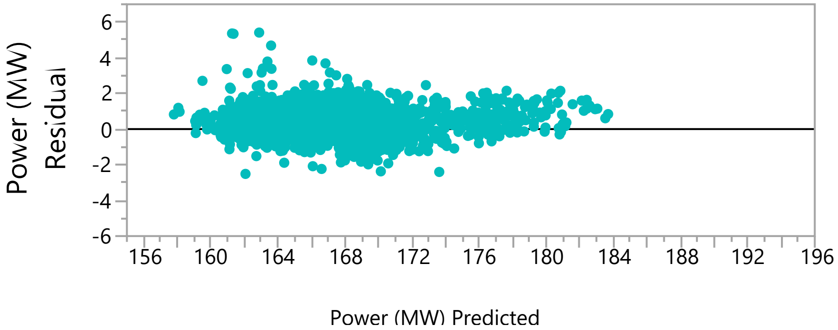




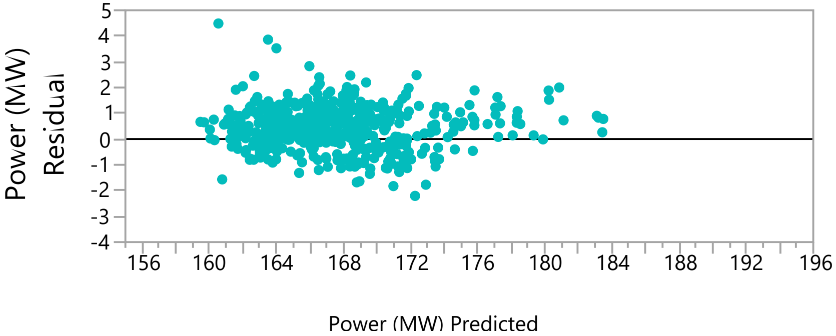
Neural Network Use Case (6 node single layer)

- Residual by predicted plots:

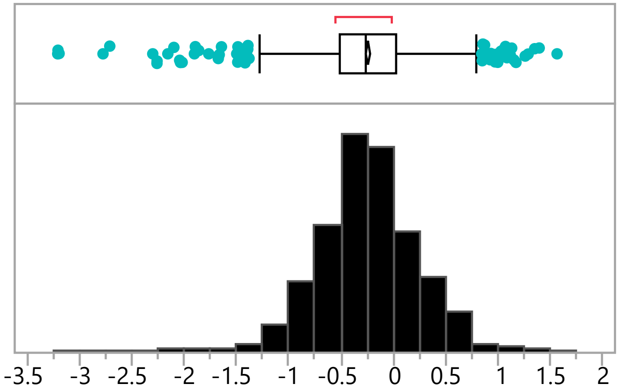
Training



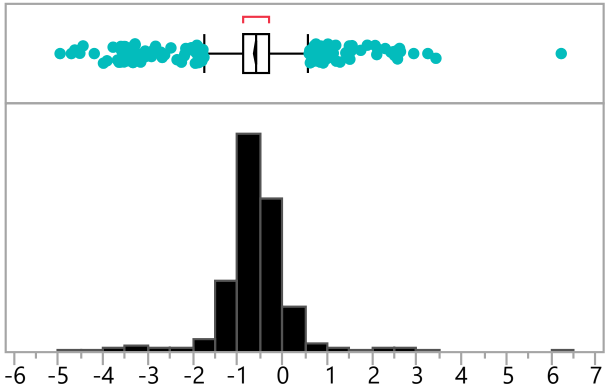
Verification



- Model error distributions (% error)



$mean = -0.25\%, std.dev = 0.47\%$

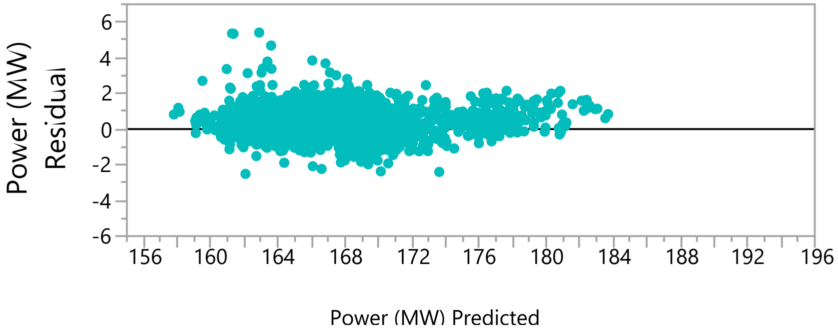


$mean = -0.59\%, std.dev = 0.67\%$

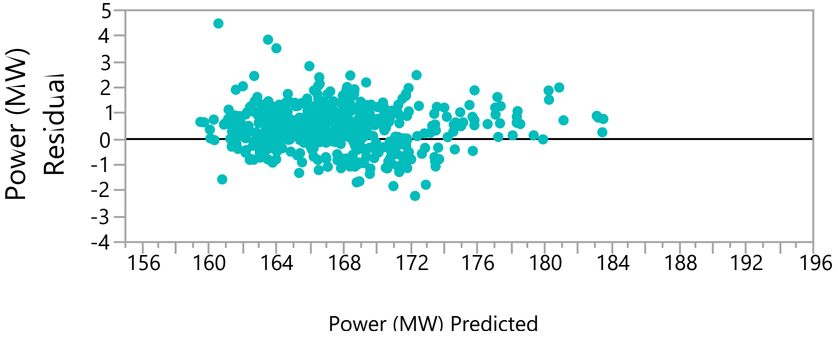
Neural Network Use Case (6 node single layer)

- Residual by predicted plots:

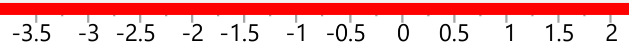
Training



Verification



Non centered mean with increase in error for verification could indicate sub-par fit

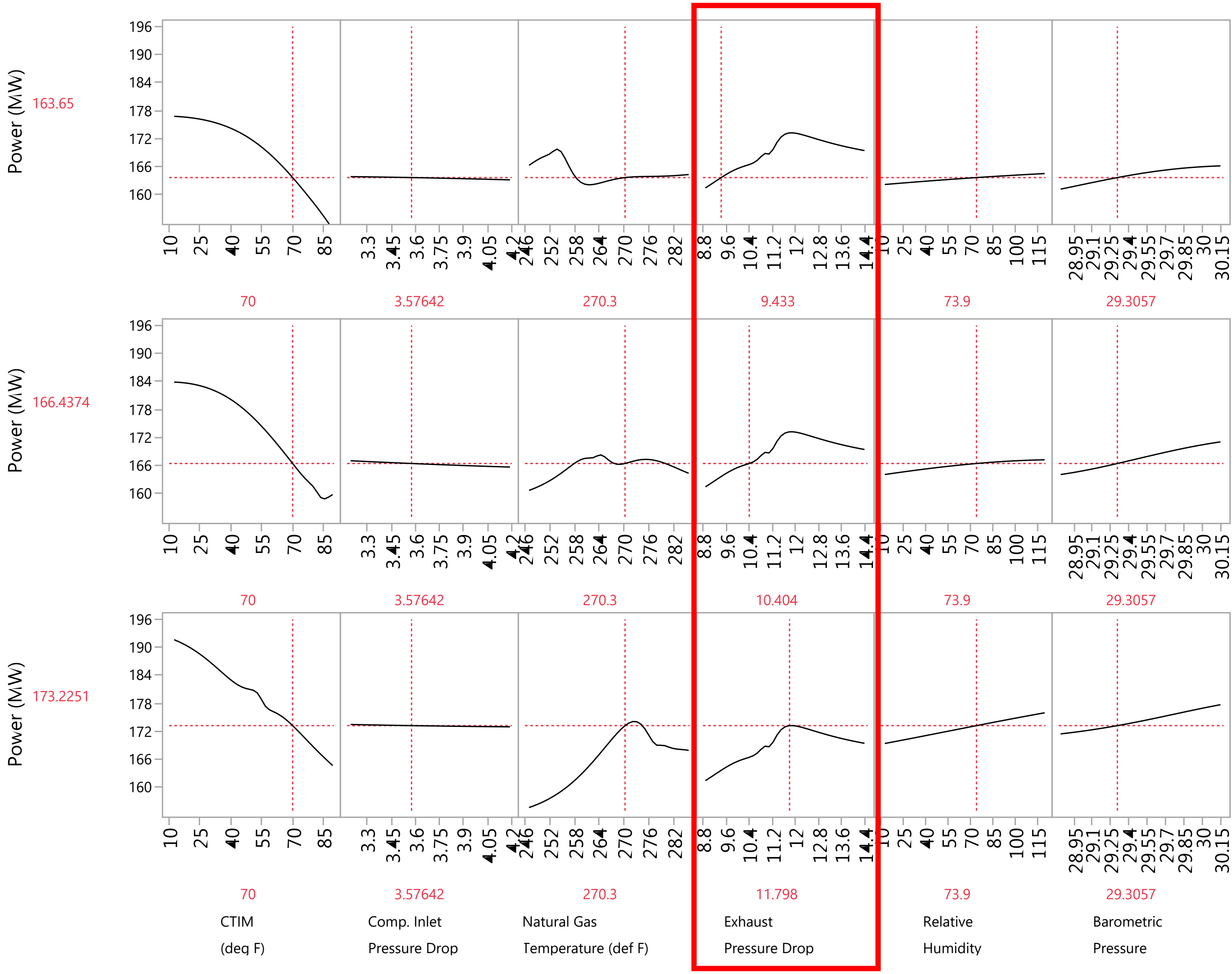


$mean = \mathbf{-0.25\%}, std.dev = 0.47\%$ $mean = \mathbf{-0.59\%}, std.dev = 0.67\%$

Neural Network Use Case (6 node single layer)

- In addition to standard diagnostic set – should check shape of regression
- Plot partial derivatives of each input against each output
- Generate by holding other parameters constant

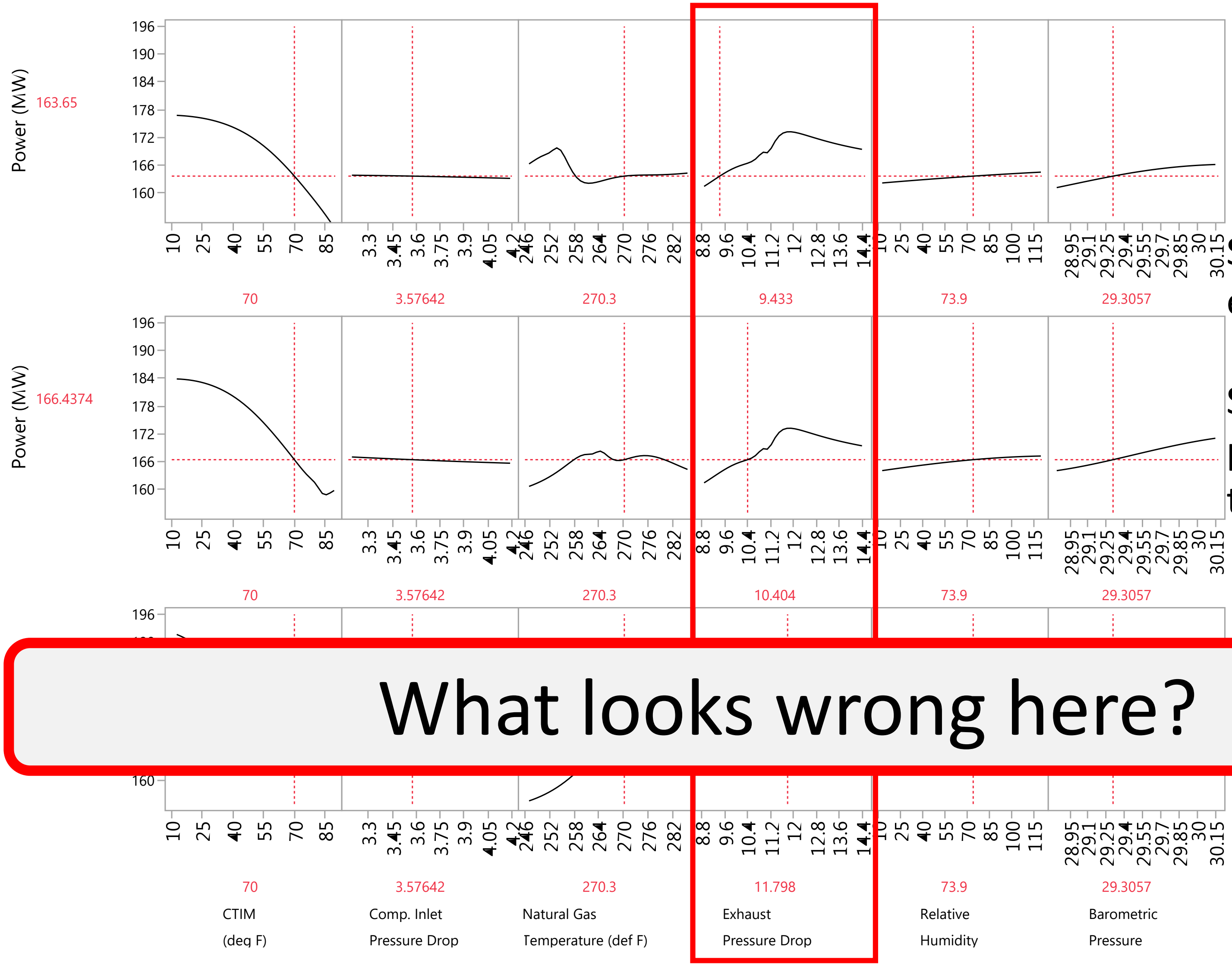
Neural Network – 6 nodes – generating partials



Shows variation with exhaust pressure drop

Should check all parameters to be thorough

Neural Network – 6 nodes – generating partials



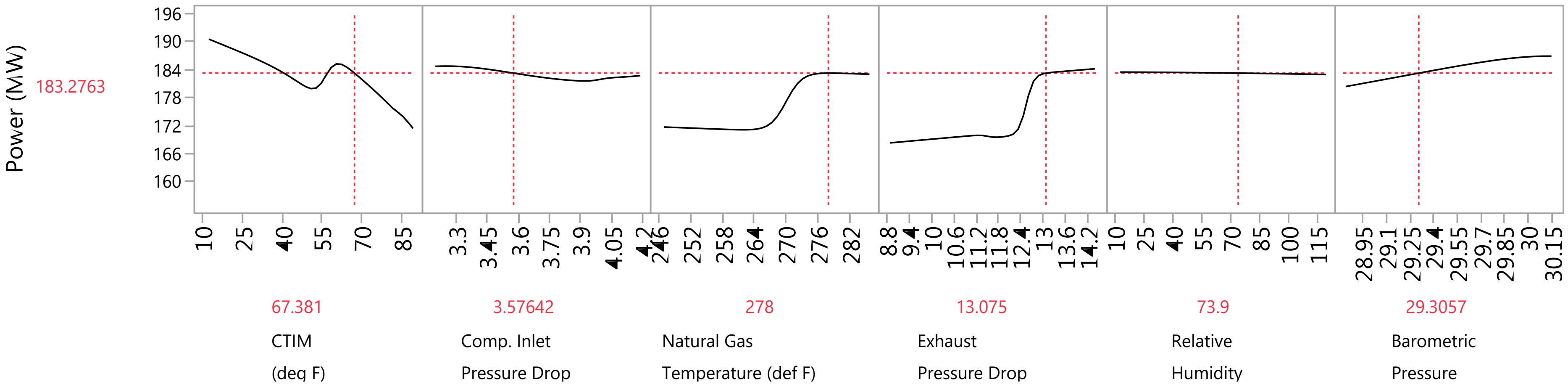
What looks wrong here?

Shows variation with exhaust pressure drop

Should check all parameters to be thorough

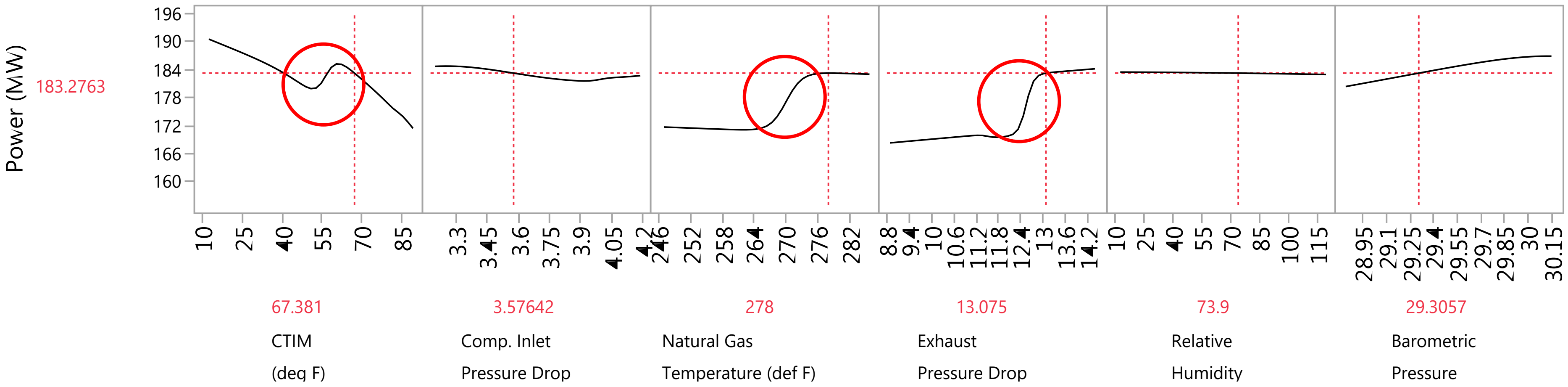
Try Additional Nodes

- Try two layer network with 6 inputs and 6 outputs
- Regular statistics show good results
- What about partial derivatives?



Trying Additional Nodes

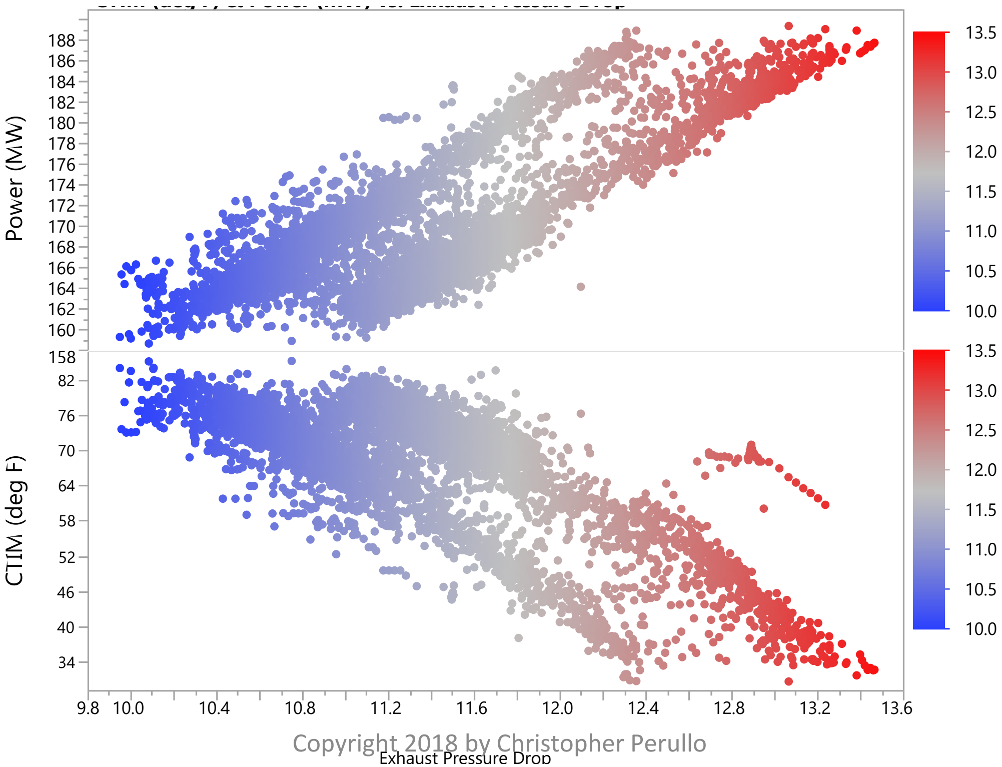
- Try two layer network with 6 inputs and 6 outputs
- Regular statistics show good results
- What about partial derivatives?



Need to examine data to understand what causes switching behavior

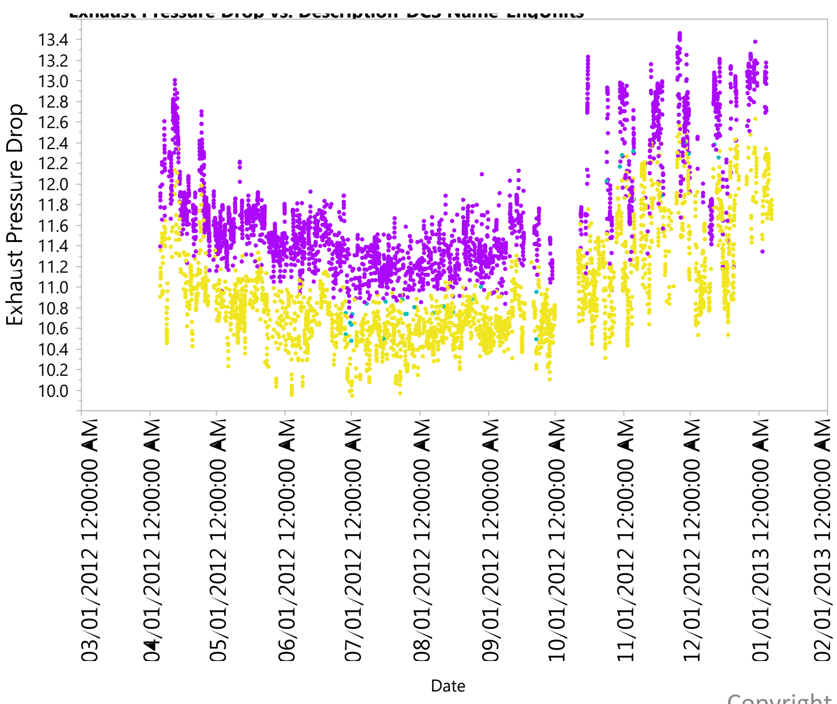
Diagnosing Strange Behavior

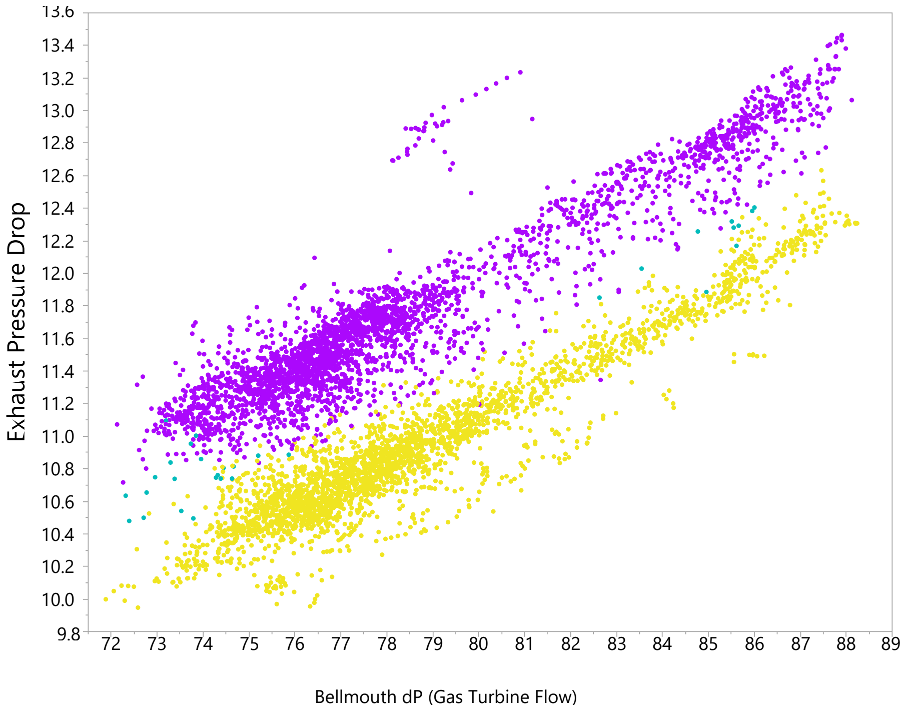
- Cross plot inputs to look for trends
- Clearly two discrete power vs. exhaust pressure drop curves
- Could it be discrete event?
  - Cross plot gas turbine parameters vs. time
  - Color the two regions to quickly identify separation



Diagnosing Strange Behavior

No obvious correlation with time or any gas turbine parameters

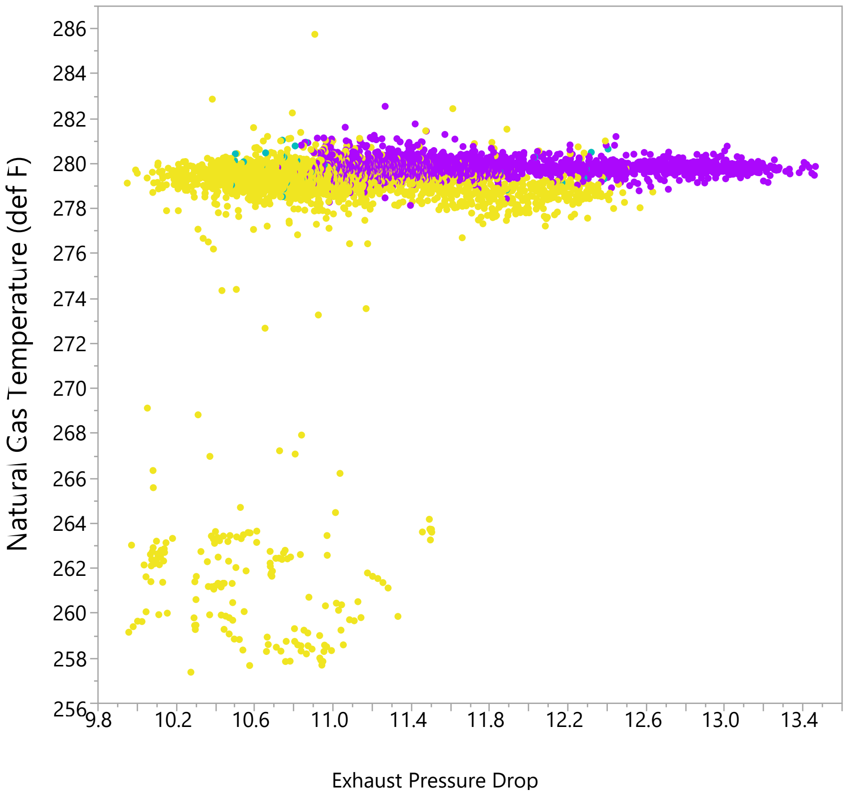




Exhaust pressure drop tracks with gas turbine flow

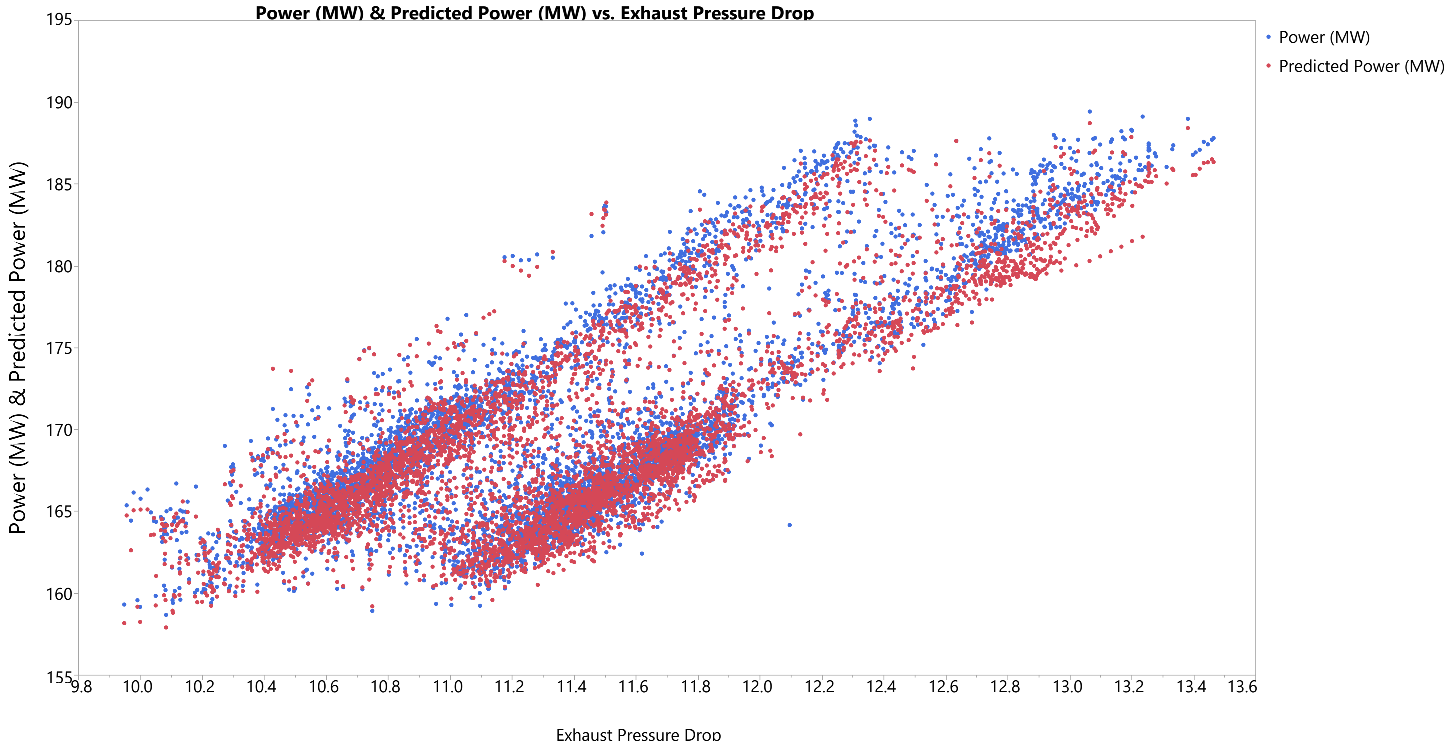
Diagnosing Strange Behavior

- Appears to be relationship between fuel gas temperature and exhaust pressure drop
- Neural network correctly captured this behavior



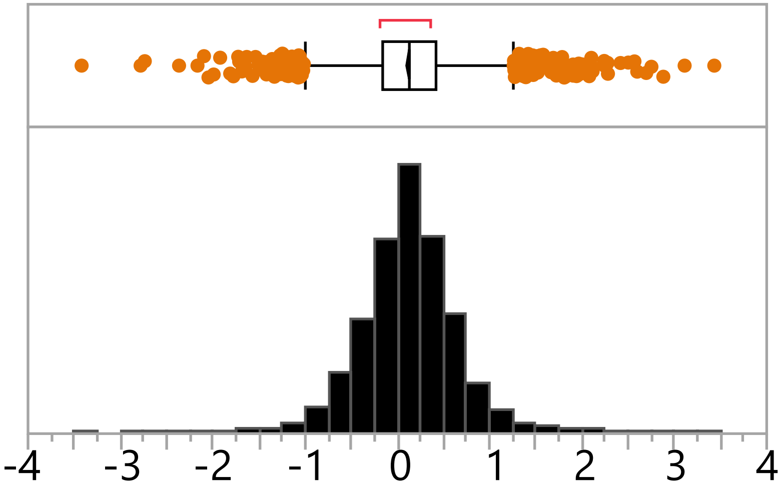
Next Steps

- A) Do you care about the physical reason?
  - Neural network appears to capture nonlinear variability
  - Check validation data set!
- B) Should track down physical reason and include additional inputs to model if necessary

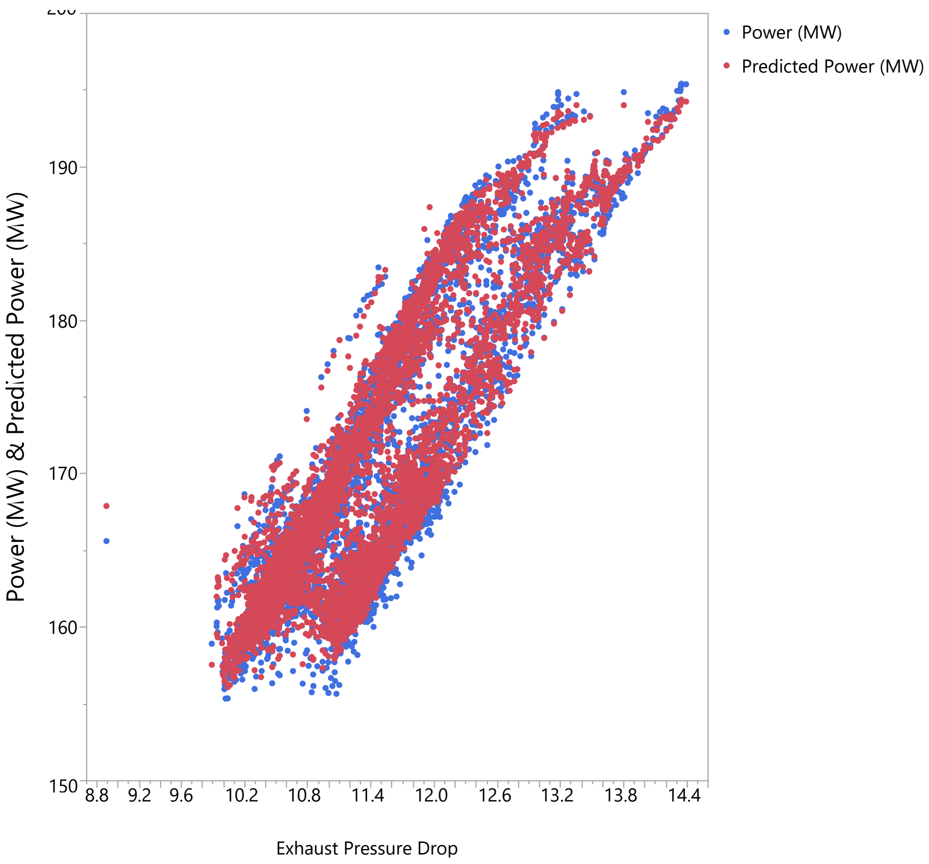


Neural Network – Checking Validation Data Set

- Trend still captured – error looks good!

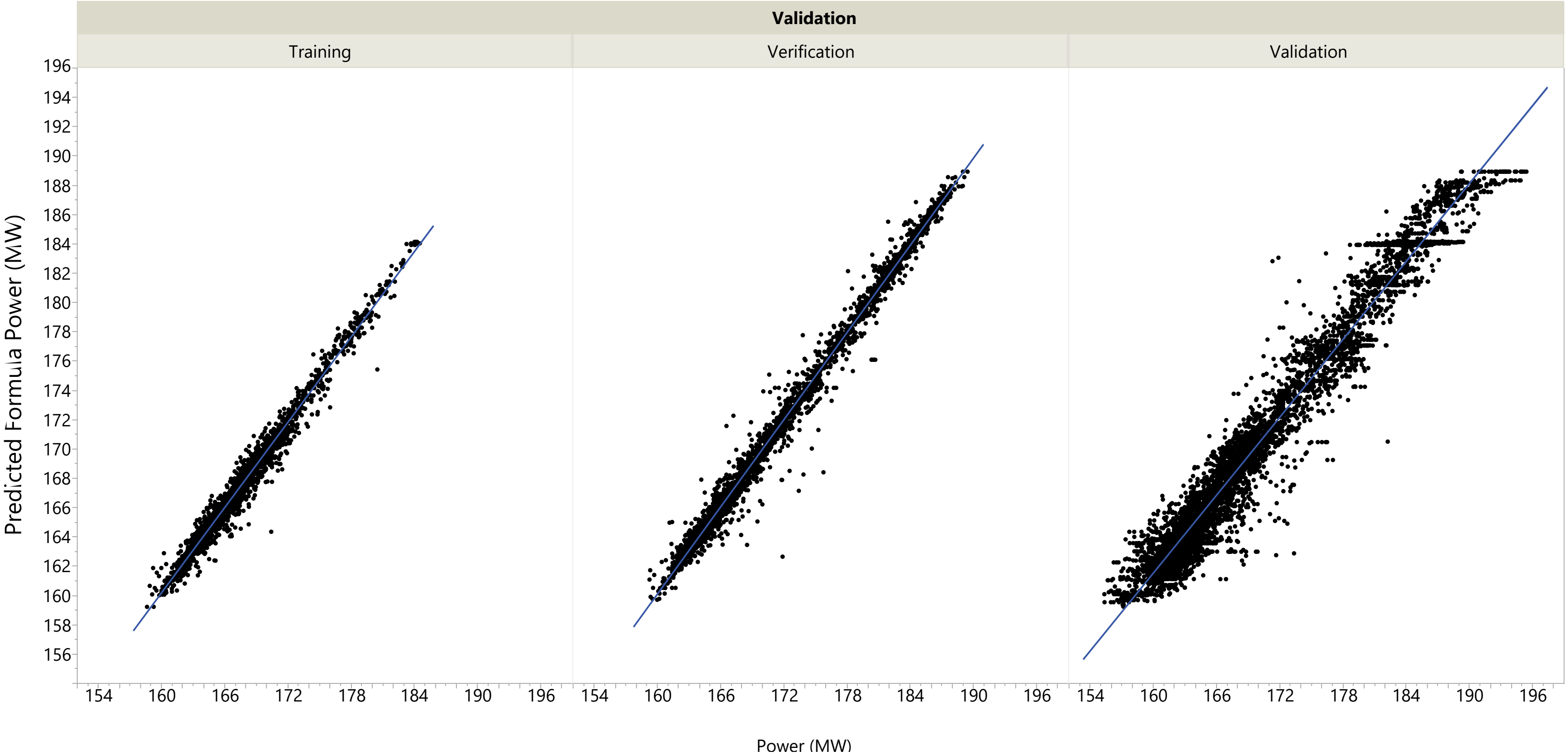


$$mean = 0.12\%, std.dev = 0.50\%$$



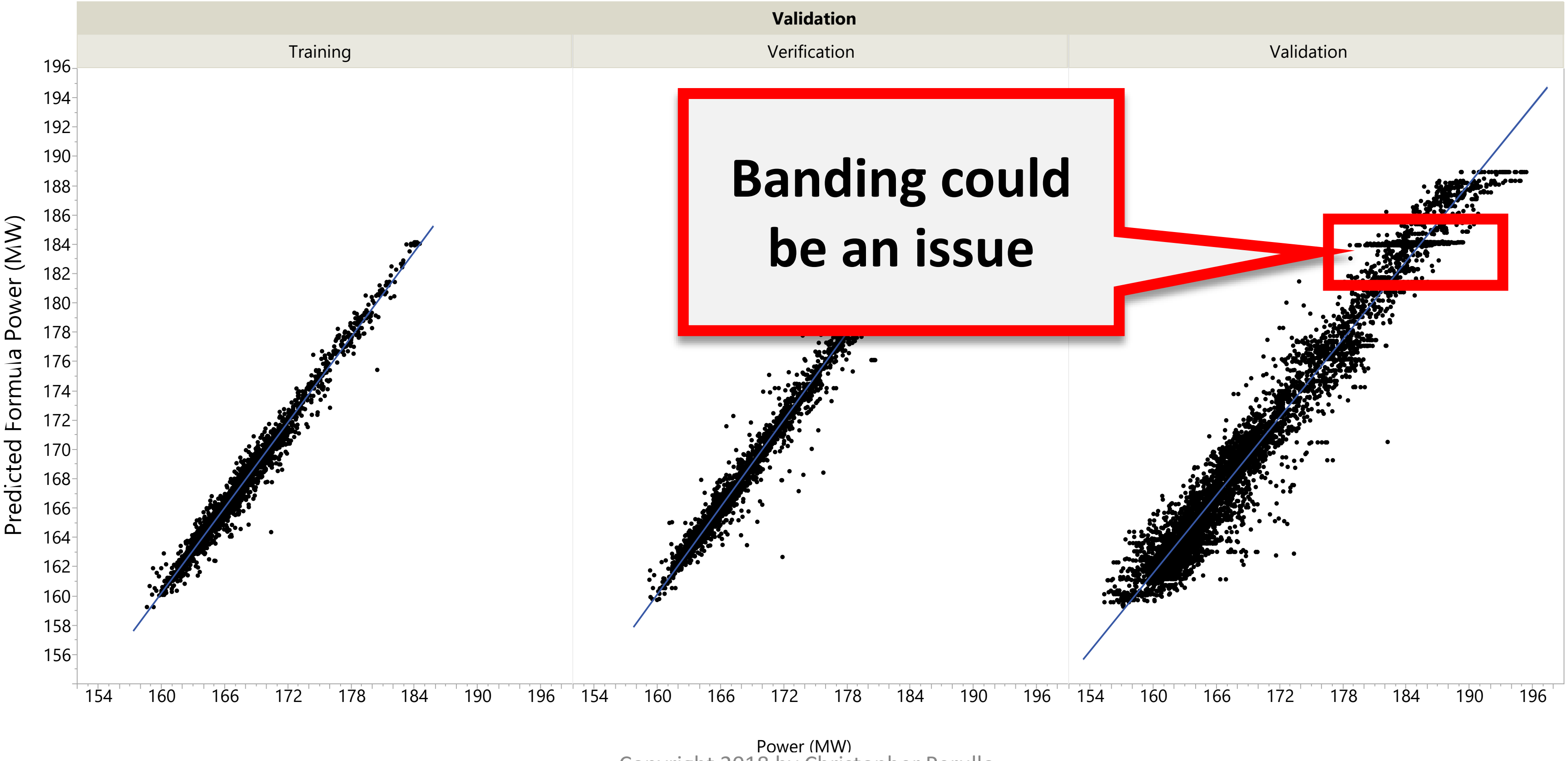
Attempting Clustering

- Use same data set as before
- K-Nearest Neighbors using 3 closest neighbors
- Actual by predicted shown below:



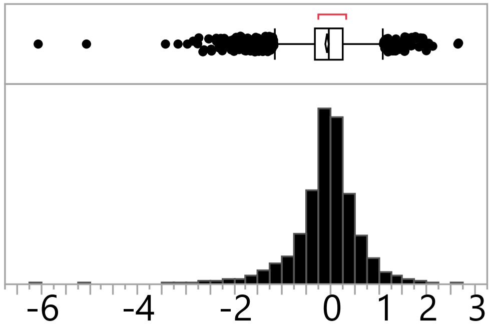
Clustering – Predicted vs. Actual

- Use same data set as before
- K-Nearest Neighbors using 3 closest neighbors
- Actual by predicted shown below:



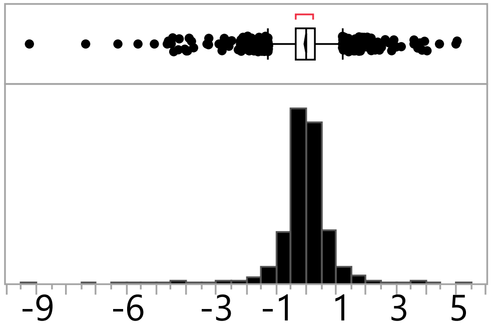
Banding could be an issue

Clustering - Residuals



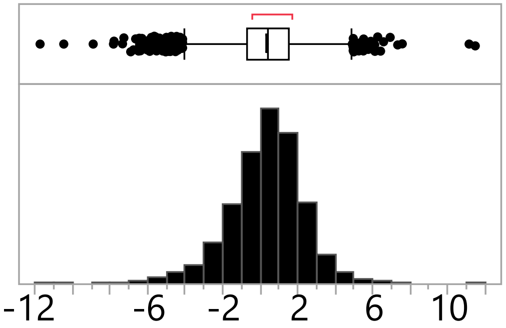
Training Data

$mean = -0.06, std.dev = 0.63$



Verification Data

$mean = -0.02, std.dev = 0.79$



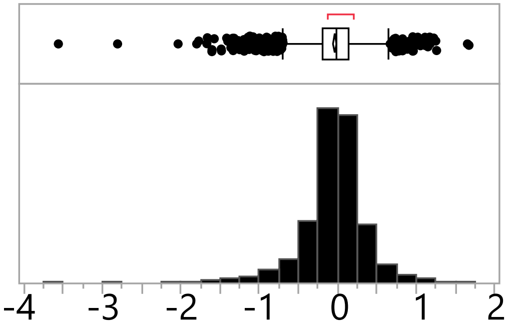
Validation Data

$mean = 0.33, std.dev = 1.92$

Clustering – Percent Error

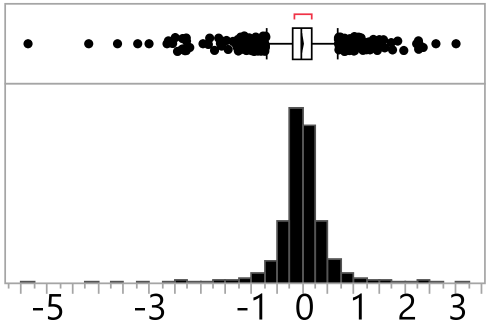
Training Data

$mean = -0.04\%, std.dev = 0.37\%$



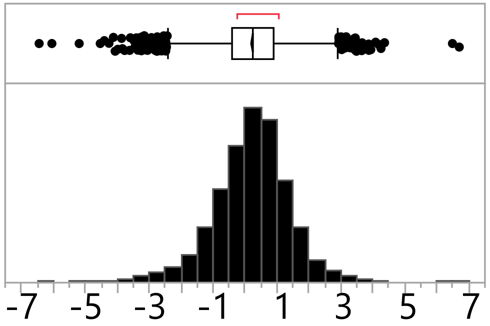
Verification Data

$mean = -0.01\%, std.dev = 0.45\%$

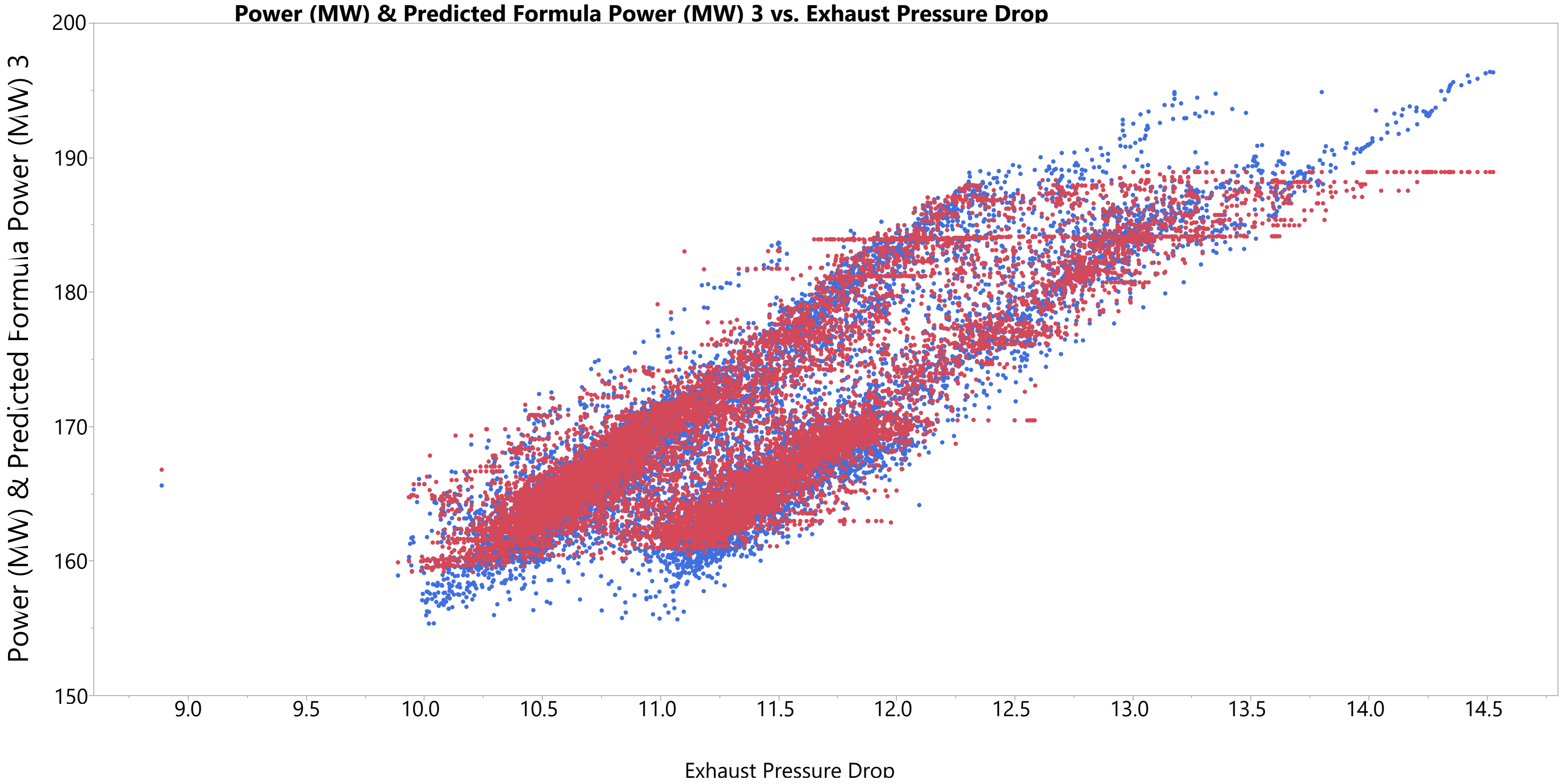


Validation Data

$mean = 0.22\%, std.dev = 1.121\%$

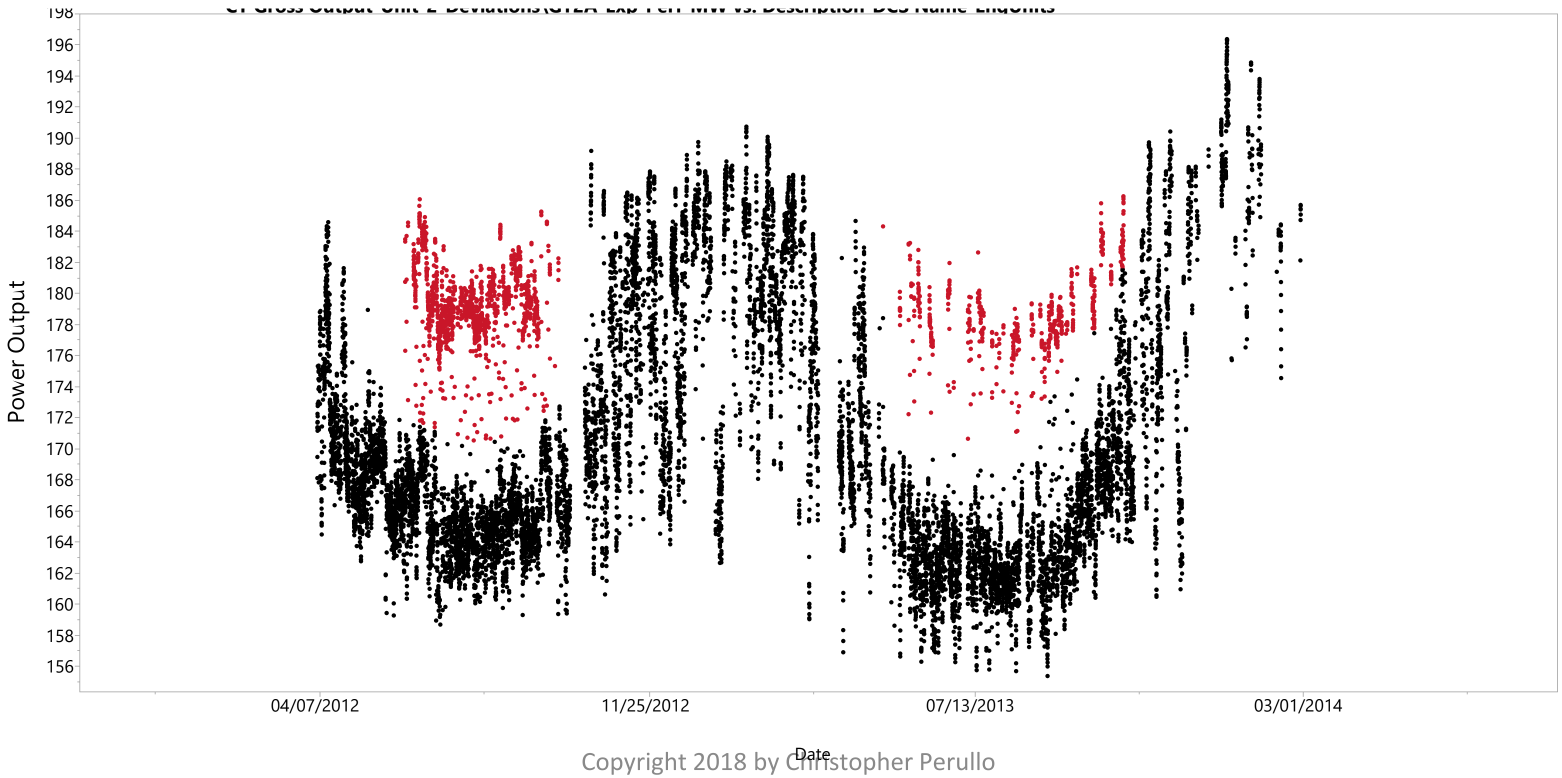


How Well Does Clustering Capture Bi-Modal Behavior?



Neural Network Categorized Model

- Maybe we want to develop neural network to pre-screen performance data
- Can we use a neural network to identify points with steam injection? (marked in red)



Input List

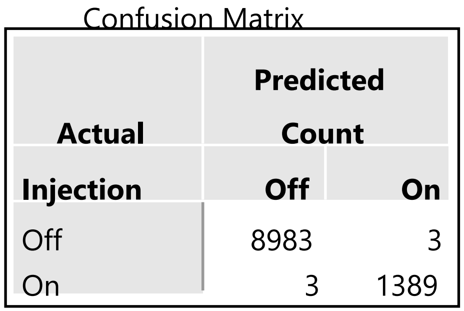
- Previous example input list
  - Compressor Inlet Temperature
  - Compressor Inlet Pressure Drop
  - Exhaust Pressure Drop
  - Barometric Pressure
  - Natural Gas (or fuel) Temperature
  - Relative Humidity
- Add additional information about state of the unit
  - Compressor discharge pressure and temperature
  - Exhaust Gas Temperature
  - Fuel flow
  - Mass flow (bellmouth sensor)
  - Power Output

What Structure?

- Already know from prior example two layer network works better
- 13 inputs & 1 output
- Let’s try:
  - 2 layers
  - 13 hidden nodes in each layer
  - TanH activation function

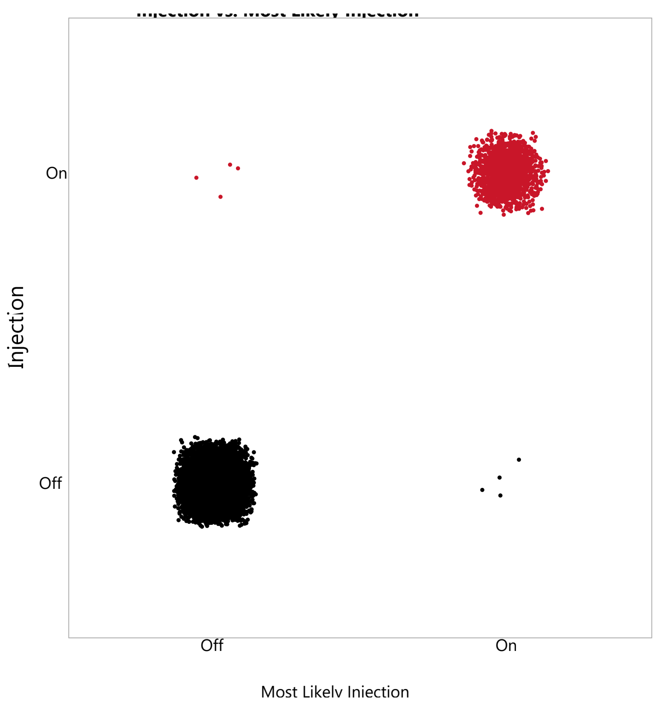
Predicting Steam Injection

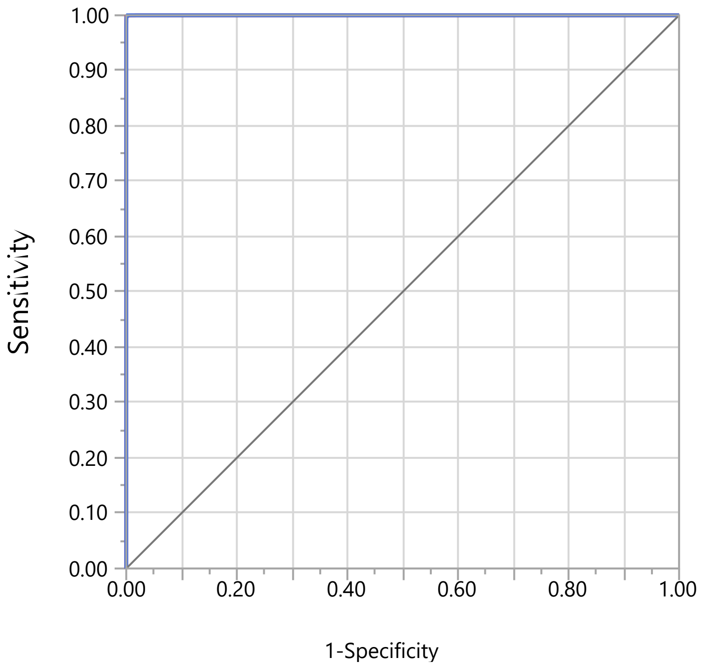
- Great Prediction!
- A word of caution
  - Models built from measured data may not be applicable to other units



Confusion Matrix

Actual	Predicted Count	
Injection	Off	On
Off	8983	3
On	3	1389





Advanced Tips and Tricks

Other Tricks

- Nesting / Layered Models
  - Create layered models where output of one becomes input to another
  - Requires model checks to work from chained error, not individual fits
- Transformed Variables
  - Apply log or exponential transformations to responses (outputs of model)
  - Make sure to un-transform for calculation of error checks
- Fit probability parameters
  - If data has random variation – fit distribution and then fit distribution parameters using machine learning

Questions?

Christopher Perullo

Senior Research Engineer

chris.perullo@ae.gatech.edu