

Building Better Models

82nd MORS Symposium

Alexandria, VA

June 16th, 2014

Tom Donnelly, PhD

Systems Engineer & Co-insurrectionist



THE
POWER
TO KNOW[®]

Outline

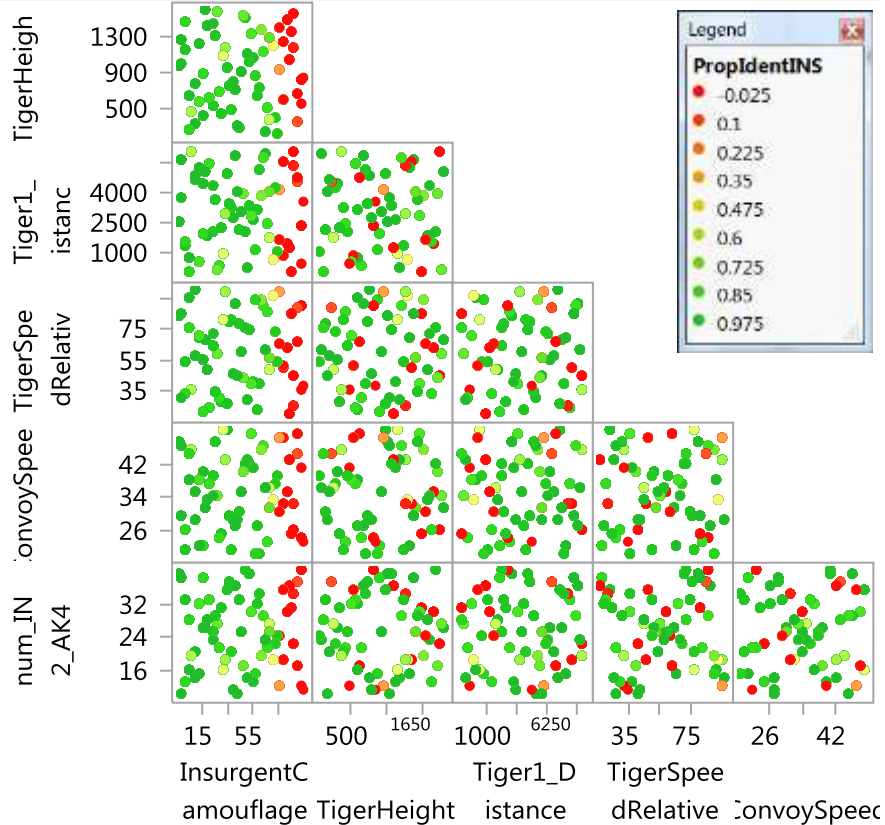
- Case Study 1 Preview
- Introduction to Modeling
- Honest Assessment Method to *Prevent Overfitting*
- Regression and Model Selection
- Case Study 2
- Decision Trees
- Neural Models
- Model Comparison

Surrogate Modeling of a Computer Simulation - Helicopter Surveillance – Identifying Insurgents

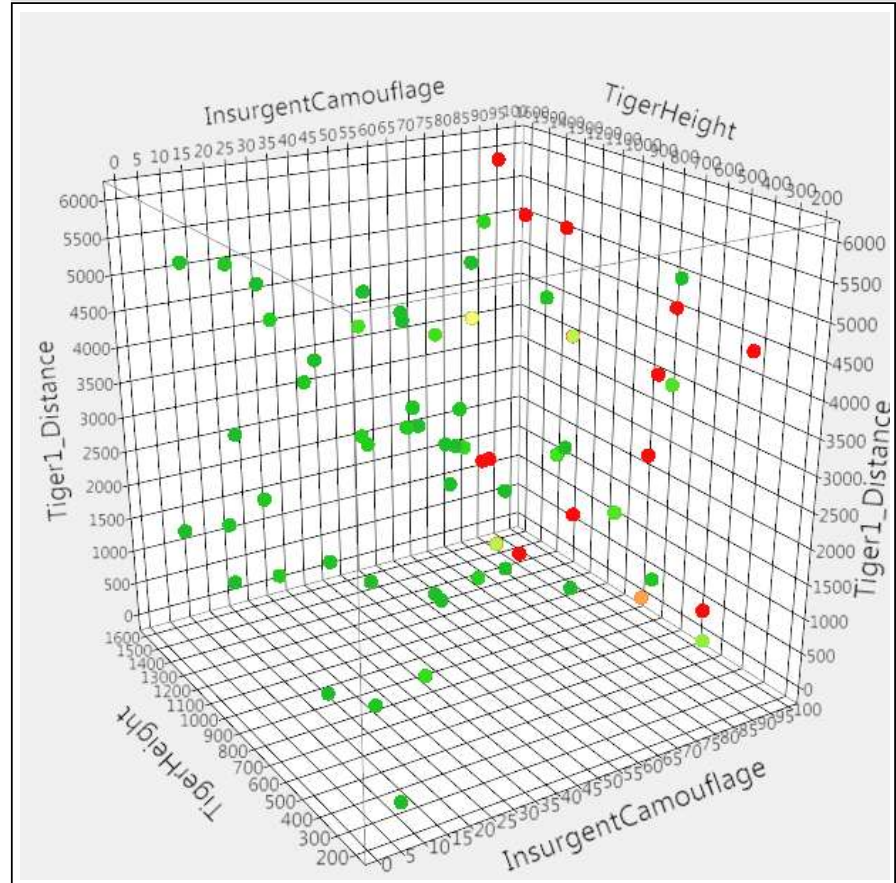
- 2009 International Data Farming Workshop - IDFW21, Lisbon, Portugal
- Largely German team (6 of 8) – their simulation
- 6500 simulations run overnight on cluster in Frankfurt
 - 65 unique combinations of 6 factors (each factor at 65 levels)
 - each case had 97 to 100 replications (lost a few)
- Response = Proportion of Insurgents Identified = *PropIdentINS* Data bounded between 0 and 1
- Explore data visually first
- Fit many different models – “Train, Validate (Tune), Test” 60/20/20 subsets
- Compare Actual vs. Predicted for Test Set

Preview End Result – Space-Filling DOE

Scatterplot Matrix

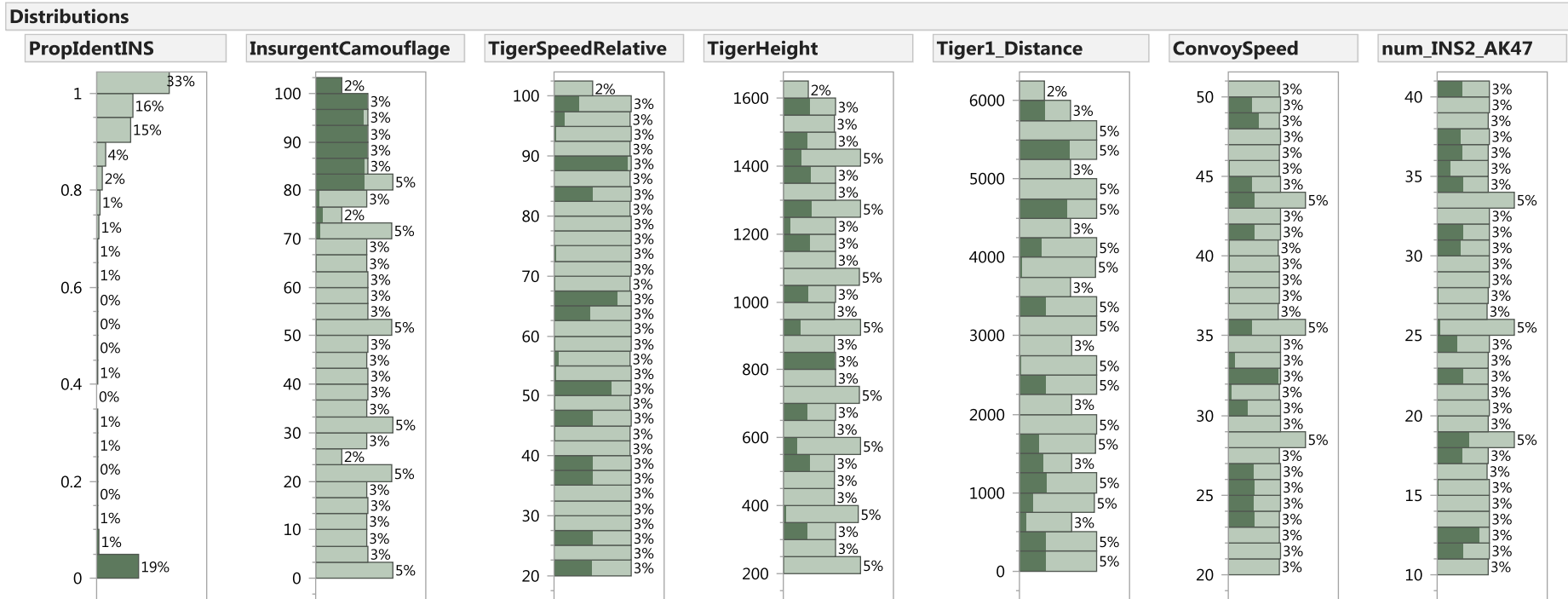


Scatterplot 3D



Data Columns: InsurgentCamouflage TigerHeight Tiger1_Distance

Distributions of Response and 6 Factors

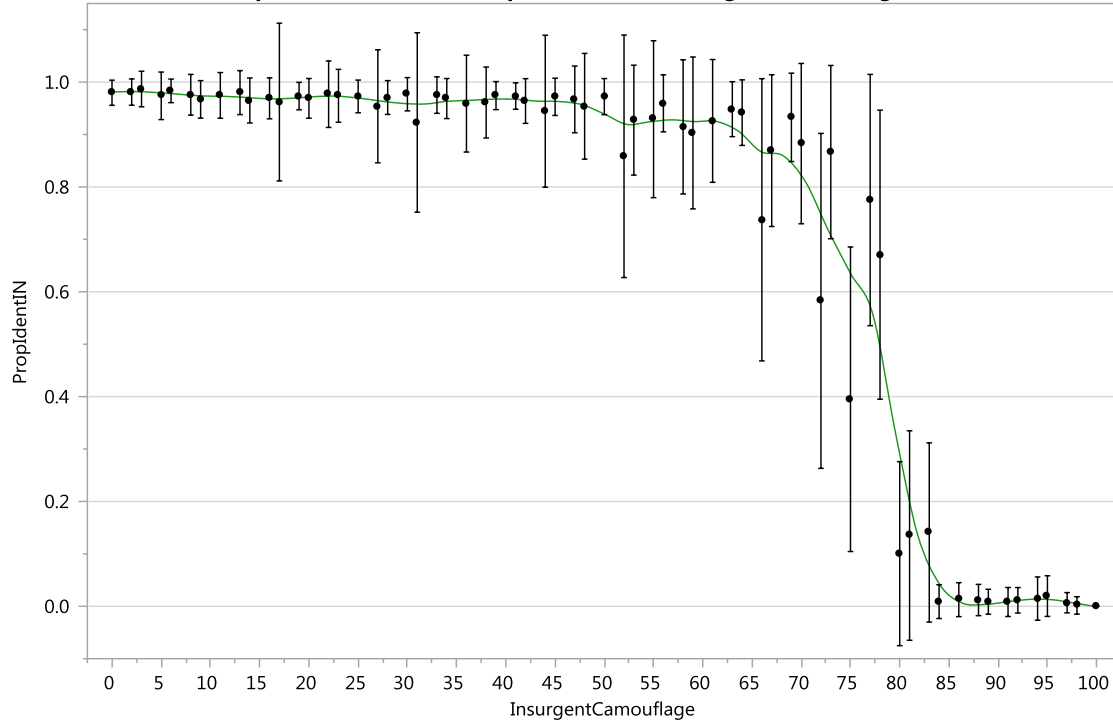


Before modeling look for correlations between good or poor levels of *PropIdentINS* and the factors. Strong correlation between poor *PropIdentINS* and high levels of *InsurgentCamouflage*. No other factor shows very much correlation with the response.

PropldentINS vs. X for 6 Factors

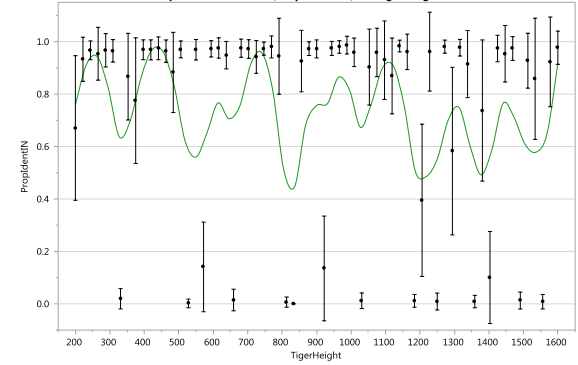
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. InsurgentCamouflage



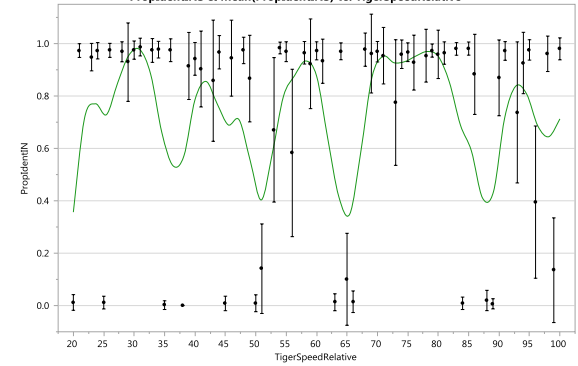
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. TigerHeight



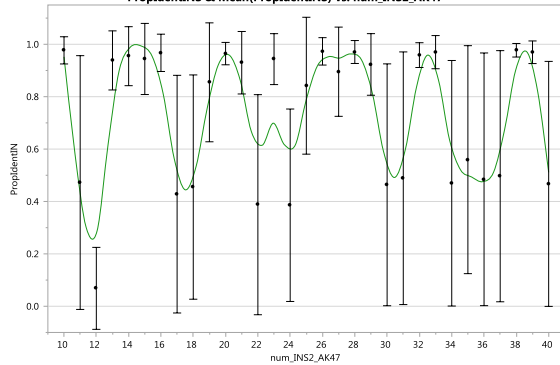
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. TigerSpeedRelative



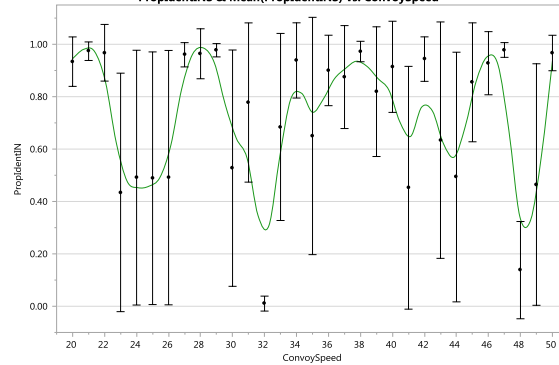
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. num_INS2_AK47



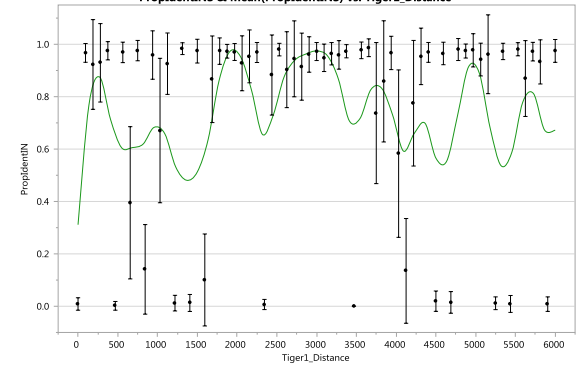
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. ConvoySpeed



Graph Builder

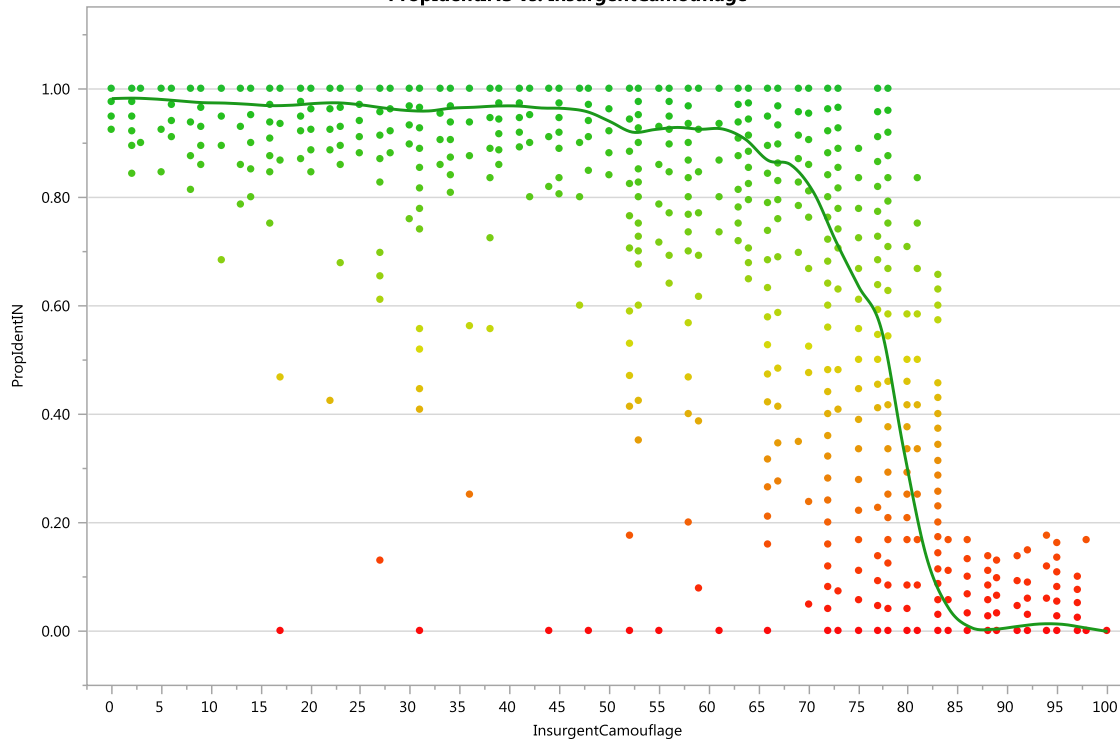
PropIdentINS & Mean(PropIdentINS) vs. Tiger1_Distance



PropldentINS vs. X for 6 Factors

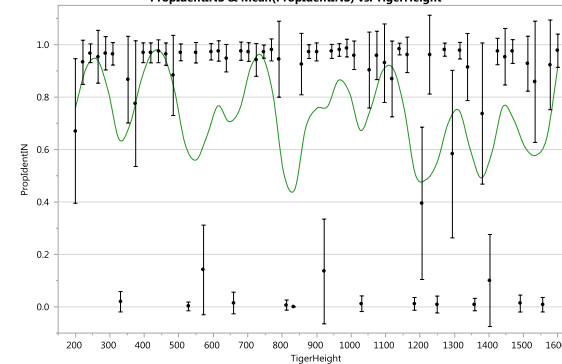
Graph Builder

PropIdentINS vs. InsurgentCamouflage



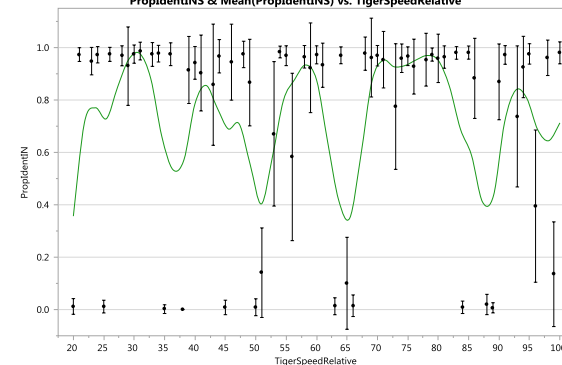
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. TigerHeight



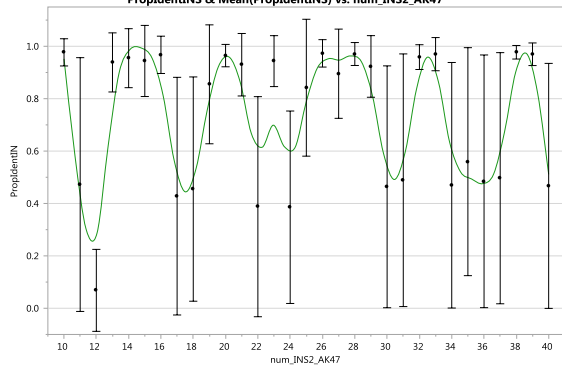
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. TigerSpeedRelative



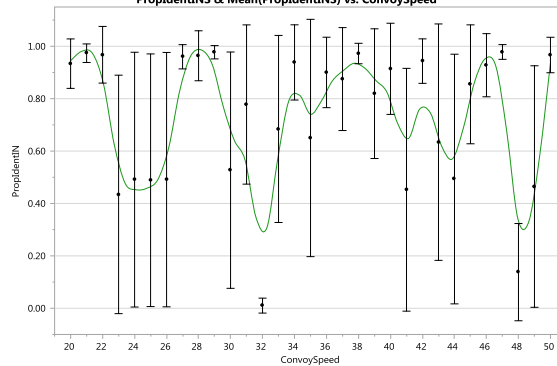
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. num_INS2_AK47



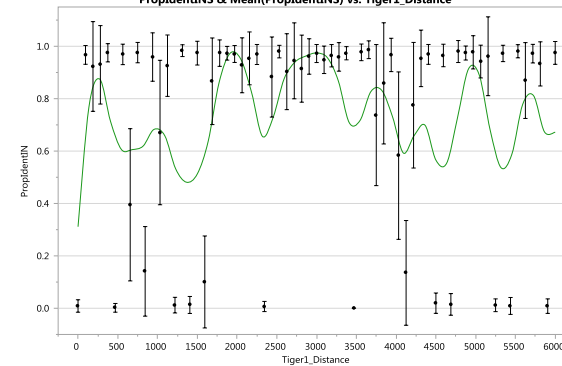
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. ConvoySpeed



Graph Builder

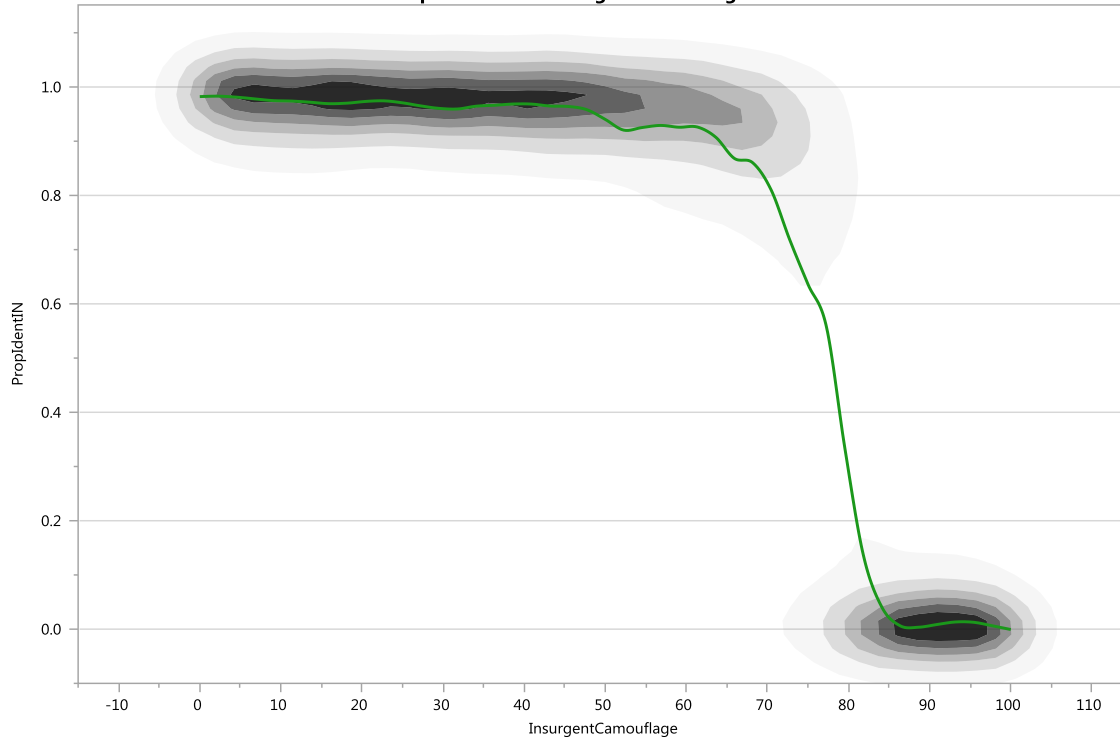
PropIdentINS & Mean(PropIdentINS) vs. Tiger1_Distance



PropldentINS vs. X for 6 Factors

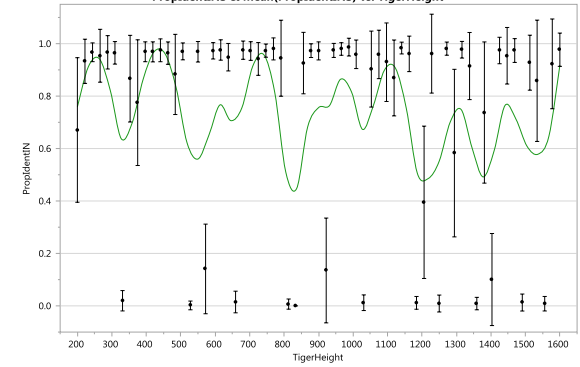
Graph Builder

PropIdentINS vs. InsurgentCamouflage



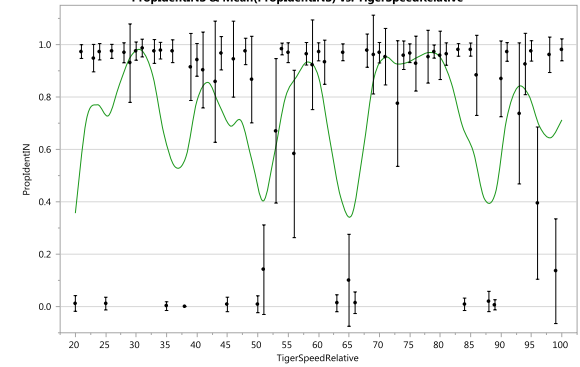
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. TigerHeight



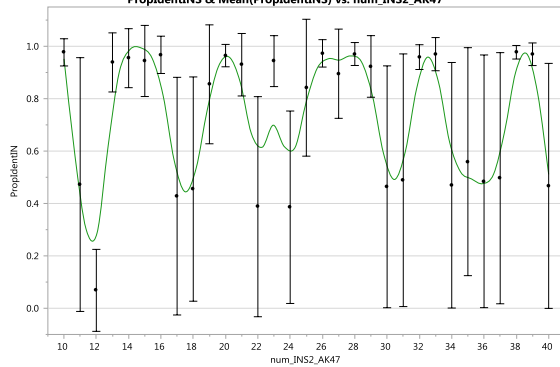
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. TigerSpeedRelative



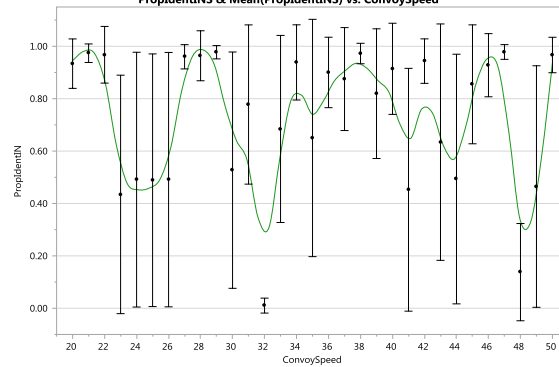
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. num_INS2_AK47



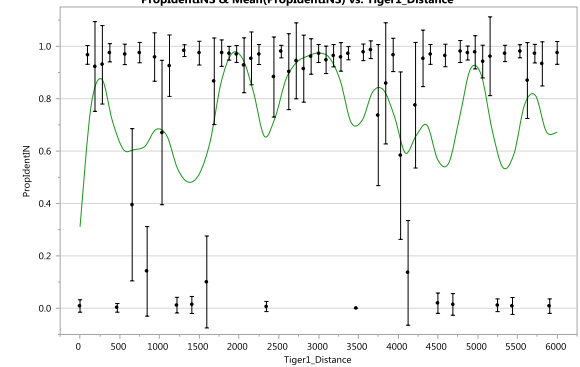
Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. ConvoySpeed

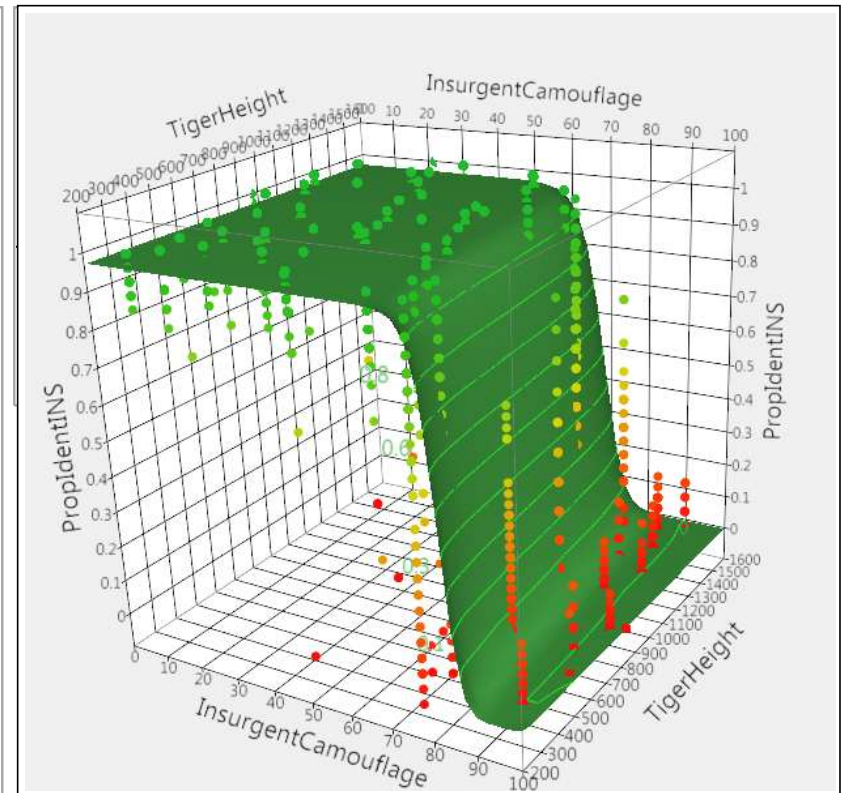
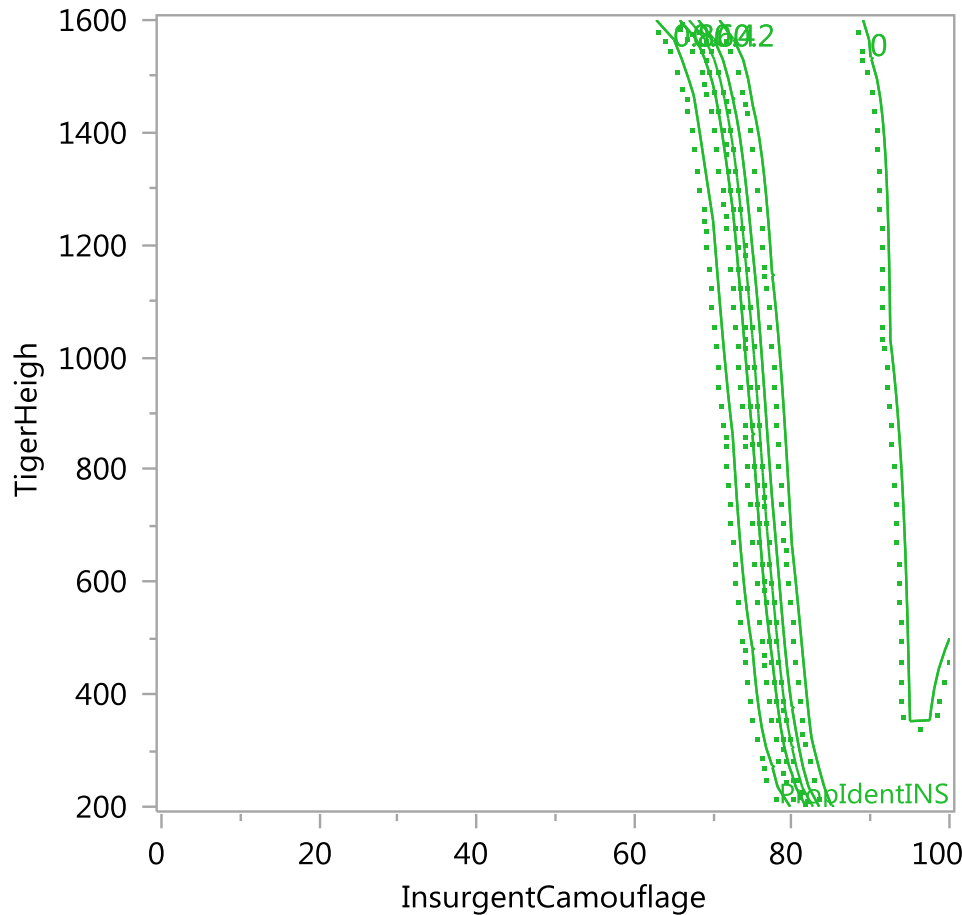


Graph Builder

PropIdentINS & Mean(PropIdentINS) vs. Tiger1_Distance

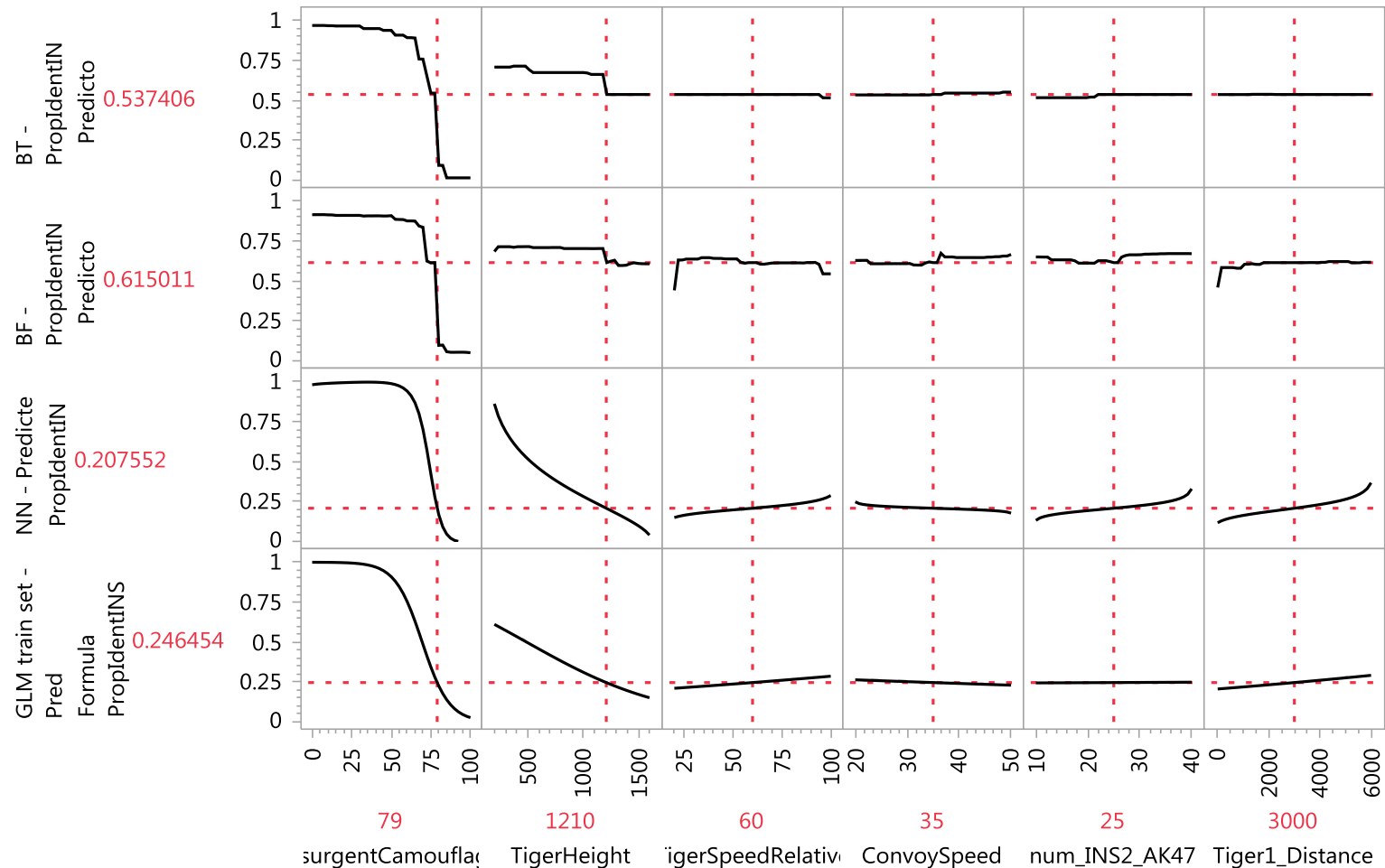


2-D Contour Plot and 3-D Response Surface ProIdentINS vs. Camouflage & Height



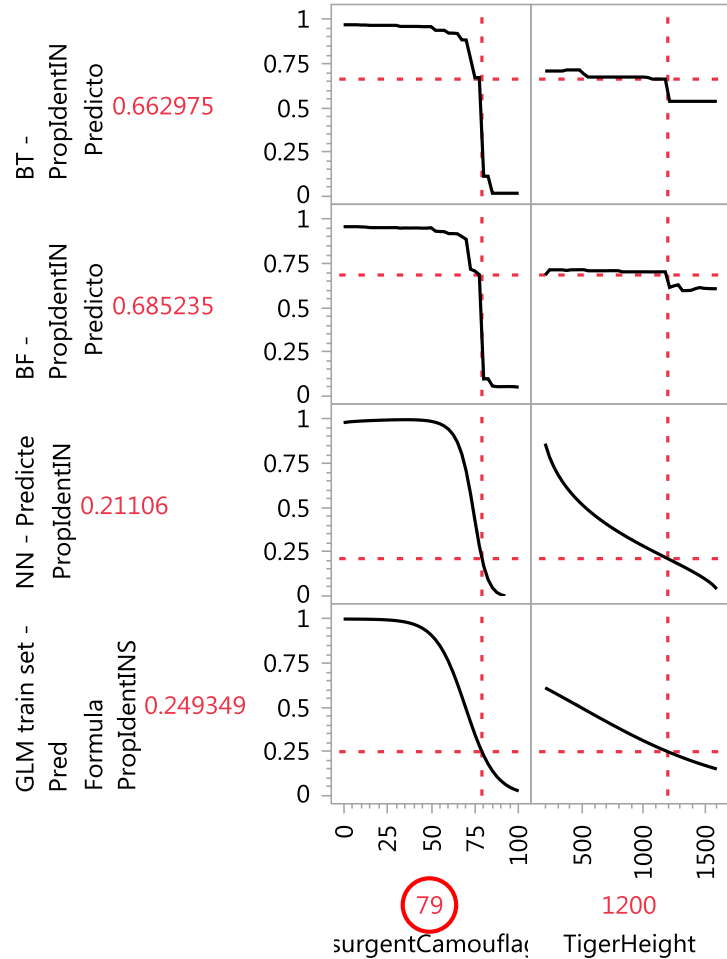
Compare Several Models – top 2 are decision tree variants bottom two are “smoother” models - Neural Net and GLM

Prediction Profiler

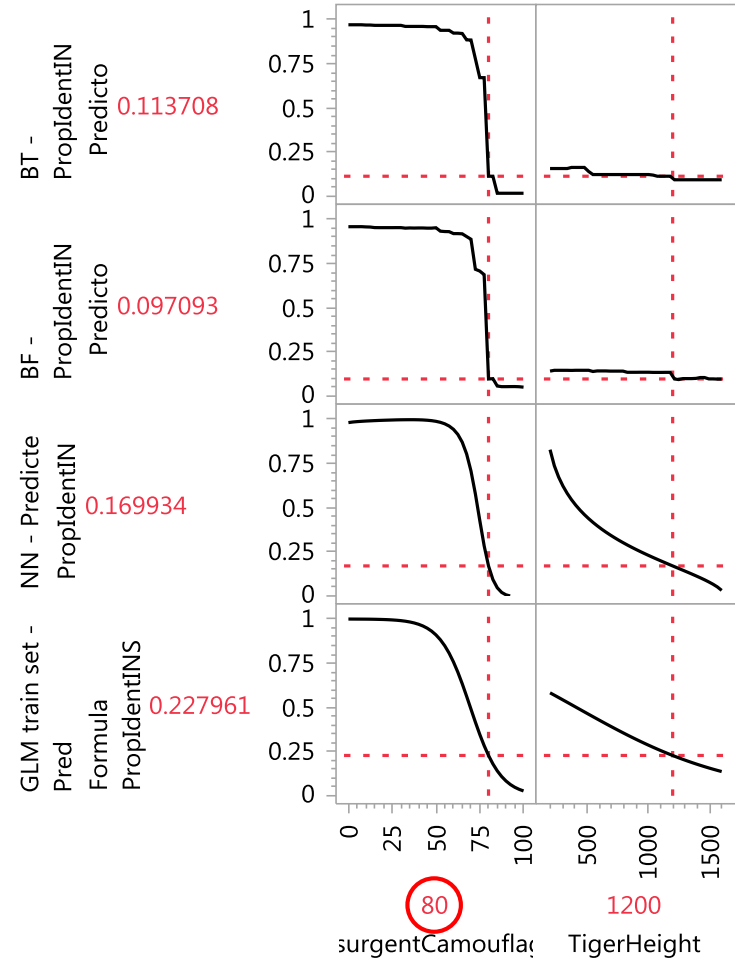


Change Camouflage from 79 to 80 and Decision Tree Predictions Drop by 6X – Talk to Developer?

Prediction Profiler

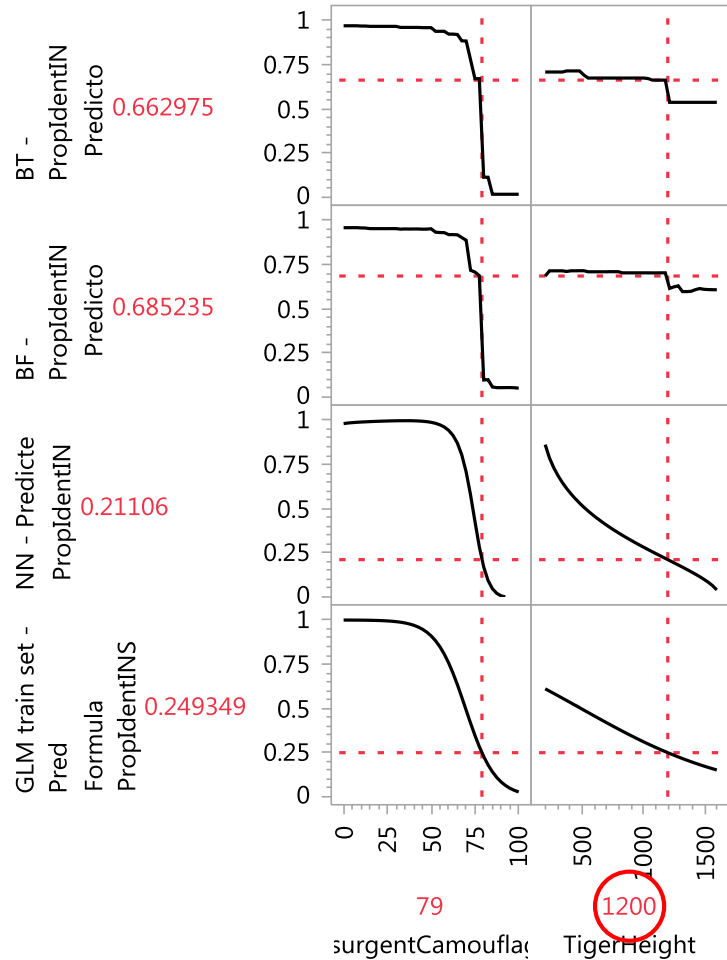


Prediction Profiler

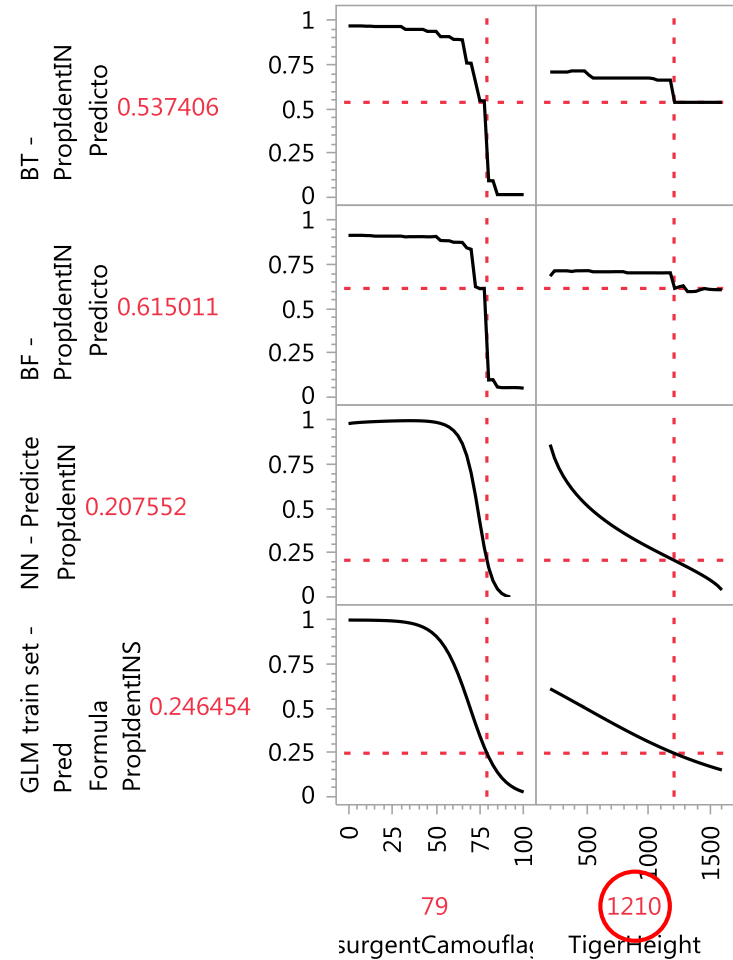


Change Tiger Height from 1200 to 1210 and Decision Tree Predictions Drop by 10% to 20%! – Plausible?

Prediction Profiler

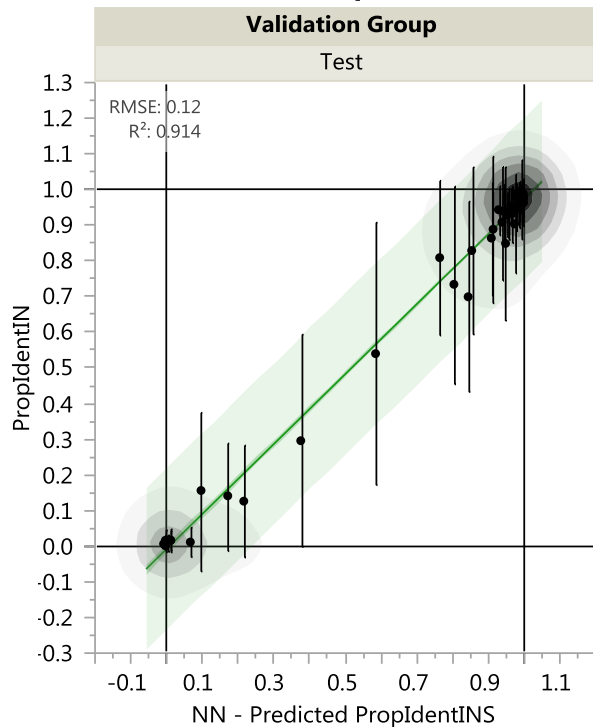


Prediction Profiler

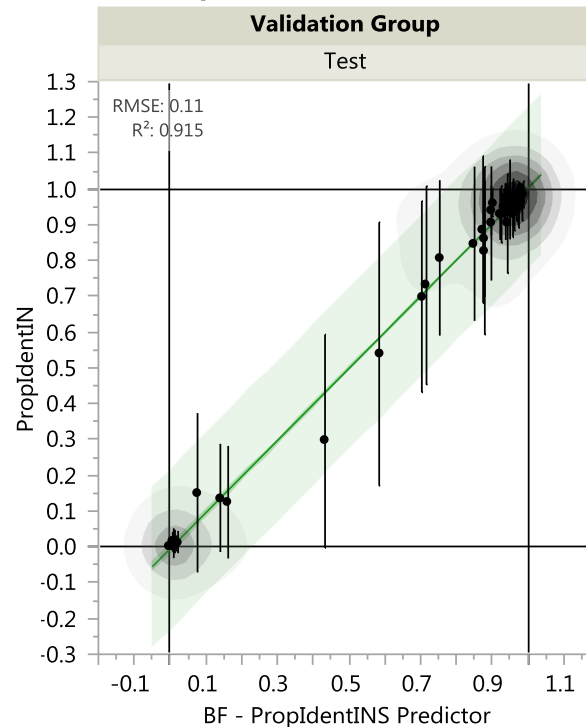


Actual vs. Predicted Plots for *Test Data* Neural Net, Bootstrap Forest and GLM Models

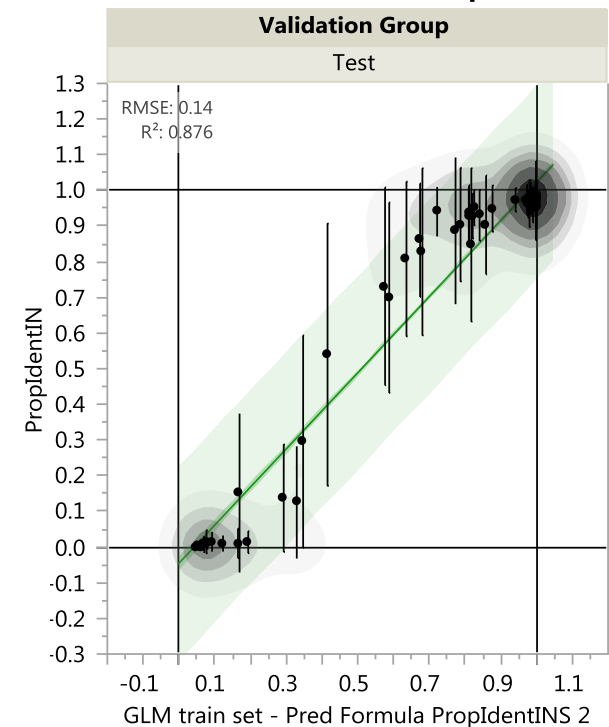
PropIdentINS & Mean(PropIdentINS) vs.
NN - Predicted PropIdentINS



PropIdentINS & Mean(PropIdentINS) vs. BF
- PropIdentINS Predictor



PropIdentINS & Mean(PropIdentINS) vs.
GLM train set - Pred Formula PropIdentIN



Introduction to Modeling



THE
POWER
TO KNOW.®

Model Quotes

- “No *good* model ever accounted for all the facts, since some data was bound to be misleading, if not wrong.”
 - James Dewey Watson (1988)
- “Essentially, *all* models are wrong, but some are useful.”
 - George Box (1987)
- “The purpose of models is *not* to fit the data but to sharpen the questions.”
 - Samuel Karlin (1983)
- “The *best* material model of a cat is another, or preferably the *same*, cat.”
 - A. Rosenbleuth (1945)

What is a statistical model?

- An empirical model that relates a set of inputs (predictors, \mathbf{X}) to one or more outcomes (responses, \mathbf{Y})
- Separates the response variation into signal and noise

$$\mathbf{Y} = f(\mathbf{X}) + \mathbf{E}$$

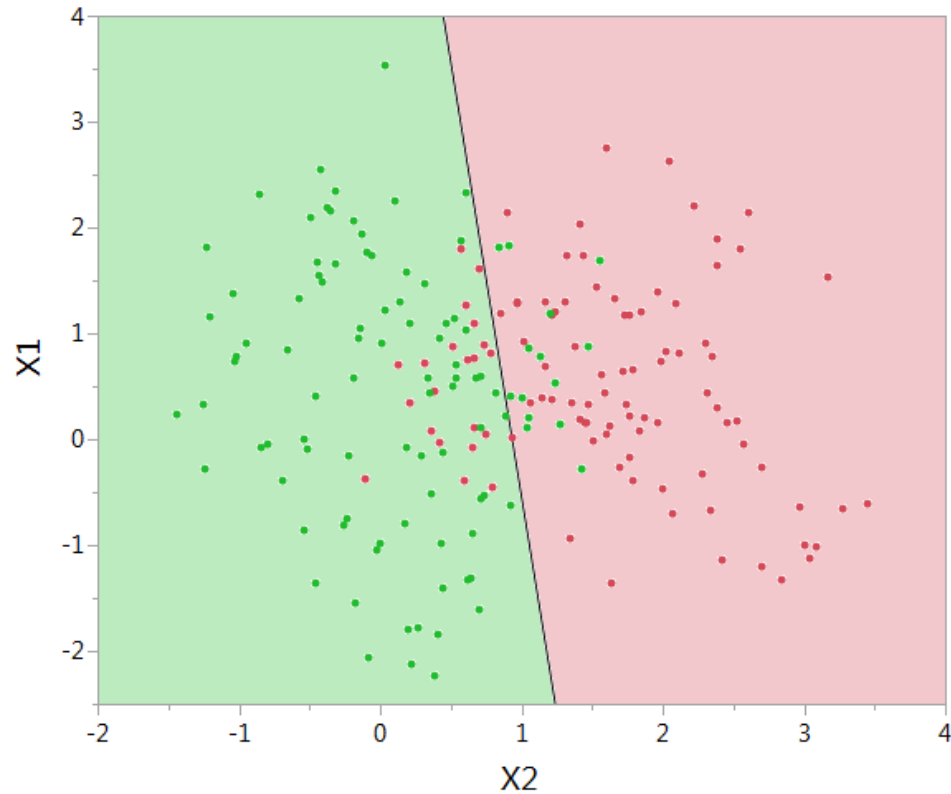
- \mathbf{Y} is one or more continuous or categorical response outcomes
 - \mathbf{X} is one or more continuous or categorical predictors
 - $f(\mathbf{X})$ describes predictable variation in \mathbf{Y} (signal)
 - \mathbf{E} describes non-predictable variation in \mathbf{Y} (noise)
- The mathematical form of $f(\mathbf{X})$ can be based on domain knowledge or mathematical convenience.

What is a predictive model?

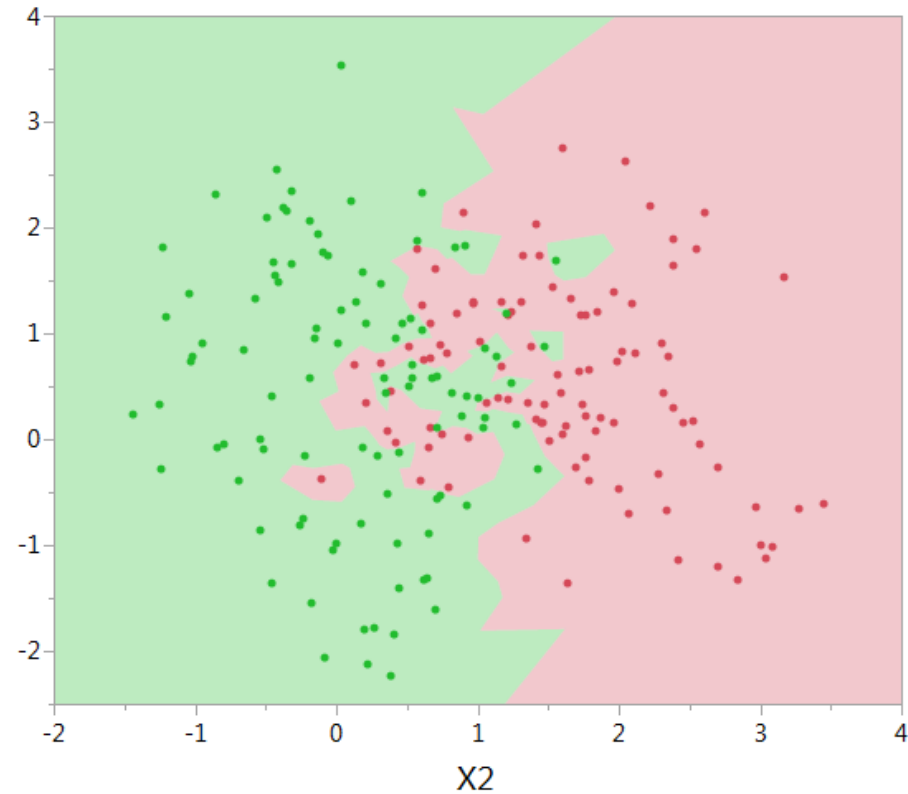
- A type of statistical model where the focus is on predicting Y independent of the form used for $f(\mathbf{X})$.
 - There is less concern about the form of the model – parameter estimation isn't important. The focus is on how well it predicts.
 - Very flexible models are used to allow for a greater range of possibilities.
 - http://en.wikipedia.org/wiki/Predictive_modelling

What is a predictive model?

- Two Examples:



Regression



Nearest Neighbor

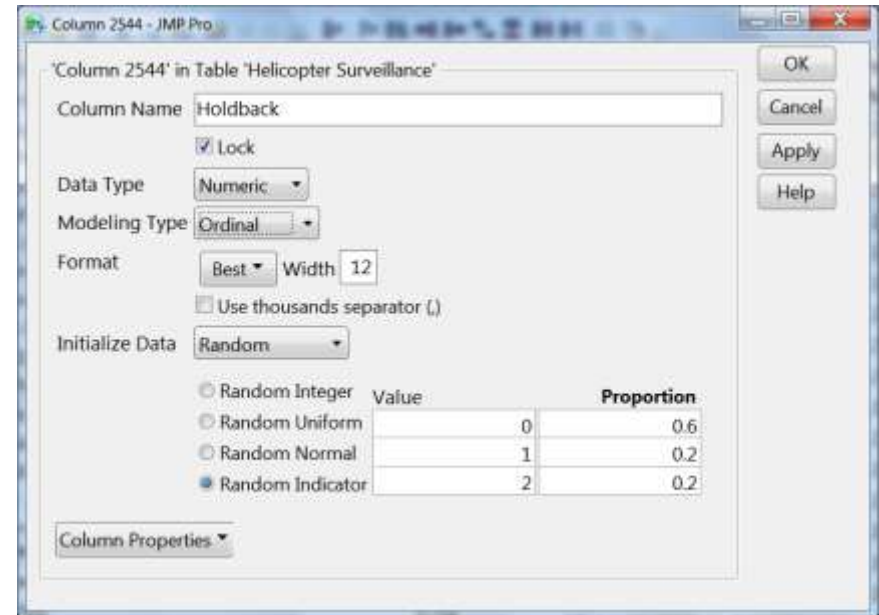
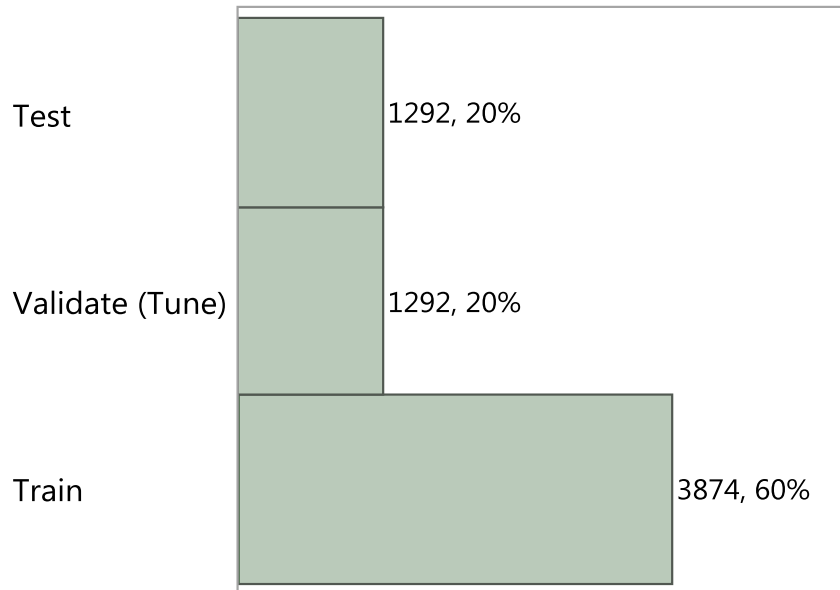
Preventing Model Overfitting

- If the model is flexible what guards against overfitting (i.e., producing predictions that are too optimistic)?
 - Put another way, how do we protect from trying to model the noise variability as part of $f(\mathbf{X})$?
- Solution – Hold back part of the data, using it to check against overfitting. Break the data into two or three sets:
 - The **training** set is used to **build** or **fit** the model
 - The **validation** set is used to **select** model by determining when the model is becoming too complex – it **tunes** the parameters
 - The **test** set is often used to **evaluate** how well model predicts independent of training and validation sets
 - Common methods include random holdback and k-fold crossvalidation

Honest Assessment Approach Using Train, Validate (Tune), and Test Subsets

Used in model selection and estimating its prediction error on new data

Validation Group



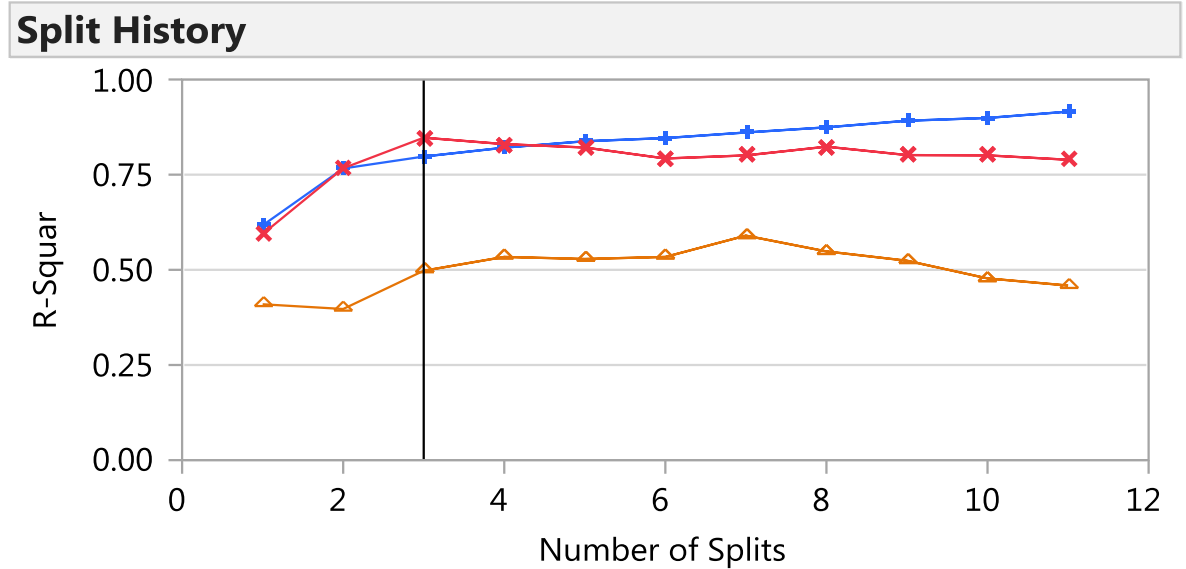
The Elements of Statistical Learning – Data Mining, Inference, and Prediction

Hastie, Tibshirani, and Friedman – 2001

(Chapter 7: Model Assessment and Selection)

Honest Assessment Approach Using Train, Validate (Tune), and Test Subsets

Train, Validate, Test
R-Square vs. #Splits
Decision Tree Model



Validation Data in Red
Test Data in Orange

K-fold Crossvalidation

- Smaller datasets may not have enough rows to split into train/validate/test, so instead we can use k-fold cross-validation:
 - Randomly divide data into k separate groups (“folds”) (k=5 to k=10 is recommended)



- Hold out one of the folds from model building and fit a model to the rest of the data.
- Use the held out fold as a validation set

K-fold Crossvalidation

- Smaller datasets may not have enough rows to split into train/validate/test, so instead we can use k-fold cross-validation:

- Randomly divide data into k separate groups (“folds”) (k=5 to k=10 is recommended)

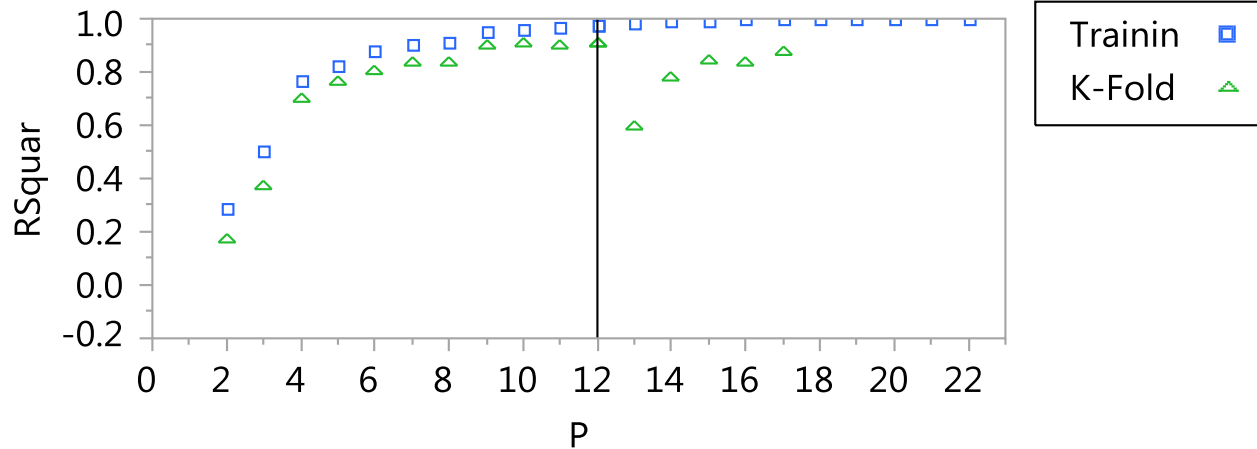


- Hold out one of the folds from model building and fit a model to the rest of the data.
- Use the held out fold as a validation set
- Repeat across all folds
- The model giving the best validation R-Square is chosen as the final model.

K-fold Crossvalidation

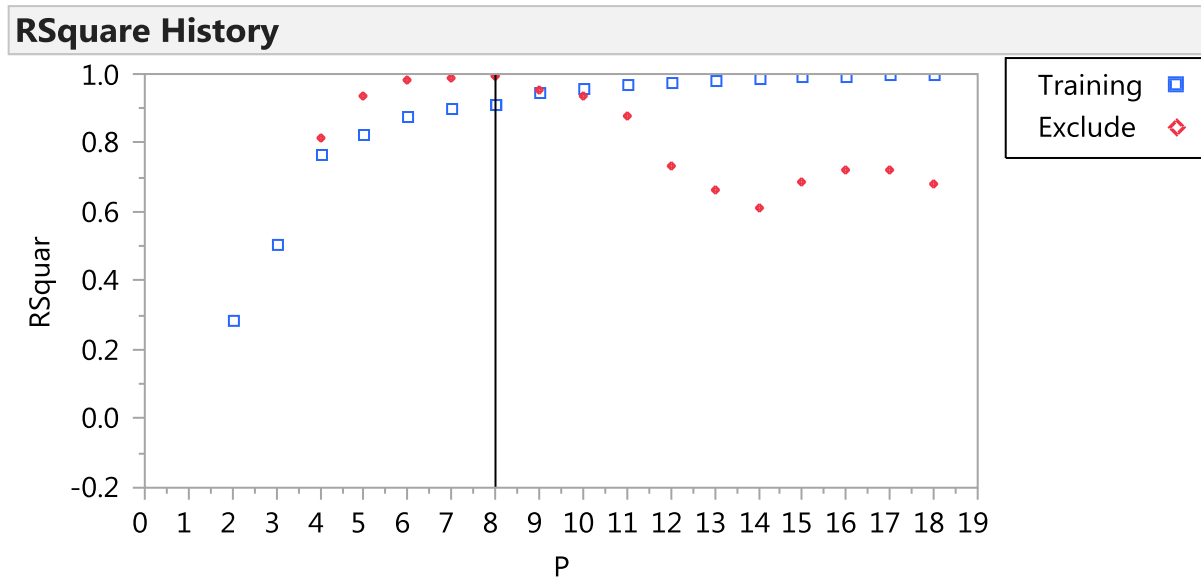
Training & K-Fold R-Square vs. Model Term History for Stepwise Regression

RSquare History



Data can be excluded and used for validation – e.g. checkpoints

Training & Excluded R-Square vs. Model Term History for Stepwise Regression



AICc and BIC Criterion

AICc and BIC deal with the trade-off between the goodness of fit of the model and the complexity of the model.

$$AIC = 2k - 2 \ln(L)$$

$$AICc = AIC + \frac{2k(k+1)}{n-k-1}$$

$$BIC = -2 \cdot \ln \hat{L} + k \cdot \ln(n).$$

For any statistical model, the Akaike Information Criterion (AIC) value is where k is the number of parameters in the model, and L is the maximized value of the likelihood function for the model.

AICc is AIC with a correction for finite sample sizes: where n denotes the sample size. Thus, AICc is AIC with a greater penalty for extra parameters.

For large n , the Bayesian Information Criterion can be approximated by:

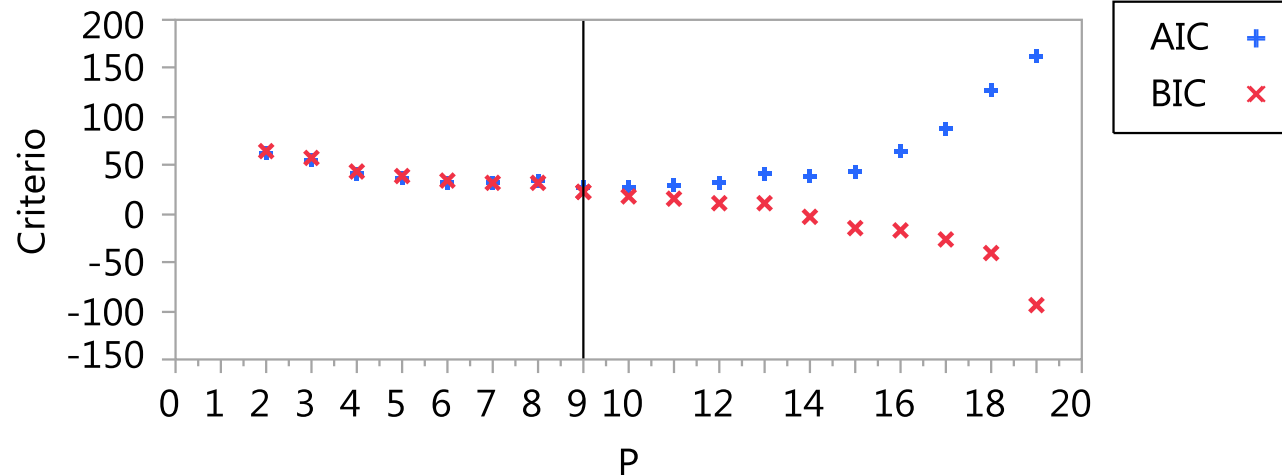
The BIC generally penalizes free parameters more strongly than does the AIC, though it depends on the size of n and relative magnitude of n and k .

AICc and BIC Criterion

Use AICc & BIC stopping criteria and pick “simpler model” – Occam’s razor

AICc and BIC Criterion vs. Model Term History for Stepwise Regression

Criterion History



Both AICc and BIC are measures of model fit that are helpful when comparing models. Smaller values indicate a better fit.

Regression and Model Selection

Overview



Regression

- General linear regression typically uses simple polynomial functions for $f(\mathbf{X})$.
 - For continuous y :

$$f(x) = \beta_0 + \sum_{i=1}^p \beta_i x_i + \sum_{i=1}^p \sum_{j=i+1}^p \gamma_{i,j} x_i x_j + \sum_{i=1}^p \delta_i x_i^2$$

- For categorical y , the logistic function of $f(\mathbf{X})$ is typically used.

$$\frac{1}{1 + e^{-f(x)}}$$

Model Selection

- Stepwise Regression
 - Start with a base model:
 - » intercept only (forward selection) or
 - » all terms (backwards selection)
 - If intercept only, find term not included that explains the most variation and enter it into the model.
 - If all terms, remove the term that explains the least.
 - Continue until some sort of stopping criterion is met.
 - » P-value, AICc, BIC, K-fold R-Square, Excluded R-Square
- A variation of stepwise regression is all possible subsets (best subset) regression.
 - Examine all 2, 3, 4, ..., etc. term models and pick the best out of each. Sometimes statistical heredity is imposed to make the problem more tractable.

Model Selection

- Drawbacks
 - Selection is all or nothing. The term either is in the model or isn't.
 - Correlated data can lead to unstable parameter estimates
 - For stepwise regression, optimal search may not follow a linear algorithmic path. Adding the best term at each step may not produce the best overall model.
 - Large models may be impossible to examine using all subsets regression.
- Shrinkage Methods
 - Attempts to simultaneously minimize the prediction error and shrink the parameter estimates toward zero. Resulting estimates are biased, but prediction error is often smaller.
 - Can be considered as continuous model term selection.
 - Common techniques: Ridge Regression, LASSO, Elastic Net.

Handling missing predictor values

- Case-wise deletion – Easy, but reduces the sample
- Simple imputation – Replace the value with the variable mean or median
- Multivariate imputation – Use the correlation between multiple variables to determine what the replacement value should be
- Model based imputation – Model with the non-missing values, replace missing values based on similar cases
- Model free imputation – e.g., distance based, hot hand, etc.
- Methods insensitive to missing values

Ways to account for missing data

- Categorical
 - Creates a separate level for missing data and treats it as such.
- Continuous
 - Informative Missing/Missing Value Coding:
 - » Regression and Neural Network: The column mean is substituted for the missing value. An indicator column is included in the predictors where rows are 1 where data is missing, 0 otherwise. This can significantly improve the fit when data is missing not at random.
 - » Partition: the missing observations are considered on both sides of the split. It is grouped with the side providing the better fit.
 - Save Tolerant Prediction Formula (Partition):
 - » The predictor is randomly assigned to one of the splits.
 - » Use when Informative Missing approach is not used.

Ways to account for missing data

- Multivariate Imputation – Based on the correlation structure of the continuous predictors (the expectation conditional on the nonmissing data).
- Model Based Imputation – Impute missing predictors based on partial least squares model.

Case Study: Regression

CO2 Capture



THE
POWER
TO KNOW®

Stepwise Regression Case Study

July 22, 2010

Secretary Chu Announces Six Projects to Convert Captured CO2 Emissions from Industrial Sources into Useful Products

\$106 Million Recovery Act Investment will Reduce CO2 Emissions and Mitigate Climate Change

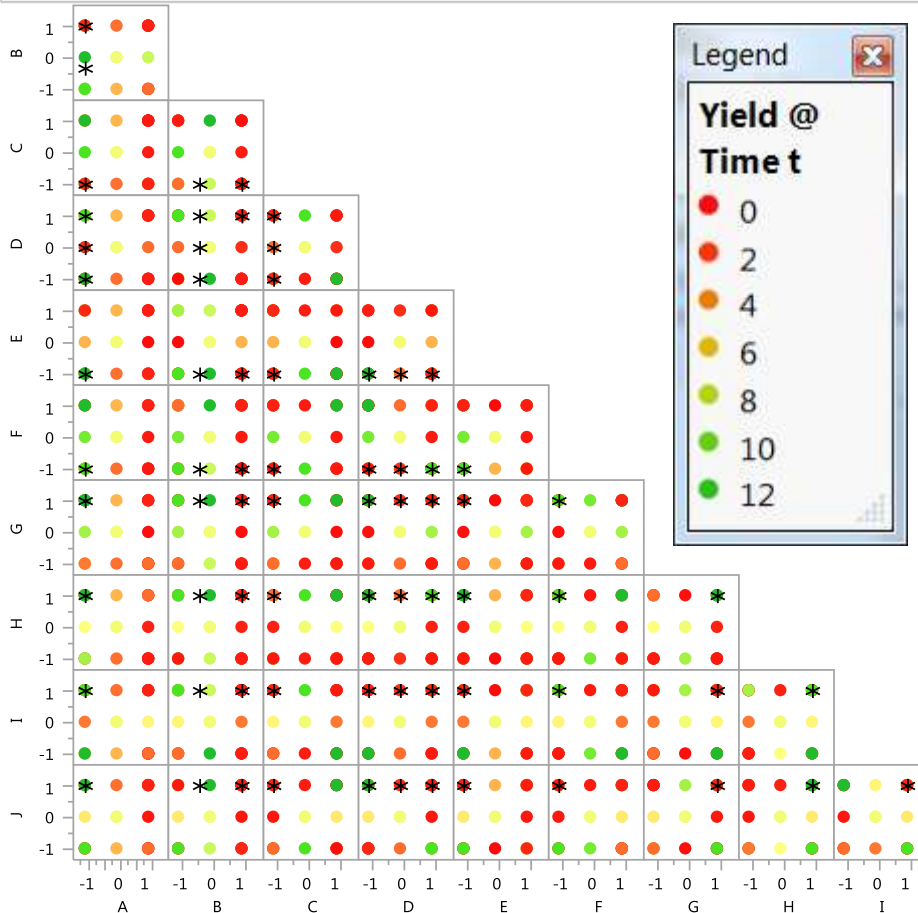
Washington, D.C. - U.S. Energy Secretary Steven Chu announced today the selections of six projects that aim to find ways of converting captured carbon dioxide (CO2) emissions from industrial sources into useful products such as fuel, plastics, cement, and fertilizers. Funded with \$106 million from the American Recovery and Reinvestment Act -matched with \$156 million in private cost-share -today's selections demonstrate the potential opportunity to use CO2 as an inexpensive raw material that can help reduce carbon dioxide emissions while producing useful by-products that Americans can use.

"These innovative projects convert carbon pollution from a climate threat to an economic resource," said Secretary Chu. "This is part of our broad commitment to unleash the American innovation machine and build the thriving, clean energy economy of the future."

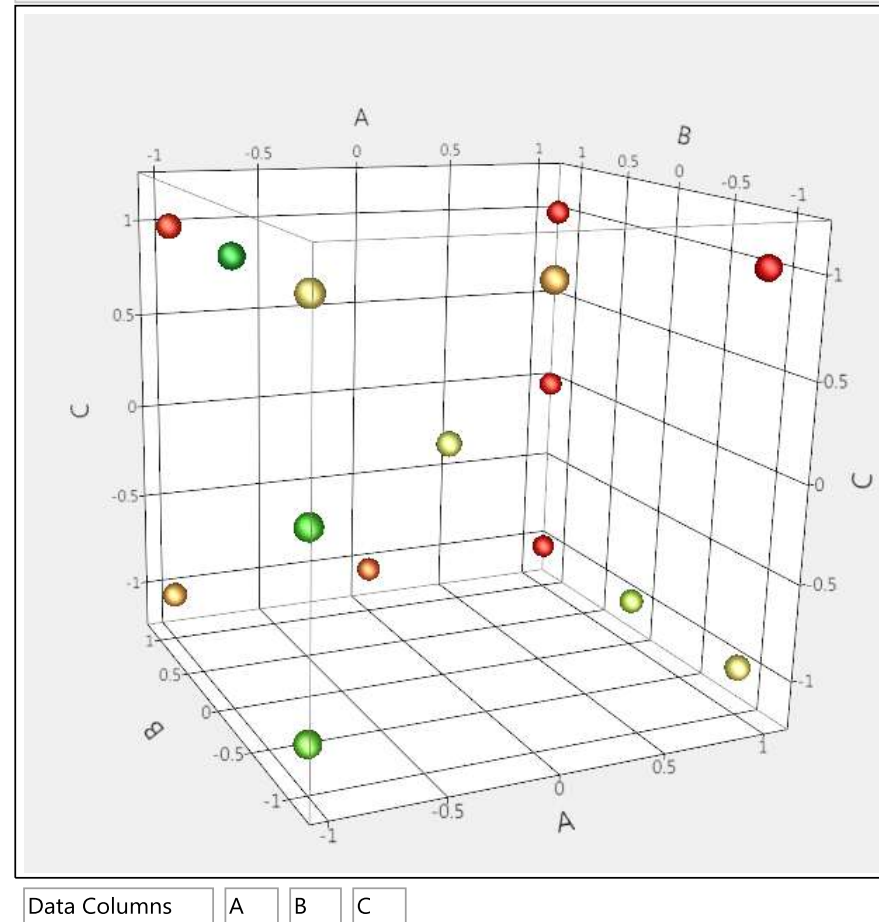
23/1		Yield @ Time t	A	B	C	D	E	F	G	H	I	J
●	1	1.38	-1	1	1	0	1	-1	1	-1	1	1
●	2	6.44	1	-1	-1	-1	1	-1	1	1	0	1
●	3	5.96	-1	-1	1	-1	-1	1	-1	1	1	0
●	4	4.34	0	-1	1	1	1	1	1	1	-1	-1
●	5	10.46	-1	-1	-1	-1	-1	0	1	-1	-1	-1
●	6	6.95	-1	-1	1	-1	1	-1	-1	0	-1	-1
●	7	8.58	1	0	-1	1	1	-1	-1	-1	1	-1
●	8	2.69	0	1	-1	-1	-1	-1	-1	-1	1	1
●	9	4.3	-1	1	-1	1	0	-1	-1	1	-1	1
●	10	0.77	1	-1	1	-1	0	1	1	-1	1	-1
●	11	2.87	-1	1	1	1	-1	1	-1	-1	0	-1
●	12	1.01	1	1	1	1	1	0	-1	1	1	1
●	13	9.47	-1	-1	-1	1	1	1	0	-1	1	1
●	14	7.49	0	0	0	0	0	0	0	0	0	0
●	15	0.98	1	1	-1	1	1	-1	1	-1	-1	0
●	16	0.86	1	1	1	-1	-1	-1	0	1	-1	-1
●	17	1.25	-1	1	-1	-1	1	1	1	1	1	-1
●	18	1.03	1	-1	1	1	-1	-1	-1	-1	-1	1
●	19	1.07	1	1	0	-1	1	1	-1	-1	-1	1
●	20	7.33	0	0	0	0	0	0	0	0	0	0
●	21	2.61	1	-1	-1	0	-1	1	-1	1	-1	-1
●	22	11.39	-1	-1	0	1	-1	-1	1	1	1	-1
●	23	12.96	-1	0	1	-1	-1	1	1	1	-1	1
●	24	1.18	1	1	-1	1	-1	1	1	0	1	1
* ●	25	15.93	-1	-0.333	-1	1	-1	-1	1	1	1	1
* ●	26	2.9	-1	1	-1	1	-1	-1	1	1	1	1
* ●	27	16.16	-1	-0.333	-1	-1	-1	-1	1	1	1	1
* ●	28	15.1	-1	-0.333	-1	0	-1	-1	1	1	1	1

Definitive Screening DOE in 10 Factors

Scatterplot Matrix



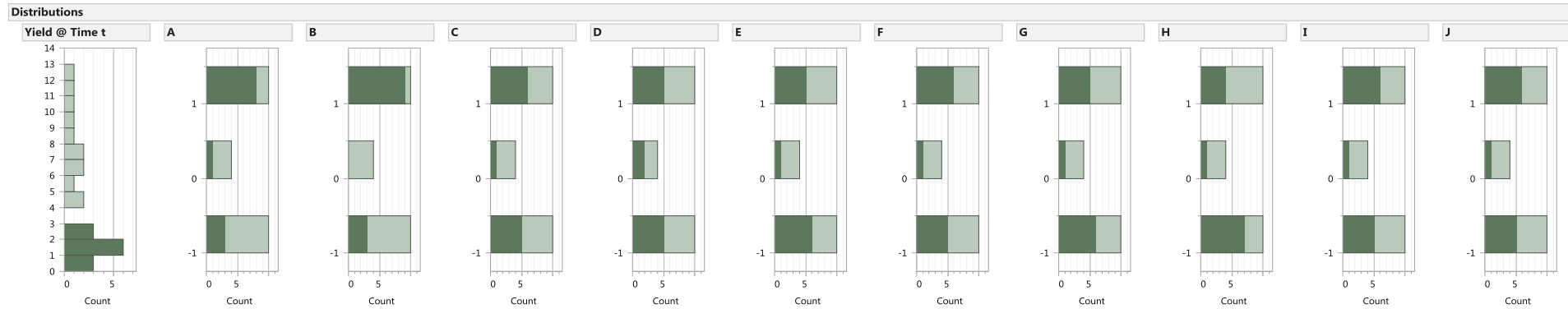
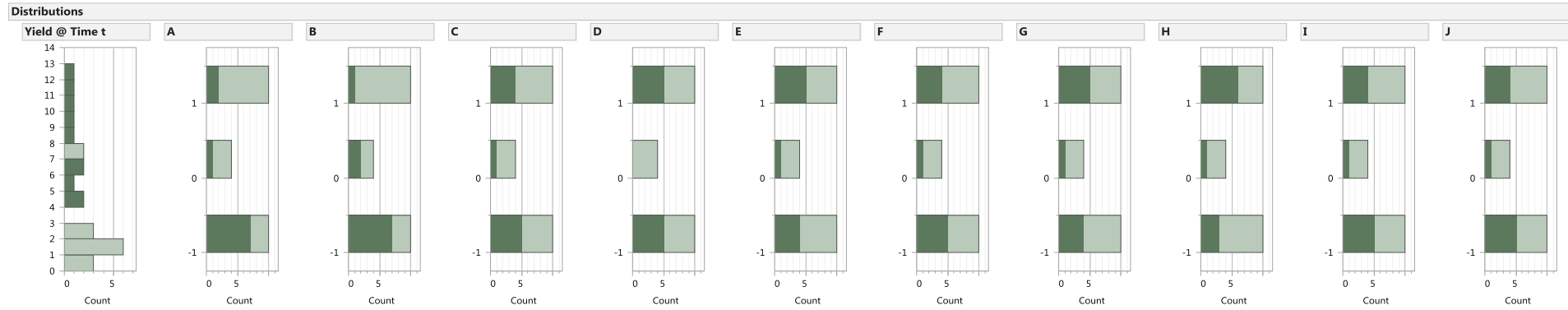
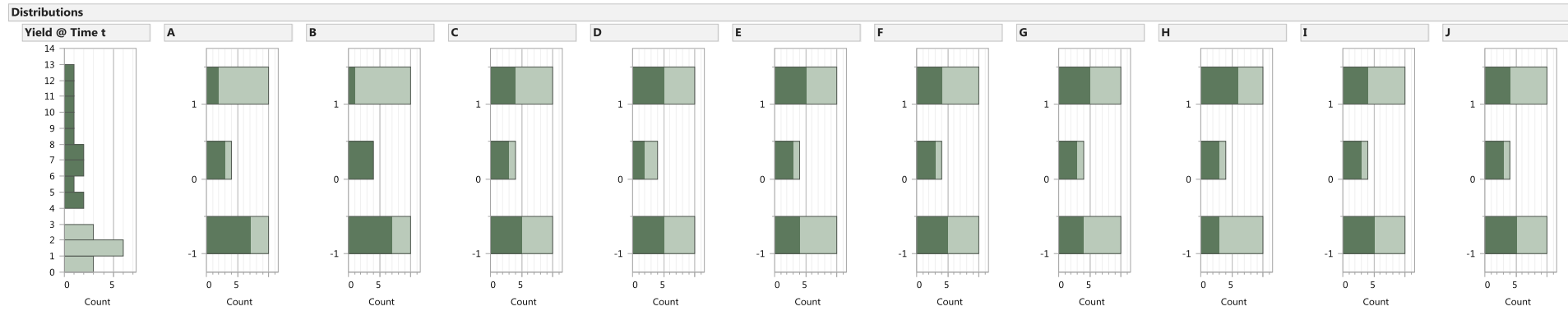
Scatterplot 3D



TOP: HIGHEST HALF OF YIELD DATA

MIDDLE: HIGHEST HALF OF YIELD MINUS 2 CENTER POINTS

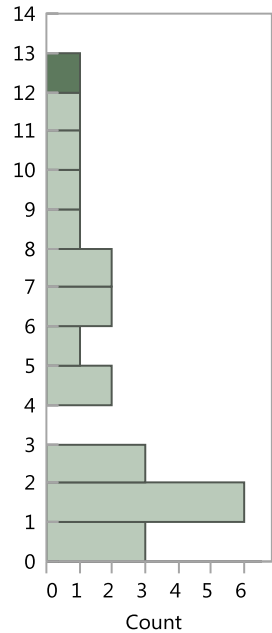
BOTTOM: LOWEST HALF OF YIELD DATA



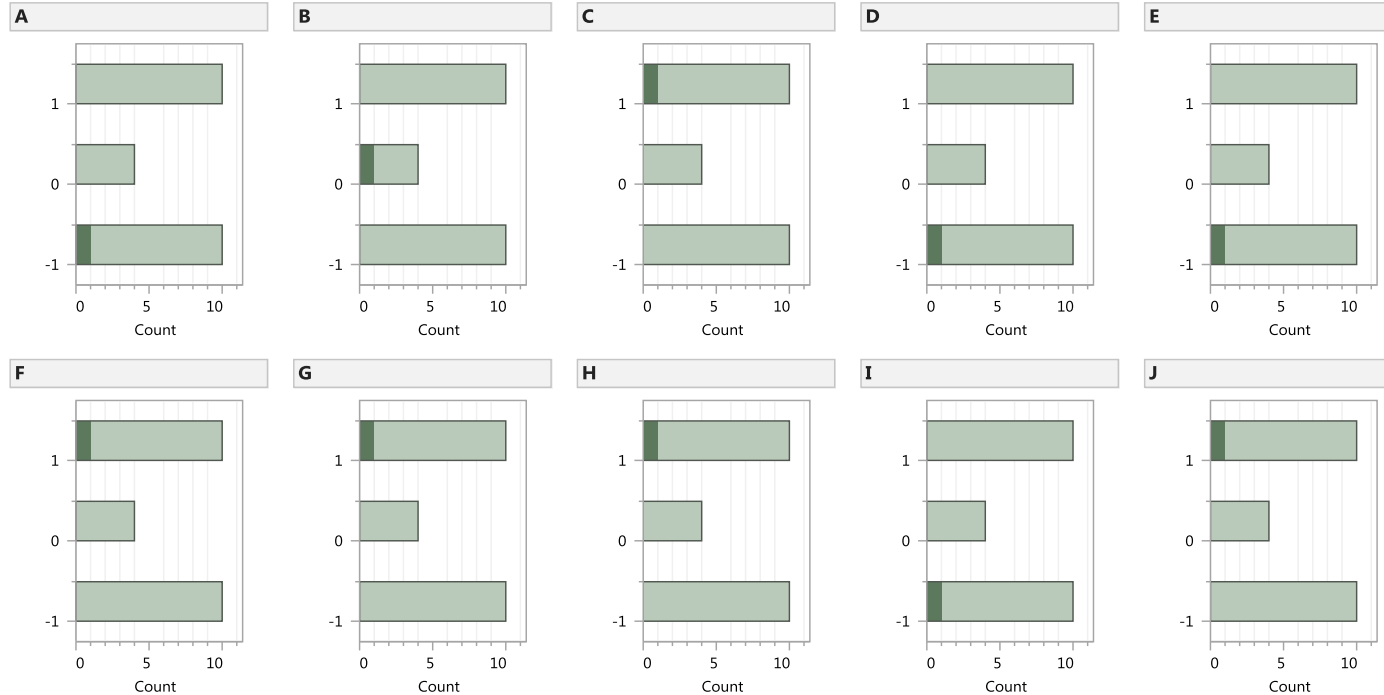
Settings of best observation of Yield = 12.96

Distributions

Yield @ Time t

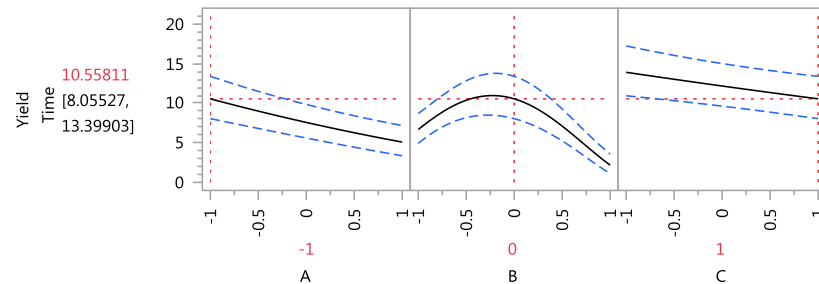


Distributions



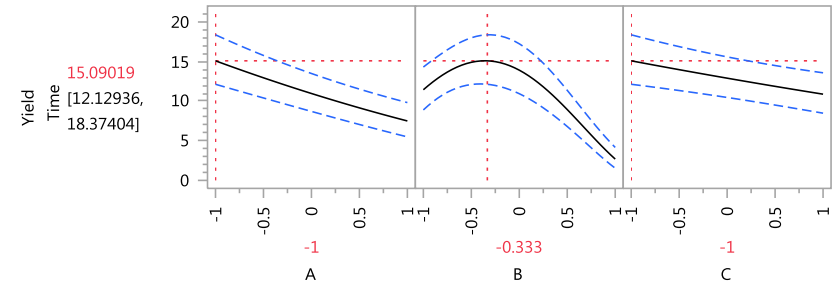
Prediction at settings of best observation

Prediction Profiler

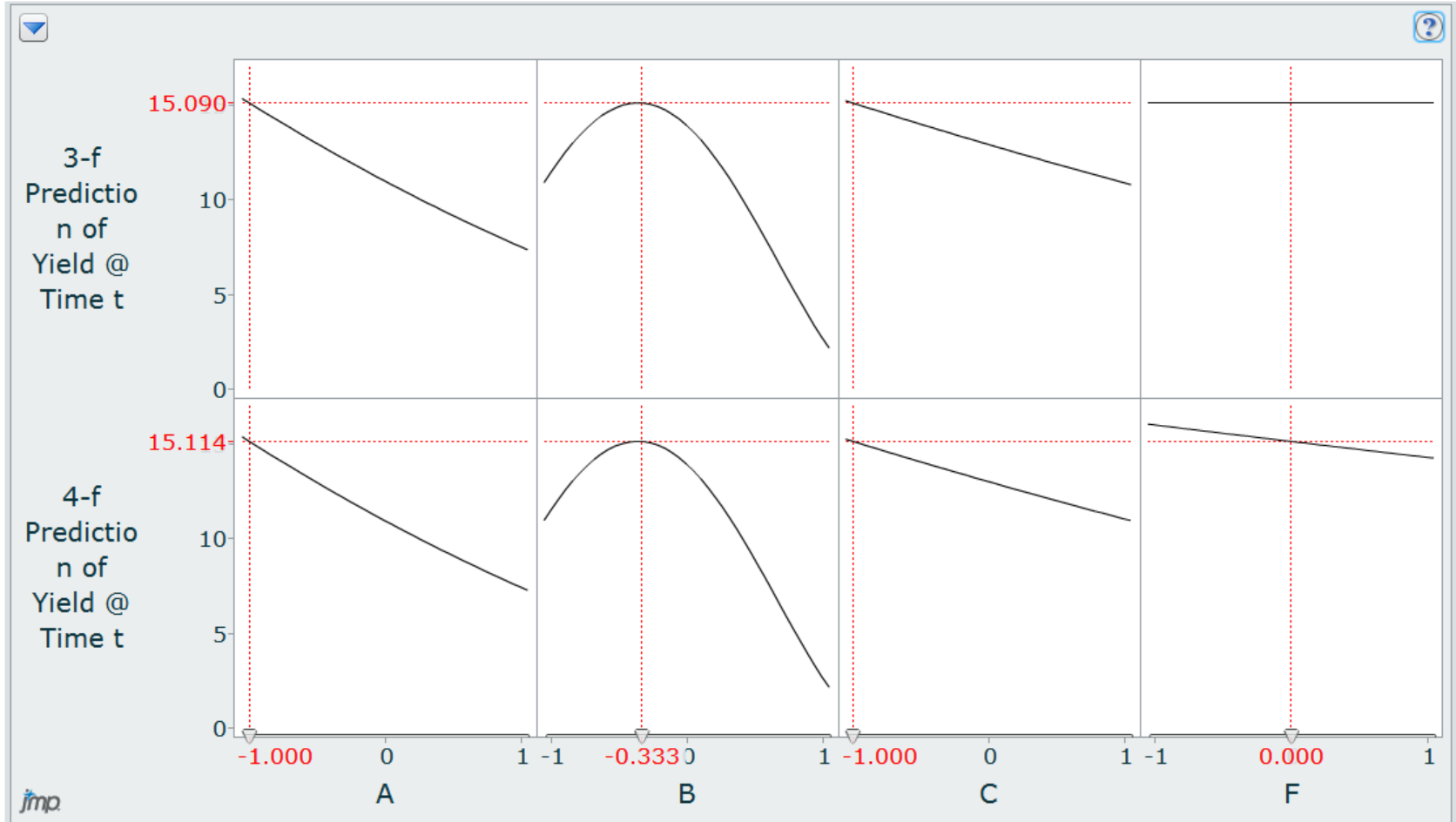


Prediction at best settings – run this checkpoint

Prediction Profiler



Predicting with Best 3-Factor and 4-Factor Model



AGGRESSIVE ANALYSES

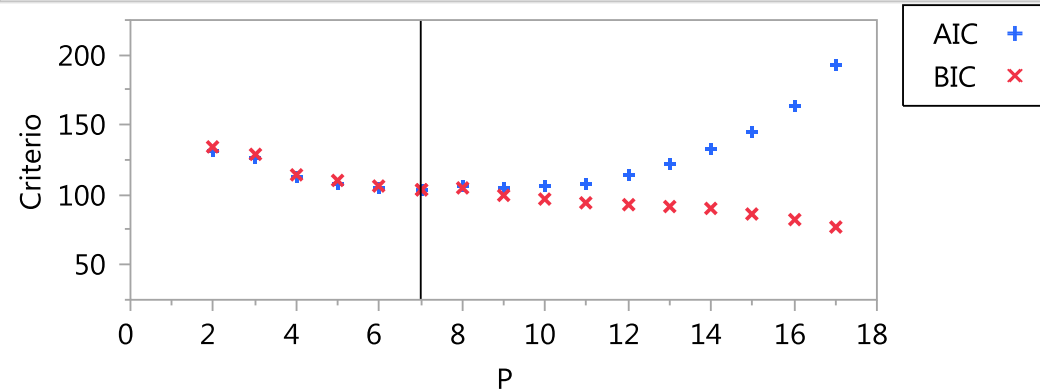
- Stepwise using full 10-factor, 66-term quadratic model with 24 observations and 4 checkpoints from a Definitive Screening Design
1 intercept + 10 ME + 10 SQ + 45 2FI (2-factor interactions)
 - Use AICc & BIC stopping criteria and pick “simpler model” – Occam’s razor
 - Use max K-Fold R-square as stopping rule to pick model (no checkpoints)
 - Use max validation R-square for checkpoints as stopping rule to pick model
 - Fit ALL possible models

USE MIN AIC OR BIC CRITERION AS STOPPING RULE

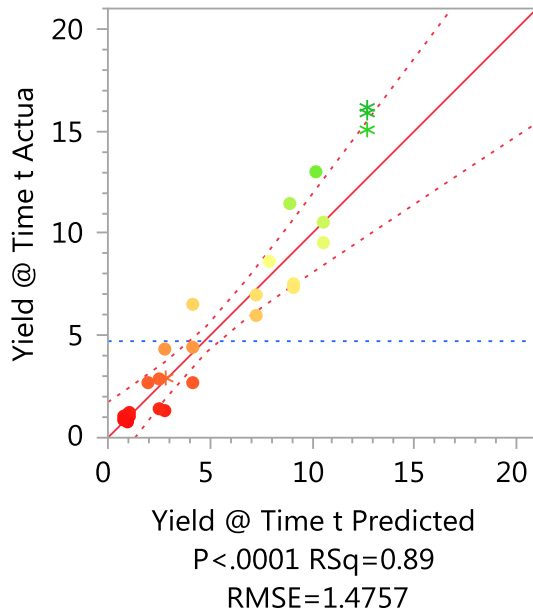
66 TERM QUADRATIC

RAW RESPONSE VALUES USED

Criterion History



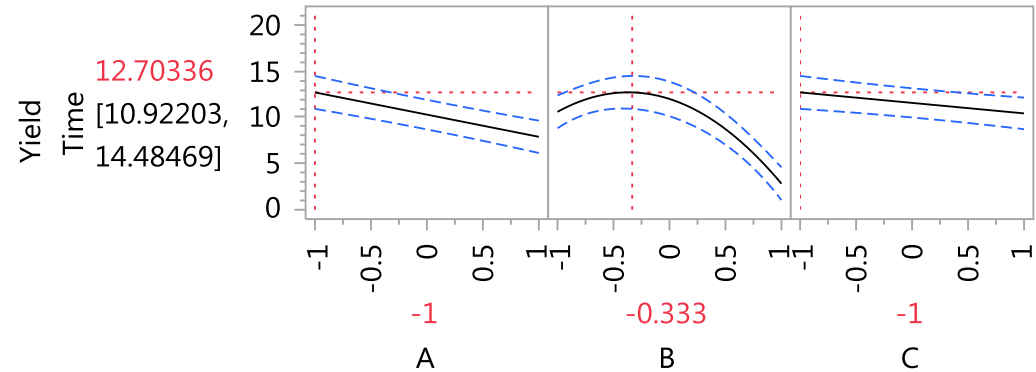
Actual by Predicted Plot



Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob > t
B*B	-5.282841	0.809809	-6.52	<.0001 *
A	-2.014167	0.333302	-6.05	<.0001 *
B	-1.979167	0.333302	-5.94	<.0001 *
A*B	1.1703157	0.349799	3.35	0.0038 *
C	-0.890833	0.333302	-2.68	0.0160 *
B*C	0.7369066	0.349799	2.11	0.0503

Prediction Profiler

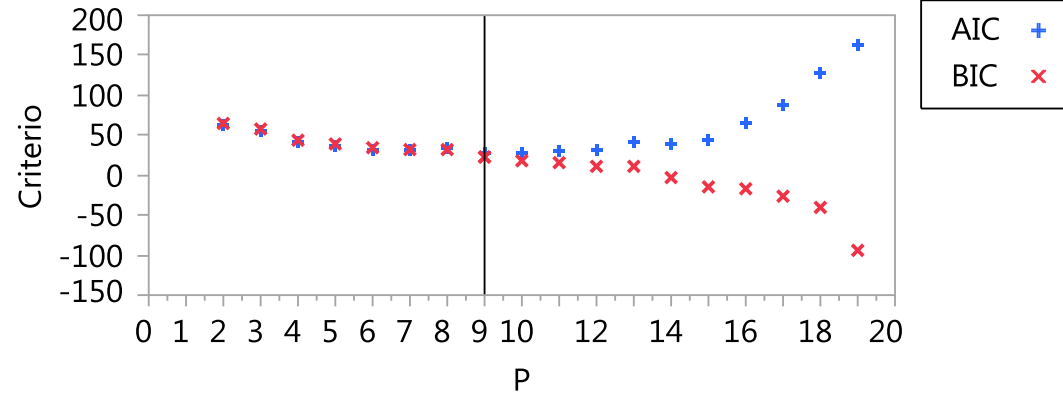


USE MIN AIC OR BIC CRITERION AS STOPPING RULE

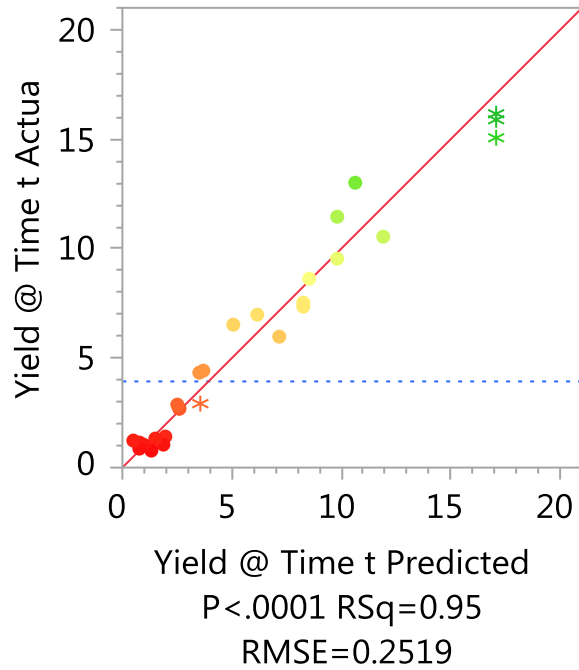
66 TERM QUADRATIC

TRANSFORMED VALUES USED

Criterion History



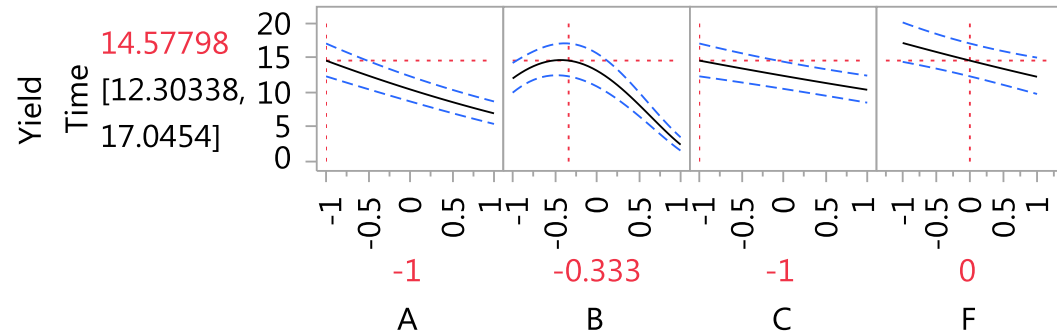
Actual by Predicted Plot



Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob > t
A	-0.505343	0.057053	-8.86	<.0001 *
B	-0.491041	0.057053	-8.61	<.0001 *
B*B	-1.111685	0.141981	-7.83	<.0001 *
A*B	0.253637	0.060121	4.22	0.0007 *
C	-0.231007	0.057053	-4.05	0.0010 *
B*C	0.2053297	0.061367	3.35	0.0044 *
C*F	0.2093075	0.063209	3.31	0.0047 *
F	-0.110087	0.057053	-1.93	0.0728

Prediction Profiler

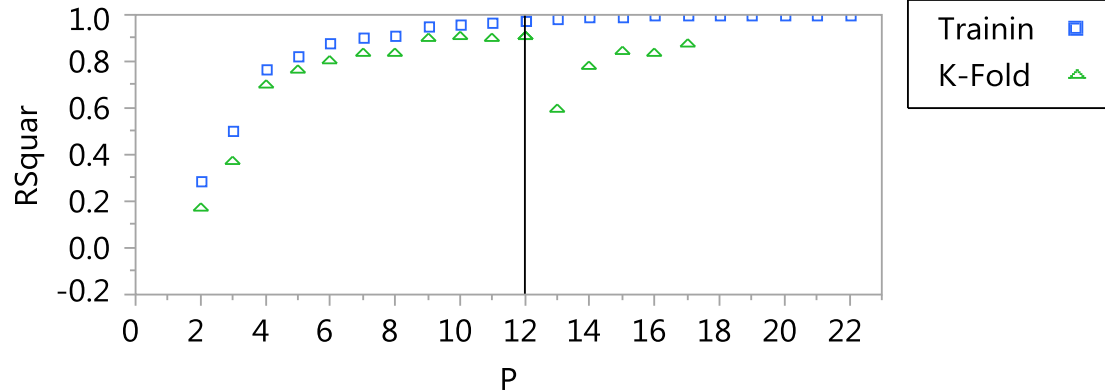


USE MAX K-FOLD R-SQUARE AS STOPPING RULE

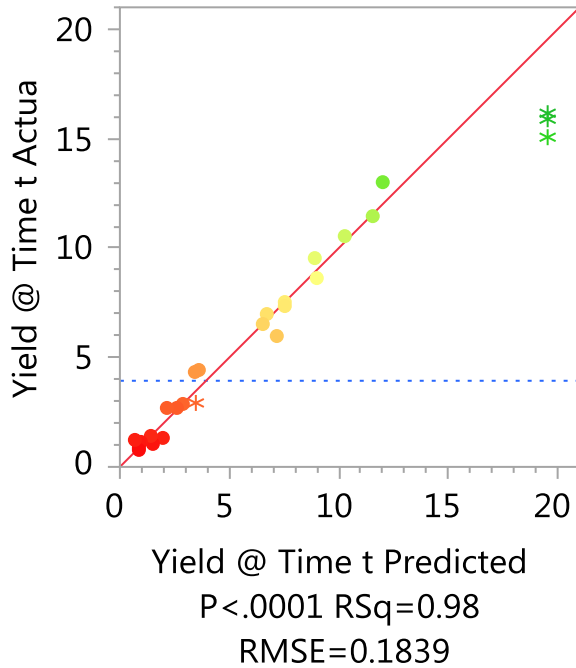
66 TERM QUADRATIC

TRANSFORMED VALUES USED

RSquare History



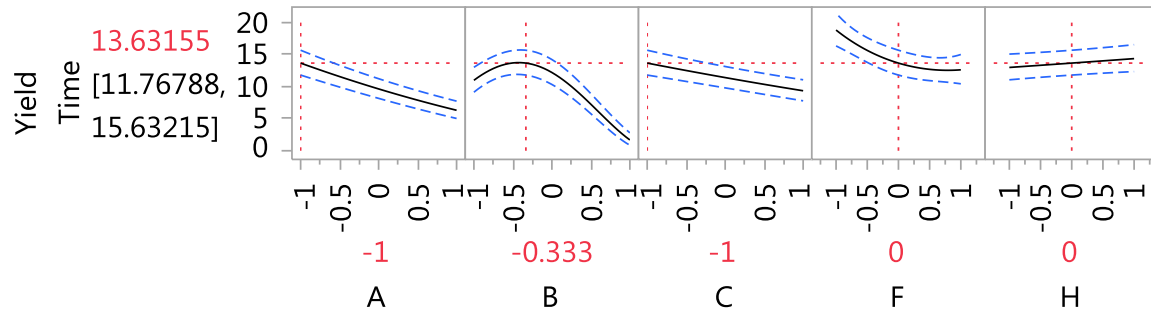
Actual by Predicted Plot



Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob > t
A	-0.498201	0.041762	-11.93	<.0001 *
B	-0.483899	0.041762	-11.59	<.0001 *
B*B	-1.184839	0.114991	-10.30	<.0001 *
A*B	0.2798015	0.045426	6.16	<.0001 *
C	-0.238149	0.041762	-5.70	<.0001 *
B*C	0.2427713	0.047097	5.15	0.0002 *
C*F	0.2349251	0.047559	4.94	0.0003 *
F	-0.117229	0.041762	-2.81	0.0158 *
B*F	0.1203014	0.0449	2.68	0.0201 *
H	0.0928467	0.041762	2.22	0.0462 *
F*F	0.2478009	0.120097	2.06	0.0614

Prediction Profiler

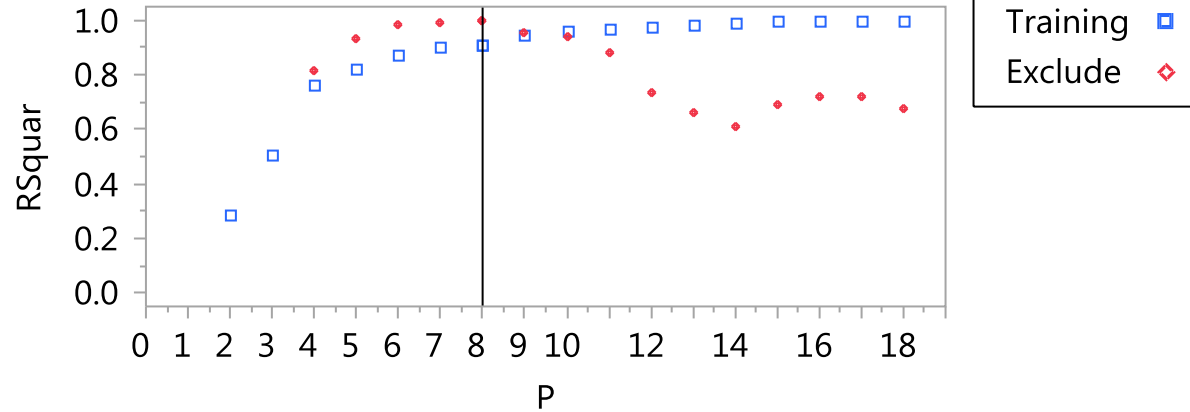


USE MAX VALIDATION R-SQUARE FOR 4 CHECKPOINTS AS STOPPING RULE

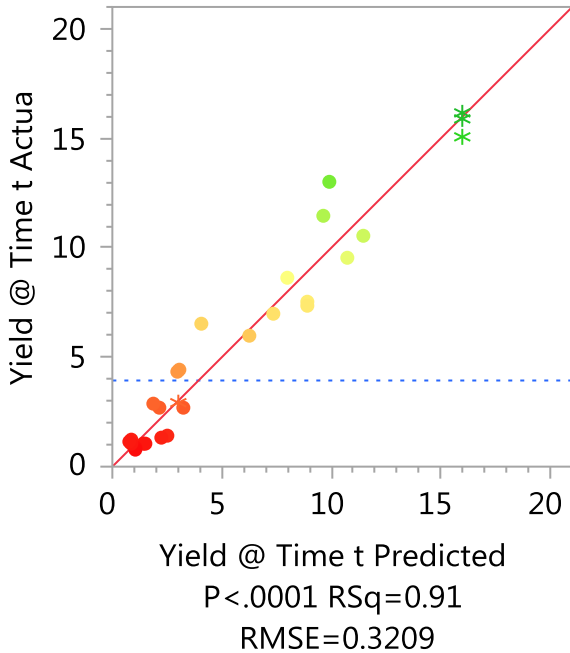
66 TERM QUADRATIC

TRANSFORMED VALUES USED

RSquare History



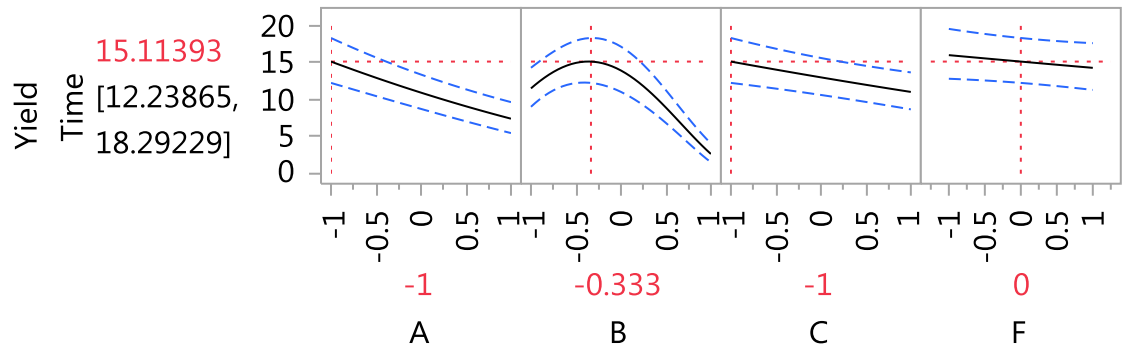
Actual by Predicted Plot



Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob > t
A	-0.505343	0.072679	-6.95	<.0001 *
B*B	-1.218717	0.17612	-6.92	<.0001 *
B	-0.491041	0.072679	-6.76	<.0001 *
C	-0.231007	0.072679	-3.18	0.0058 *
A*B	0.2306449	0.076075	3.03	0.0079 *
B*C	0.1585526	0.076075	2.08	0.0535
F	-0.110087	0.072679	-1.51	0.1494

Prediction Profiler



FIT ALL POSSIBLE MODELS UP TO 8 TERMS

- 1-term A
- 2-term B, B*B
- 3-term A, B, B*B
- 4-term A, B, C,
B*B
- 5-term A, B, C,
A*B, B*B
- 6-term A, B, C,
A*B, B*B, B*C
- 7-term A, B, C, G,
A*B, B*B, B*G
- 8-term A, B, C, G,
A*B, B*B, A*C,
B*G

CO2 Capture Process - Fit Stepwise 2 - JMP Pro

Stepwise Fit for Sqrt(Yield @ Time t)

All Possible Models

Model	Number	RSquare	RMSE	AICc	BIC
B,F,H	3	0.2934	0.8177	67.4058	69.9628
A,E,(E--0.1429)*(E--0.1429)	3	0.2922	0.8184	67.4456	70.0025
A,B,C,B*B	4	0.8260	0.4163	37.3830	39.5102
A,B,A*B,B*B	4	0.8169	0.4270	38.5990	40.7262
A,B,F,B*B	4	0.7835	0.4644	42.6270	44.7542

4-term

CO2 Capture Process - Fit Stepwise 2 - JMP Pro

Stepwise Fit for Sqrt(Yield @ Time t)

All Possible Models

Model	Number	RSquare	RMSE	AICc	BIC
A,B,H,(H-0.14286)*(H-0.14286)	4	0.5358	0.6800	60.9300	63.0571
A,B,D,A*A	4	0.5352	0.6804	60.9587	63.0858
A,B,C,A*B,B*B	5	0.8768	0.3599	33.1552	34.4016
A,B,C,B*B,B*C	5	0.8504	0.3966	37.8124	39.0588
A,B,C,F,B*B	5	0.8385	0.4121	39.6548	40.9011

5-term

CO2 Capture Process - Fit Stepwise 2 - JMP Pro

Stepwise Fit for Sqrt(Yield @ Time t)

All Possible Models

Model	Number	RSquare	RMSE	AICc	BIC
A,B,E,A*B,A*(E--0.1429)	5	0.6402	0.6150	58.8712	60.1175
A,B,E,F,A*(E--0.1429)	5	0.6401	0.6151	58.8813	60.1277
A,B,C,A*B,B*B,B*C	6	0.9004	0.3329	32.6422	32.4667
A,B,C,F,A*B,B*B	6	0.8893	0.3511	35.1906	35.0150
A,B,C,H,A*B,B*B	6	0.8840	0.3593	36.3016	36.1261

6-term

CO2 Capture Process - Fit Stepwise 2 - JMP Pro

Stepwise Fit for Sqrt(Yield @ Time t)

All Possible Models

Model	Number	RSquare	RMSE	AICc	BIC
A,B,C,D,B*B,A*D	6	0.8348	0.4289	44.7940	44.6185
A,B,E,F,A*B,B*B	6	0.8347	0.4290	44.8087	44.6331
A,B,C,G,A*B,B*B,B*(G-0.14286)	7	0.9239	0.3000	31.4479	29.1933
A,B,C,E,B*B,A*(E--0.1429),B*(E--0.1429)	7	0.9145	0.3180	34.2381	31.9835
A,B,C,F,A*B,B*B,B*C	7	0.9129	0.3209	34.6833	32.4287

7-term

ALL ANALYSES RANK FACTORS A, B & C AS TOP 3

FACTOR F APPEARS TO BE MOST LIKELY FOURTH FACTOR

- Linear terms only – fourth factor is F
 - Linear + Squared terms – fourth factor is D
 - Stepwise with min AICc stopping rule – fourth factor is F
 - Stepwise with max K-Fold R-Square stopping rule – fourth factor is F
 - Stepwise with max Validation R-Square as stopping rule – fourth factor is F
 - All possible models – fourth factor is G
-
- When D & F are in same 5-factor (with A, B, & C) stepwise model, D drops out
 - When G & F are in same 5-factor (with A, B, & C) stepwise model, G drops out
 - When D & G are in same 5-factor (with A, B, & C) stepwise model, both drop out
-
- There is an important difference between saying, “*Factor F has no effect.*” and, “*Given the amount of data taken an effect for factor F was not detected.*”
-
- Augmenting design to support 6-factor quadratic model in A, B, C, D, F & G will
 - help resolve the relative contributions of D, F & G
 - increase the power for all – but especially - the squared terms

Decision Trees

Overview



Decision Trees

- Also known as Recursive Partitioning, CHAID, CART
- Models are a series of nested IF() statements, where each condition in the IF() statement can be viewed as a separate branch in a tree.
- Branches are chosen so that the difference in the average response (or average response rate) between paired branches is maximized.
 - For all factors bin factor values or levels into two buckets such that the means of the two buckets are as far apart as possible.
 - Split on factor with the biggest difference in bucket means.
- Tree models are “grown” by adding more branches to the tree so the more of the variability in the response is explained by the model

Decision Tree Step-by-Step


Goal is to predict “Rejects” & “Accepts”

Overall Accept Rate is 84.44%

Overall Reject Rate is 15.56%

RSquare

0.000

All Rows		
		
Count	G^2	
90	77.800668	
Level	Rate	Prob
Accep	0.8444	0.8444
Reject	0.1556	0.1556

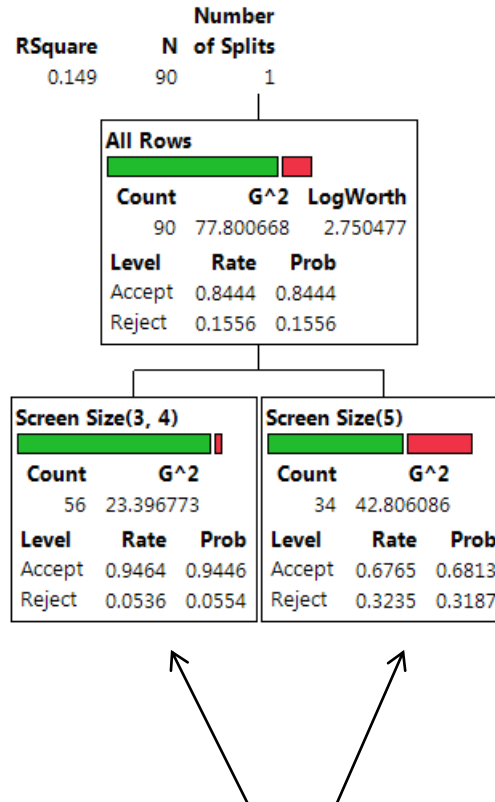
Candidates

Term	Candidate	G^2	LogWorth	Cut Point
API Particle Size	4.04050319	0.986886932	Small, Large	
Mill Time	10.63219688	1.912625603	11	
Screen Size	11.59780917	> 2.750476973	3,4	
MgSt Supplier	1.99715970	0.802459554	Jones Inc	
Lactose Supplier	1.07597470	0.523458492	James Ind	
Sugar Supplier	3.99502860	1.340705011	Sour	
Talc Supplier	0.00000000	0.000000000	Rough	
Blend Time	2.46622023	0.066048548	15.887	
Blend Speed	6.86574102	0.717212865	60.772	
Compressor	0.00153207	0.013776004	COMPRESS	
Force	7.53188562	0.855446810	24.691	
Coating Supplie	0.82675321	0.217072294	Mac	
Coating Viscositi	4.66879353	0.322714711	96.413	
Inlet Temp	7.28399996	0.803171227	106.39	
Exhaust Temp	7.17119361	0.779703315	68.592	
Spray Rate	15.01998363	< 2.736639439	403.26	
Atom. Pressure	3.36570749	0.149475063	58.787	

Candidate “X’s”

- Search through each of these
- Examine Splits for each unique level in each X
- Find Split that maximizes “LogWorth”
 - Will find split that maximizes difference in proportions of the target variable

Decision Tree Step-by-Step



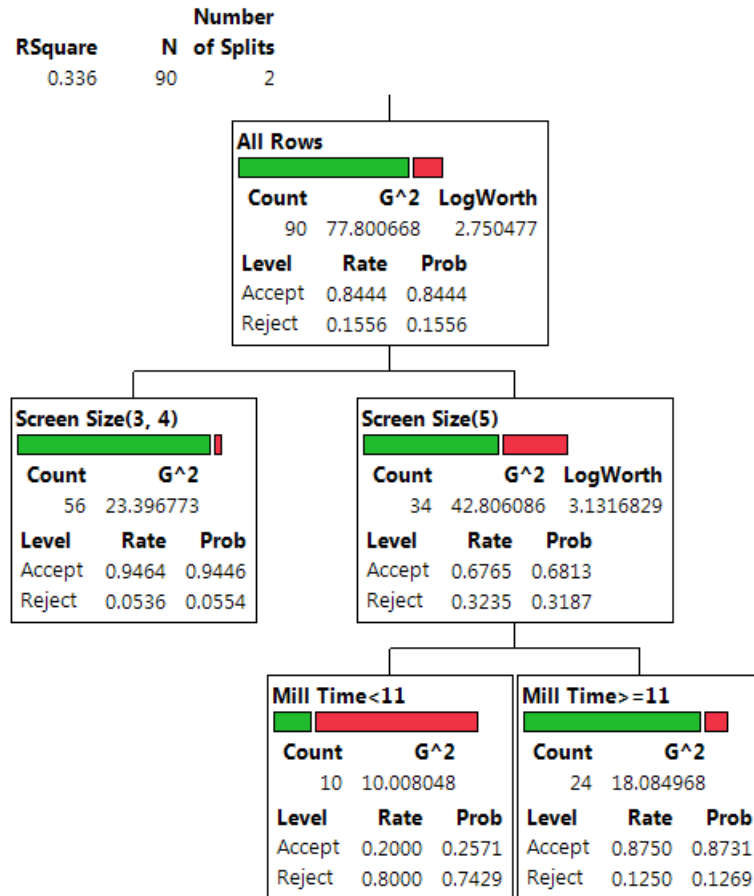
1st Split:

Optimal Split Screen Size 3 & 4 vs. Screen Size 5

Notice the difference in the rates in each branch of the tree

Repeat "Split Search" across both "Partitions" of the data. Find optimal split across both branches.

Decision Tree (Step by Step)



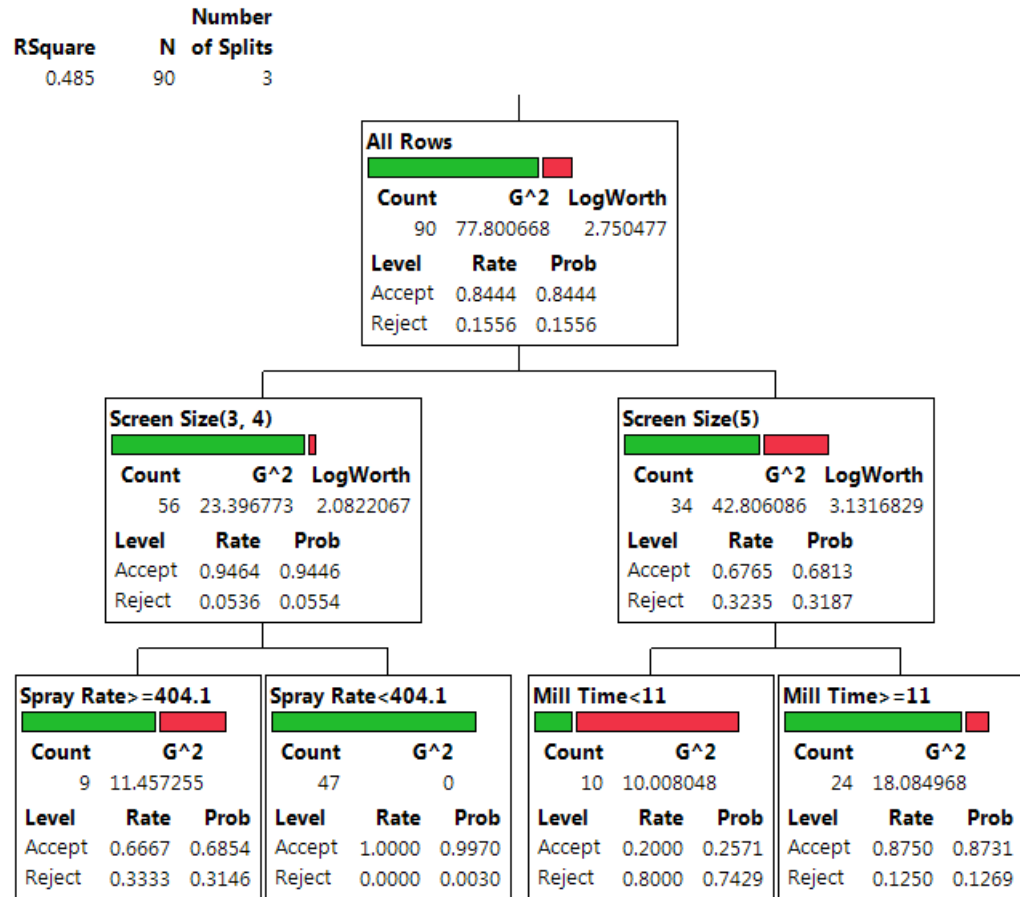
2nd split on Mill Time
(< 11 vs. >= 11)

Notice variation in
proportion of "1" in each
branch

Decision Tree (Step by Step)

3rd split on Spray Rate
(>= 404.1 vs. < 404.1))

Notice variation in
proportion of "1" in each
branch

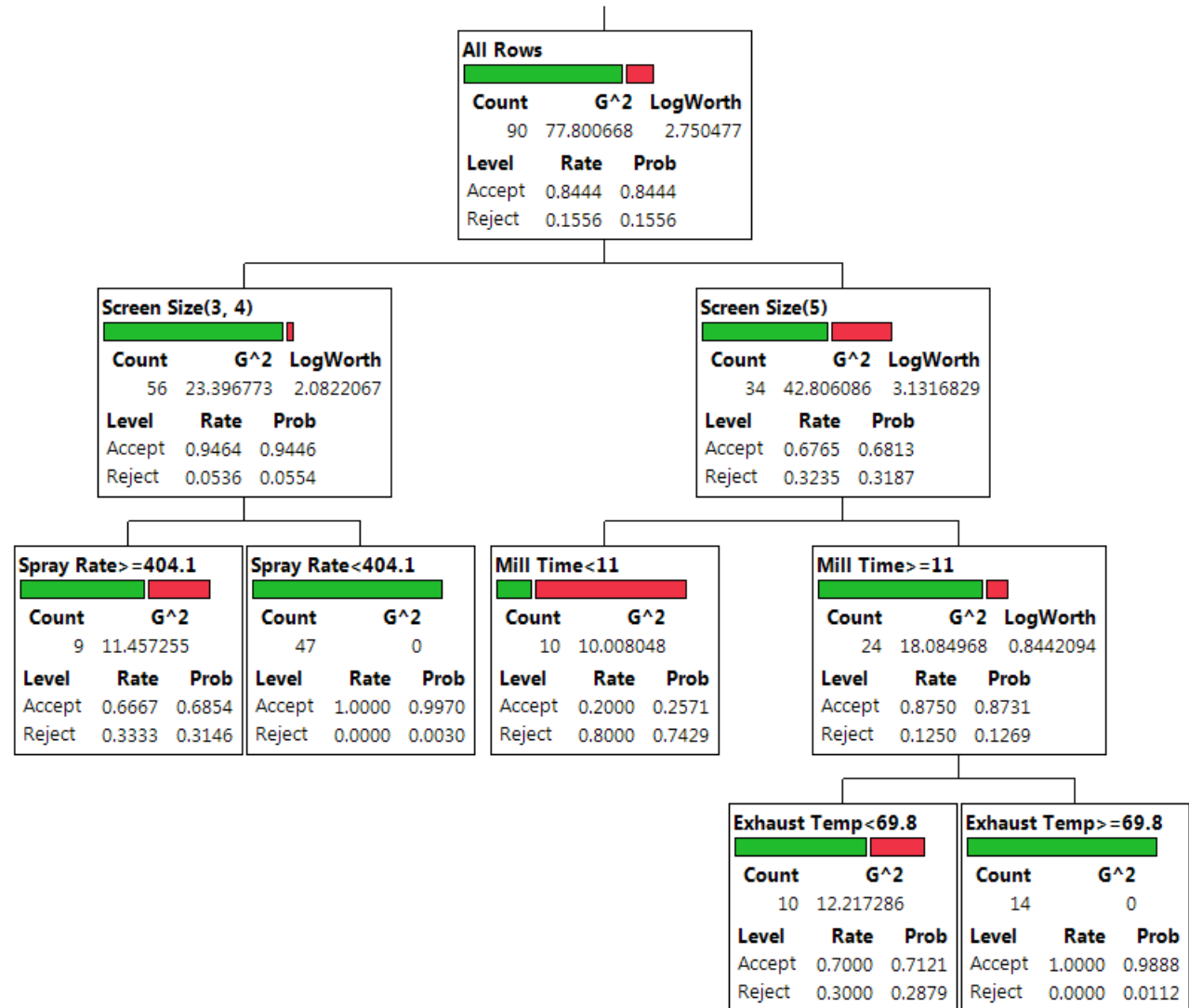


Decision Tree (Step by Step)

4th split on Exhaust Temp
(< 69.8 vs. ≥ 69.8)

Notice variation in
proportion of "1" in each
branch

RSquare 0.557
Number of Splits 90 4

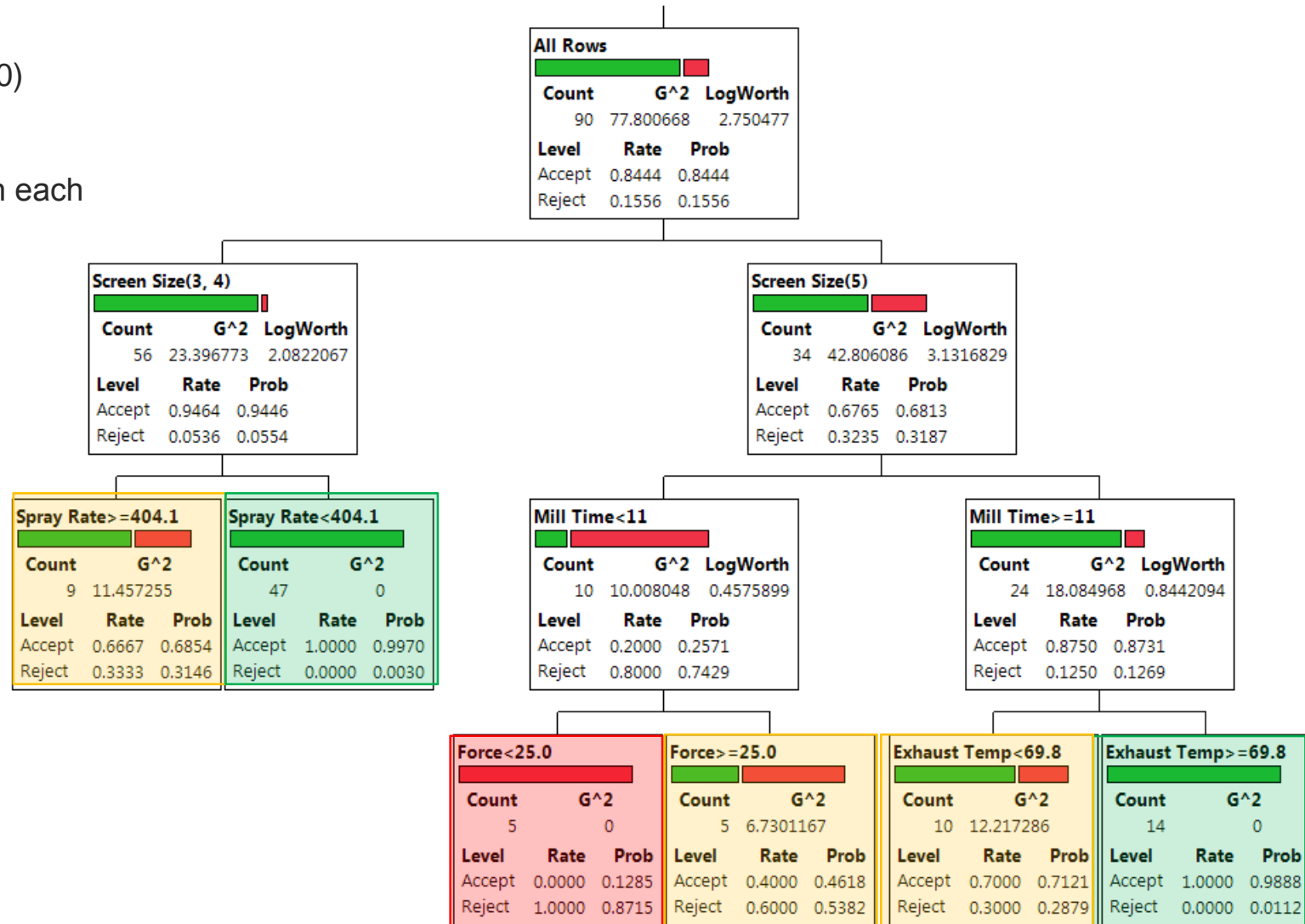


Decision Tree (Step by Step)

RSquare 0.583
Number of Splits 90 5

5th split on Force
(< 25.0 vs. ≥ 25.0)

Notice variation in proportion of "1" in each branch

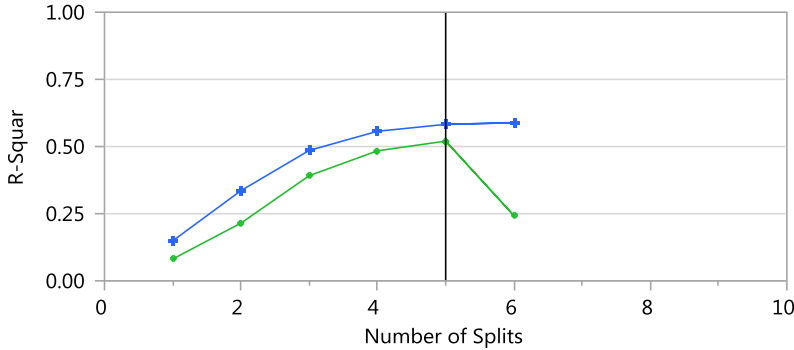


Decision Tree (Step by Step)

Crossvalidation

k-fold	-2LogLike	RSquare
5 Folde	37.3288048	0.5202
Overa	30.4046577	0.5825

Split History

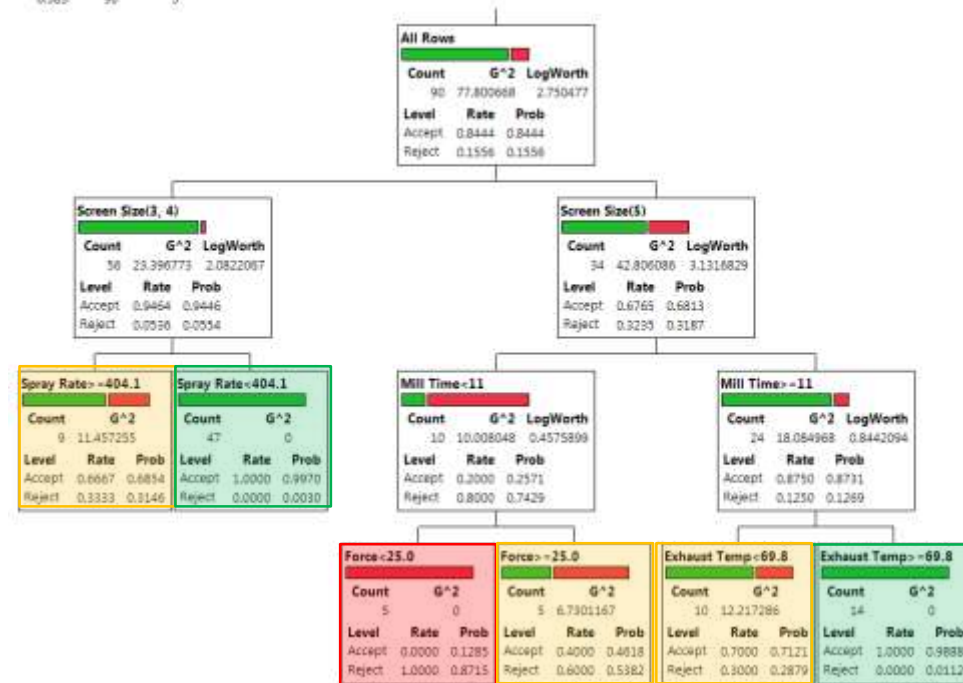


K-Fold in Green

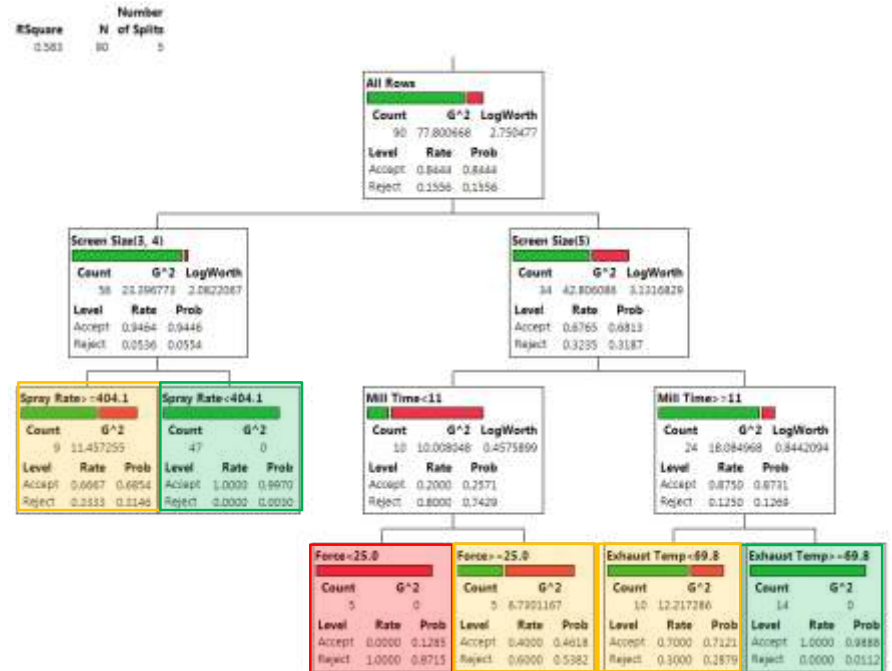
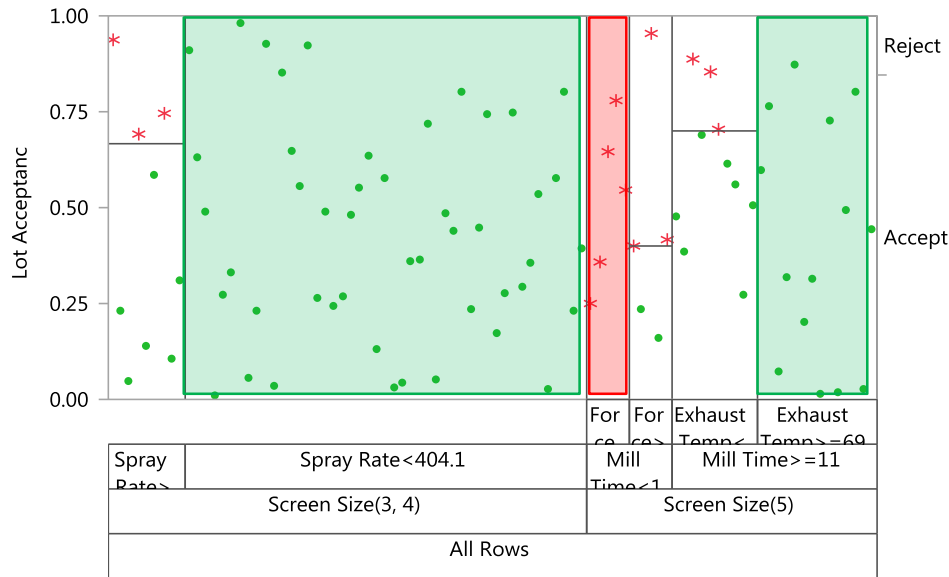
Column Contributions

Term	Number of Splits	G ²	Portion
Mill Time	1	14.7130695	0.3104
Spray Rate	1	11.9395178	0.2519
Screen Size	1	11.5978092	0.2447
Exhaust Temp	1	5.8676817	0.1238
Force	1	3.2779318	0.0692
API Particle Size	0	0	0.0000
MgSt Supplier	0	0	0.0000
Lactose Supplier	0	0	0.0000
Sugar Supplier	0	0	0.0000
Talc Supplier	0	0	0.0000
Blend Time	0	0	0.0000
Blend Speed	0	0	0.0000
Compressor	0	0	0.0000
Coating Supplie	0	0	0.0000
Coating Viscosit	0	0	0.0000
Inlet Temp	0	0	0.0000
Atom. Pressure	0	0	0.0000

RSquare: 0.5825
Number N of Splits: 5



Decision Tree (Step by Step)



Leaf Report

Response Prob

Leaf Label

Leaf Label	Accept	Reject
Screen Size(3, 4)&Spray Rate>=404.1	0.6854	0.3146
Screen Size(3, 4)&Spray Rate<404.1	0.9970	0.0030
Screen Size(5)&Mill Time<11&Force<25.0	0.1285	0.8715
Screen Size(5)&Mill Time<11&Force>=25.0	0.4618	0.5382
Screen Size(5)&Mill Time>=11&Exhaust Temp<69.8	0.7121	0.2879
Screen Size(5)&Mill Time>=11&Exhaust Temp>=69.8	0.9888	0.0112

Response Counts

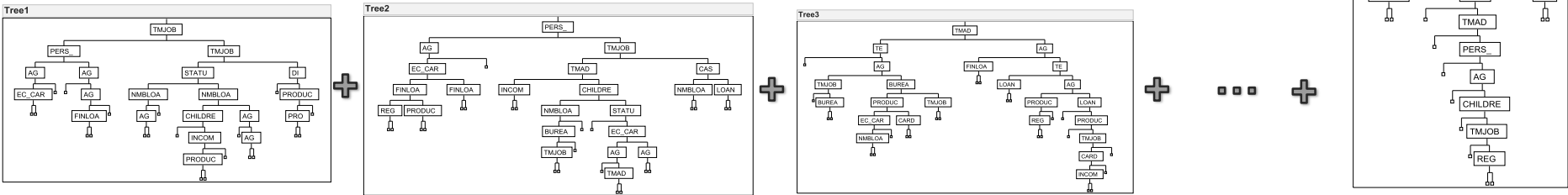
Leaf Label

Leaf Label	Accept	Reject
Screen Size(3, 4)&Spray Rate>=404.1	6	3
Screen Size(3, 4)&Spray Rate<404.1	47	0
Screen Size(5)&Mill Time<11&Force<25.0	0	5
Screen Size(5)&Mill Time<11&Force>=25.0	2	3
Screen Size(5)&Mill Time>=11&Exhaust Temp<69.8	7	3
Screen Size(5)&Mill Time>=11&Exhaust Temp>=69.8	14	0

Bootstrap Forest

- Bootstrap Forest
 - For each tree, take a random sample of the predictor variables (***with replacement***) – e.g. pick half of the variables. Build out a decision tree on that subset of variables.
 - Make many trees and average their predictions (bagging)
 - This is also known as a random forest technique
 - Works very well on wide tables.
- Can be used for ***both*** predictive modeling and variable selection.
- Allows for dominant variables to be excluded from some trees giving less dominant – but still important – variables a chance to be selected.
- Valuable approach for screening variables for use with other modeling methods – e.g. neural networks.

Average the Trees in the Forest



100

Bootstrap Forest Model

Columns Contributions for Bootstrap Forest Analysis of Cyber Data – Variable Selection w/44 Factors – 3 of which were Random Data!

Column Contributions

Term	Number of Splits	G ²	Portion
service	450	10603400.8	0.2831
dst_bytes	382	5308498.33	0.1417
src_bytes	820	4771327.16	0.1274
count	337	2700247.28	0.0721
dst_host_srv_count	528	1990388.66	0.0531
dst_host_diff_srv_rate	415	1575488.06	0.0421
flag	168	1153015.42	0.0308
srv_count	238	1115688.05	0.0298
dst_host_serror_rate	175	1060259.19	0.0283
duration	276	991351.909	0.0265
dst_host_count	499	714300.159	0.0191
dst_host_same_src_port_rat	389	616742.634	0.0165
hot	159	535399.996	0.0143
same_srv_rate	103	422795.794	0.0113
dst_host_same_srv_rate	334	421699.768	0.0113
diff_srv_rate	145	382986.204	0.0102
serror_rate	65	365667.013	0.0098
dst_host_rerror_rate	233	318445.492	0.0085
dst_host_srv_serror_rate	117	308717.284	0.0082
logged_in	40	305603.637	0.0082
srv_serror_rate	30	219339.913	0.0059
root_shell	32	203921.266	0.0054
dst_host_srv_diff_host_rate	253	196905.011	0.0053
Random Uniform	228	195145.878	0.0052
dst_host_srv_rerror_rate	81	153228.513	0.0041
protocol_type	53	152857.046	0.0041
is_guest_login	12	137886.036	0.0037
Random Normal	194	110253.474	0.0029
num_compromised	39	76703.4706	0.0020
num_file_creations	20	75279.6937	0.0020
wrong_fragment	29	72313.7688	0.0019
rerror_rate	45	59525.1111	0.0016
num_root	23	41990.5367	0.0011
Random Integer	146	21117.3276	0.0006
srv_diff_host_rate	33	17448.0232	0.0005
num_failed_logins	7	17407.5895	0.0005
srv_rerror_rate	30	16080.2873	0.0004
num_access_files	11	11528.8834	0.0003
num_shells	11	8067.77994	0.0002
urgent	4	3131.15585	0.0001
su_attempted	1	42.7170189	0.0000
land	0	0	0.0000
num_outbound_cmds	0	0	0.0000
is_host_login	0	0	0.0000

Column Contributions

Term	Number of Splits	G ²	Portion
service	450	10603400.8	0.2831
dst_bytes	382	5308498.33	0.1417
src_bytes	820	4771327.16	0.1274
count	337	2700247.28	0.0721
dst_host_srv_count	528	1990388.66	0.0531
dst_host_diff_srv_rate	415	1575488.06	0.0421
flag	168	1153015.42	0.0308
srv_count	238	1115688.05	0.0298
dst_host_serror_rate	175	1060259.19	0.0283
duration	276	991351.909	0.0265
dst_host_count	499	714300.159	0.0191
dst_host_same_src_port_rat	389	616742.634	0.0165
hot	159	535399.996	0.0143
same_srv_rate	103	422795.794	0.0113
dst_host_same_srv_rate	334	421699.768	0.0113

Top 11 of 44

Model Validation-Set Summaries

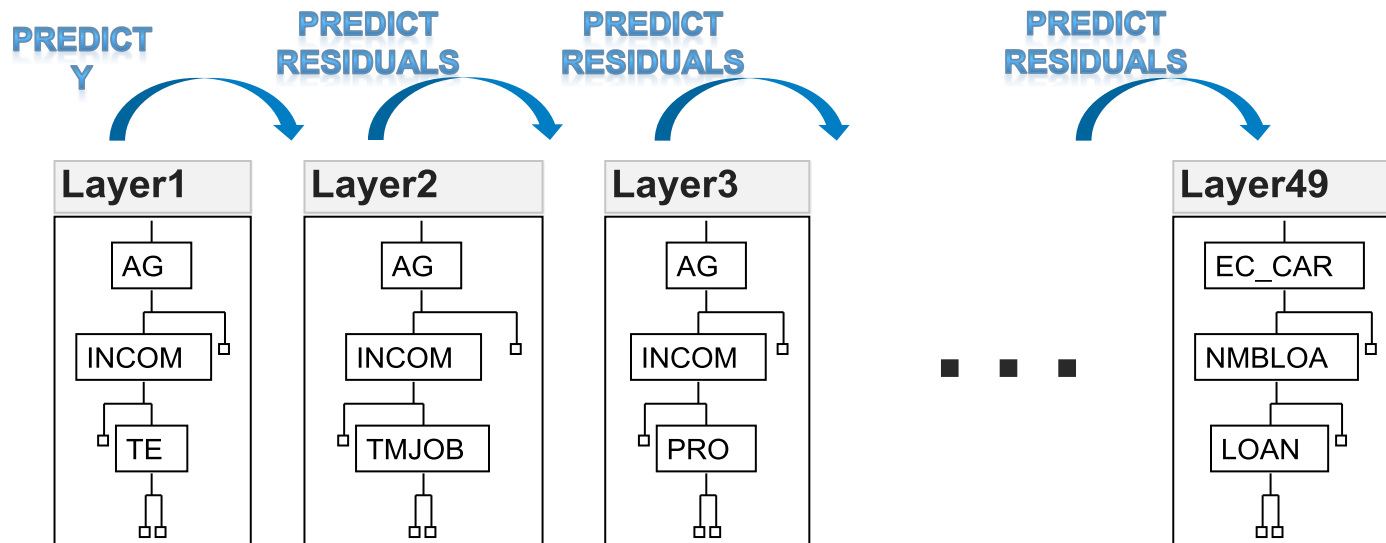
The fit below was the best of these models fit.

	N Terms	N Trees	Entropy RSquare	Misclassification Rate	Avg -Log p	Avg Abs RMS Error	Avg Abs Error
11	11	200	0.9786	0.0040	0.0336	0.0856	0.0279
14	14	53	0.9811	0.0040	0.0297	0.0816	0.0243
18	18	48	0.9831	0.0039	0.0265	0.0770	0.0215
Random Uniform	228	195145.878					0.0052

Boosted Tree

- Beginning with the first tree (layer) build a small simple tree.
- From the residuals of the first tree, build another small simple tree.
- This continues until a specified number of layers has been fit, or a determination has been made that adding successive layers doesn't improve the fit of the model.
- The final model is the weighted accumulation of all of the model layers.

Boosted Tree Illustrated



Models M1

M2

M3

M49

Final Model

$$M = M1 + \varepsilon \cdot M2 + \varepsilon \cdot M3 + \dots + \varepsilon \cdot M49$$

ε is the learning rate

Case Study: Decision Trees

Already Previewed



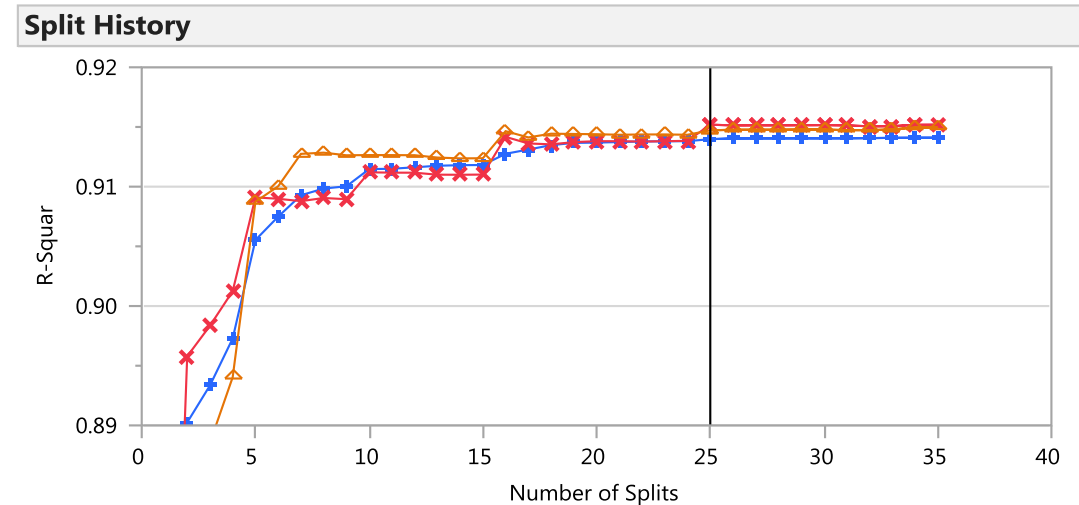
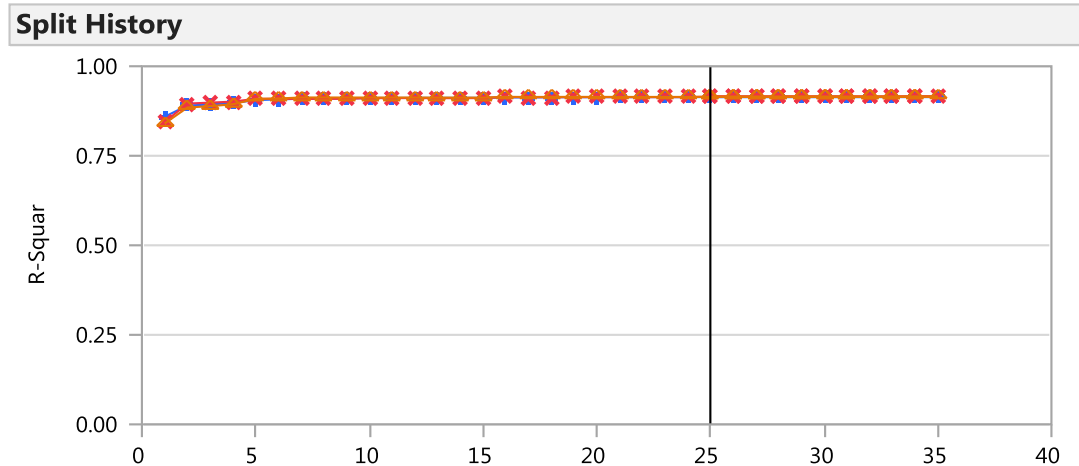
THE
POWER
TO KNOW®

Surrogate Modeling of a Computer Simulation - Helicopter Surveillance – Identifying Insurgents

- 2009 International Data Farming Workshop - IDFW21, Lisbon, Portugal
- Largely German team (6 of 8) – their simulation
- 6500 simulations run overnight on cluster in Frankfurt
 - 65 unique combinations of 6 factors (each factor at 65 levels)
 - each case had 97 to 100 replications (lost a few)
- Response = Proportion of Insurgents Identified = *PropIdentINS* Data bounded between 0 and 1
- Explore data visually first
- Fit many different models – “Train, Validate (Tune), Test” 60/20/20 subsets
- Compare Actual vs. Predicted for Test Set

Honest Assessment Approach Using Train, Validate (Tune), and Test Subsets

Train, Validate, Test
R-Square vs. #Splits
Decision Tree Model
(6458 rows of
simulation data for
helicopter flying
surveillance.)



Validation Data in Red
Test Data in Orange

Similar Results for Different Decision Tree Methods for Helicopter Simulation Data

Column Contributions

DECISION TREE

Term	Number of Splits	SS	Portion
InsurgentCamouflag	6	553.514843	0.9819
TigerHeight	4	5.23947275	0.0093
ConvoySpeed	6	2.66493548	0.0047
TigerSpeedRelative	3	1.58563474	0.0028
num_INS2_AK47	4	0.66588349	0.0012
Tiger1_Distance	2	0.06006294	0.0001

	RSquare	RMSE	N
Training	0.914	0.1170276	3874
Validatio	0.915	0.1132339	1292
Test	0.915	0.1147605	1292

Column Contributions

BOOTSTRAP FOREST

Term	Number of Splits	SS	Portion
InsurgentCamouflag	50	1328.61688	0.9338
TigerSpeedRelative	36	31.1106368	0.0219
Tiger1_Distance	48	28.8649626	0.0203
TigerHeight	48	22.2499023	0.0156
num_INS2_AK47	40	8.36974799	0.0059
ConvoySpeed	32	3.6452873	0.0026

	RSquare	RMSE	N
Training	0.914	0.1170121	3874
Validatio	0.915	0.1132062	1292
Test	0.915	0.1148662	1292

Column Contributions

BOOSTED TREE

Term	Number of Splits	SS	Portion
InsurgentCamouflag	101	2922.60546	0.9864
TigerHeight	33	34.7333591	0.0117
TigerSpeedRelative	1	2.2917136	0.0008
ConvoySpeed	8	2.16230243	0.0007
num_INS2_AK47	5	0.75525236	0.0003
Tiger1_Distance	2	0.23141595	0.0001

	RSquare	RMSE	N
Training	0.913	0.1179096	3874
Validatio	0.915	0.1136506	1292
Test	0.913	0.1159973	1292

Neural Networks

Overview

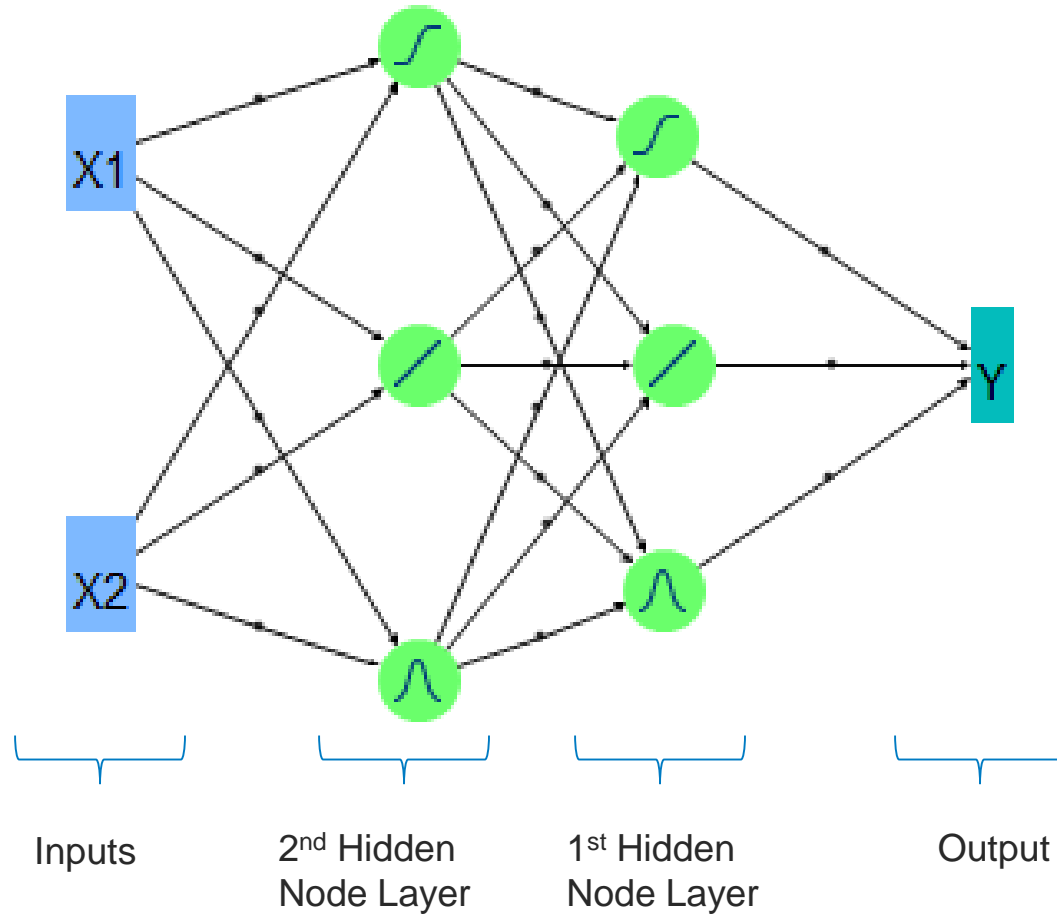


THE
POWER
TO KNOW.®

Neural Networks

- Neural Networks are highly flexible nonlinear models.
- A neural network can be viewed as a weighted sum of nonlinear functions applied to linear models.
 - The nonlinear functions are called activation functions. Each function is considered a (hidden) node.
 - The nonlinear functions are grouped in layers. There may be more than one layer.
- Consider a generic example where there is a response Y and two predictors $X1$ and $X2$. An example type of neural network that can be fit to this data is given in the diagram that follows

Example Neural Network Diagram



Neural Networks

- Big Picture
 - Can model:
 - » Continuous and categorical predictors
 - » Continuous and categorical responses
 - » Multiple responses (simultaneously)
 - Can be numerically challenging and time consuming to fit
 - NN models are very prone to overfitting if you are not careful
 - » There are several ways to help prevent overfitting
 - » Some type of validation is required

Case Study: Neural Networks

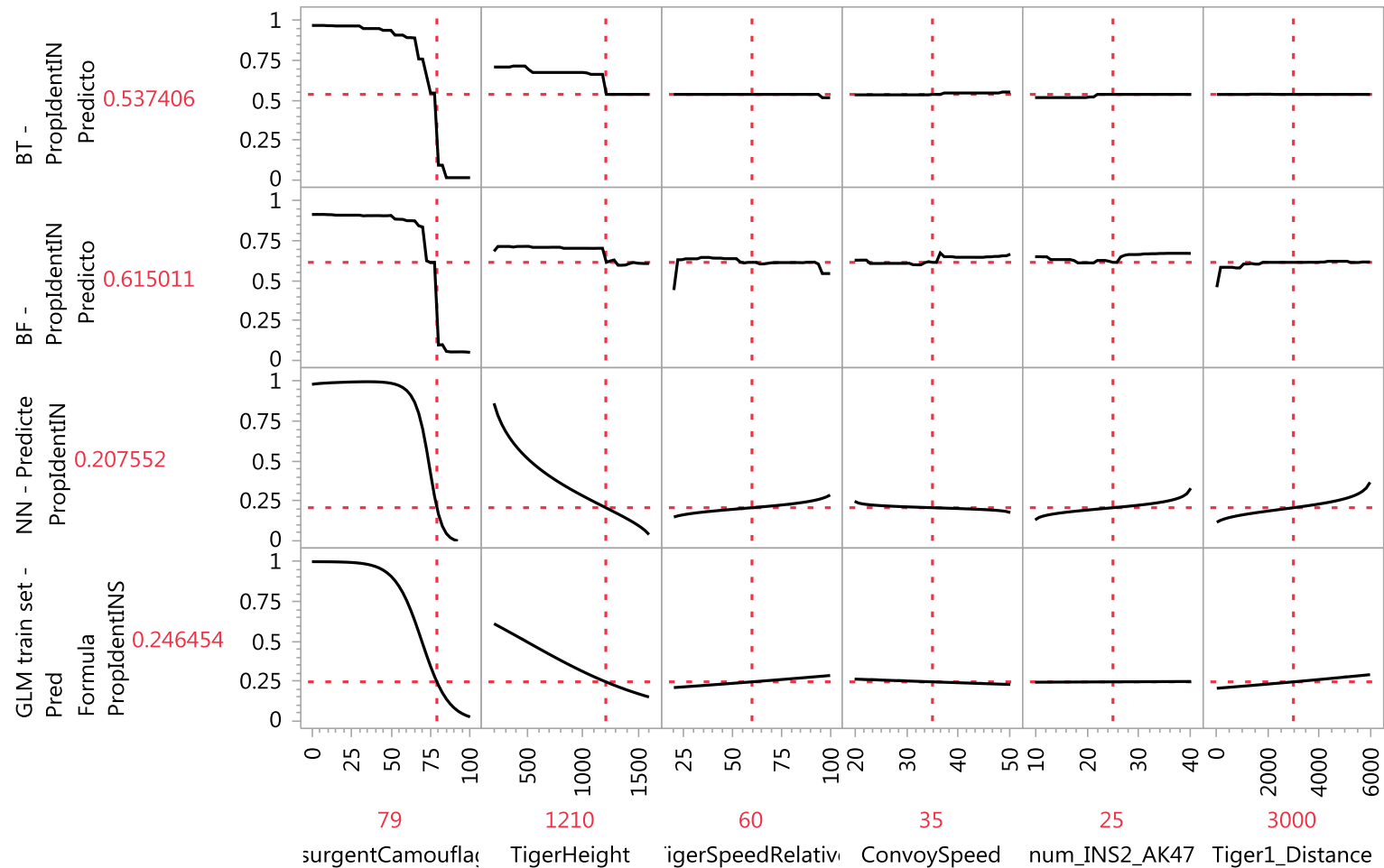
Helicopter Surveillance



THE
POWER
TO KNOW®

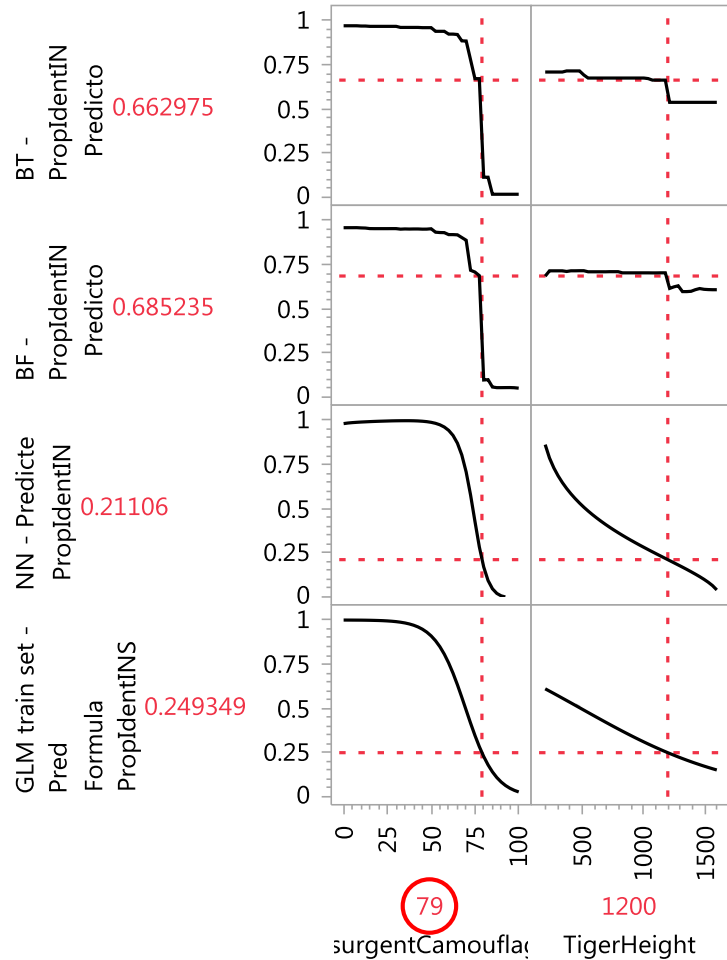
Compare Several Models – top 2 are decision tree variants bottom two are “smoother” models - Neural Net and GLM

Prediction Profiler

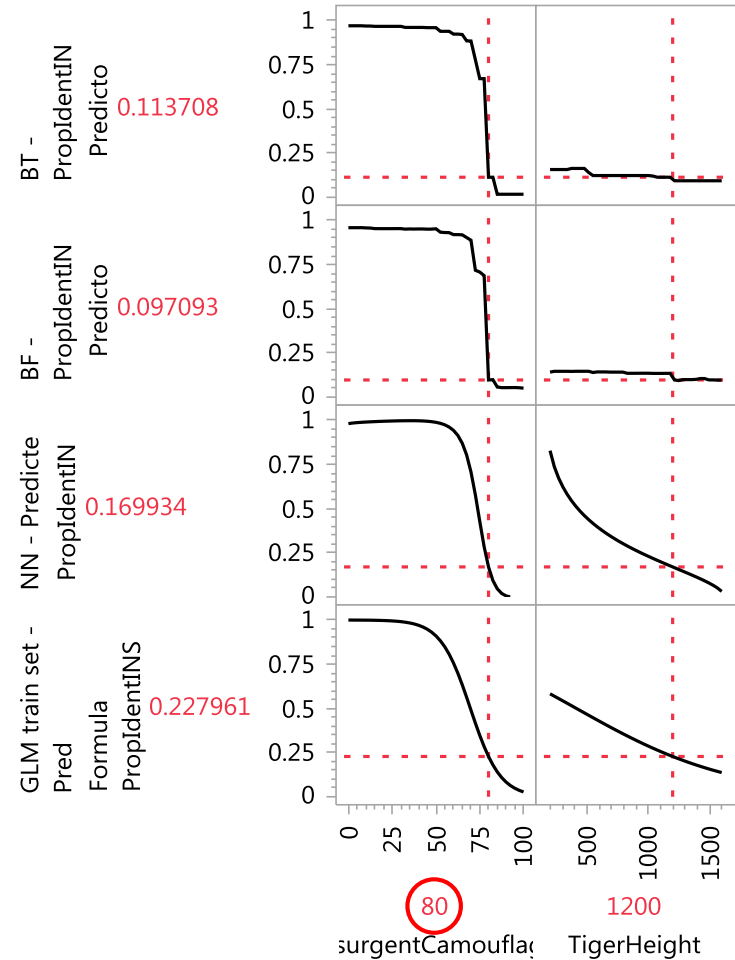


Change Camouflage from 79 to 80 and Decision Tree Predictions Drop by 6X – Talk to Developer?

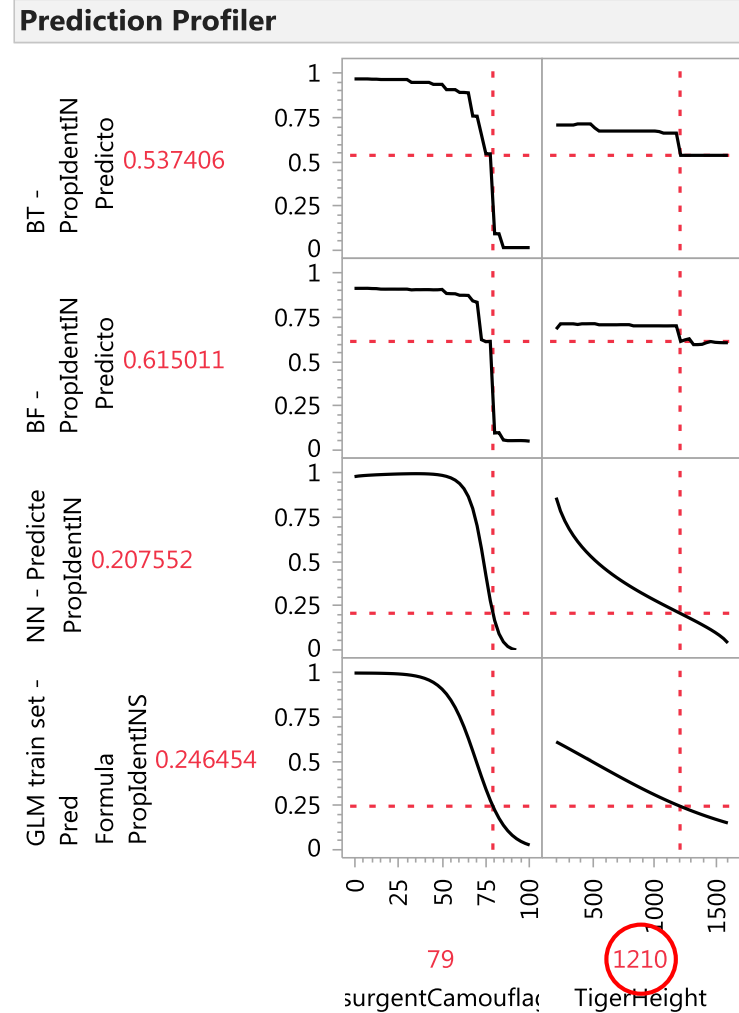
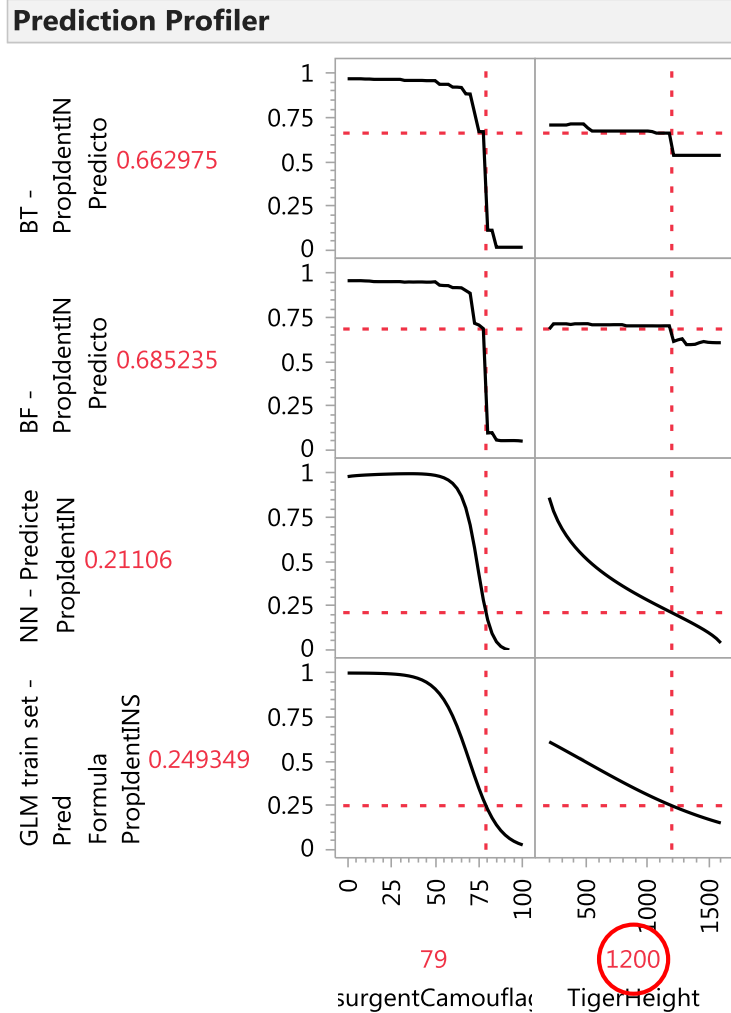
Prediction Profiler



Prediction Profiler

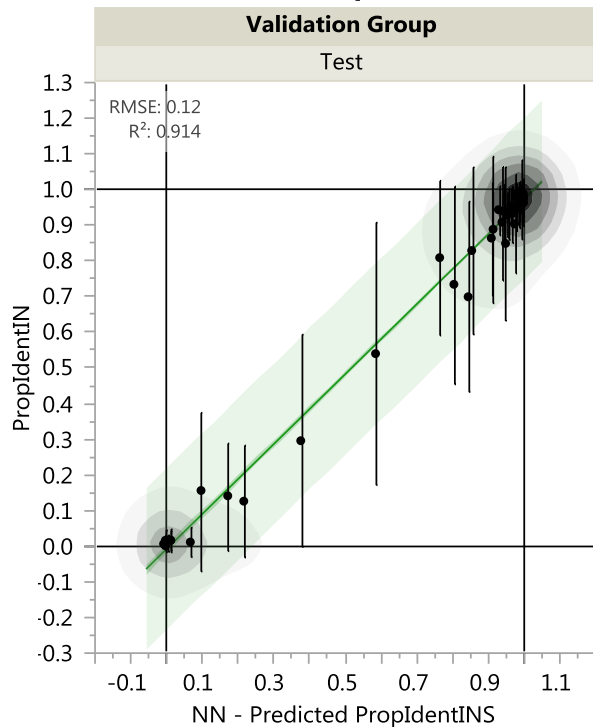


Change Tiger Height from 1200 to 1210 and Decision Tree Predictions Drop by 10% to 20%! – Plausible?

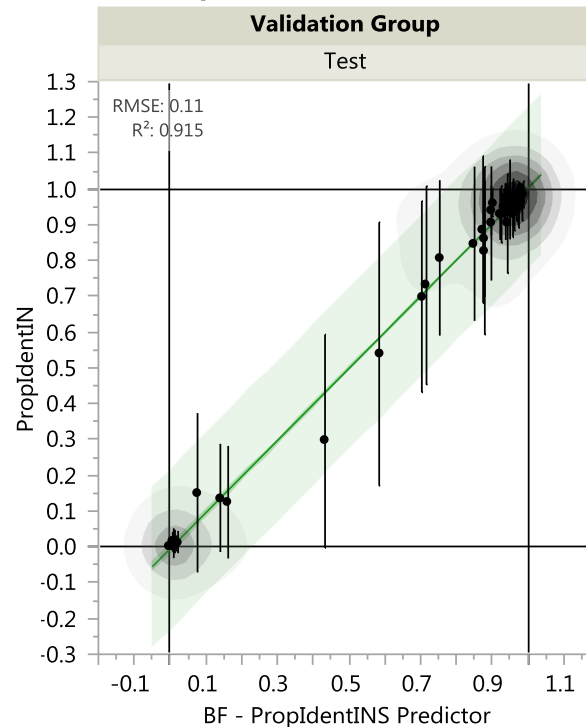


Actual vs. Predicted Plots for Test Data Neural Net, Bootstrap Forest and GLM Models

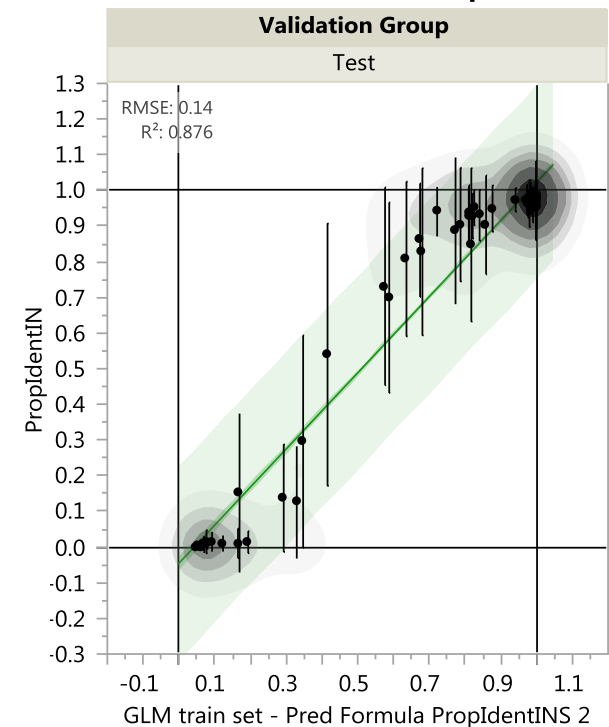
PropIdentINS & Mean(PropIdentINS) vs.
NN - Predicted PropIdentINS



PropIdentINS & Mean(PropIdentINS) vs. BF
- PropIdentINS Predictor



PropIdentINS & Mean(PropIdentINS) vs.
GLM train set - Pred Formula PropIdentINS



Model Comparison

Finding the Most Useful Model



Choosing the Best Model

- In many situations you would try many different types of modeling methods
- Even within each modeling method, there are options to create different models
 - In Stepwise, the base/full model specification can be varied
 - In Bootstrap Forest, the number of trees and number of terms sample per split
 - In Boosted Tree, the learning rate, number of layers, and base tree size
 - In Neural, the specification of the model, as well as the use of boosting
- So how can you choose the “best”, most useful model?

The Importance of the Test Set

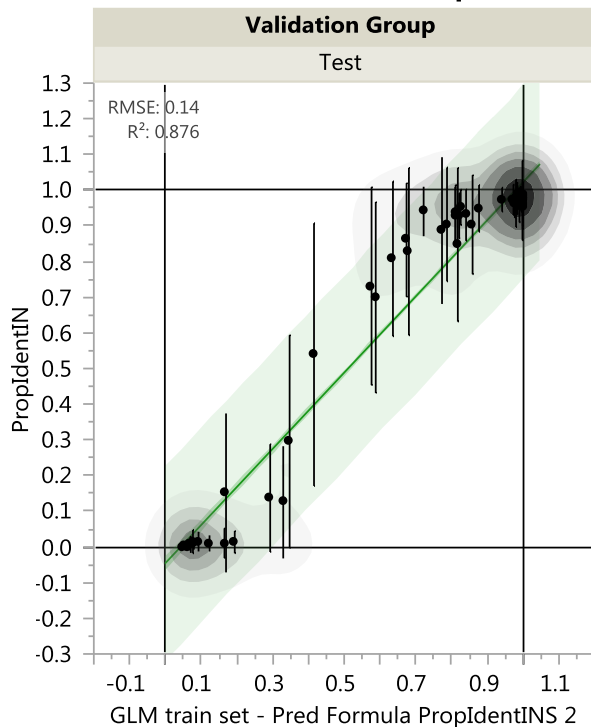
- One of the most important uses of having a training, validation, AND **test set** is that you can use the test set to assess each model on the same basis.
- Using the test set allows you to compare competing models on the basis of model quality metrics
 - R^2
 - Misclassification Rate
 - Actual vs. Prediction (Confusion Matrix)
 - ROC (Receiver Operating Characteristics) Curves and AUC (Area Under Curve – of ROC Curve)

Measures of Fit for PropIdentINS

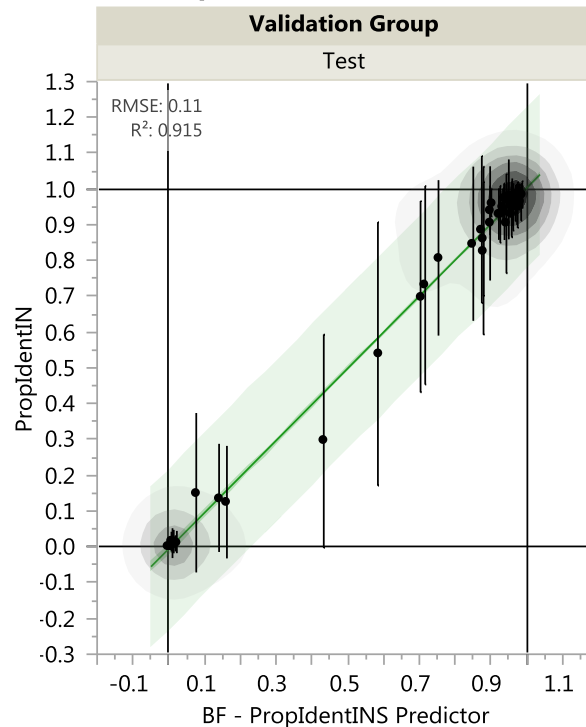
Metrics for Just the Test Subset

Predictor	Creator	.2.4.6.8	RSquare	RASE	AAE	Freq
GLM ALL Data Pred Formula PropIdentINS			0.8736	0.1397	0.0917	1292
Partition K-Fold PropIdentINS Predictor	Partition		0.9172	0.1131	0.0595	1292
BF - PropIdentINS Predictor			0.9149	0.1147	0.0609	1292
BT - PropIdentINS Predictor			0.9130	0.1159	0.0619	1292
NN Single Layer 33% Predicted PropIdentIN	Neural		0.9069	0.1199	0.0560	1292
NN - Predicted PropIdentINS			0.9105	0.1176	0.0570	1292
Probability(PropIdentINS=1)	Fit Generalize		0.8719	0.1407	0.0925	1292

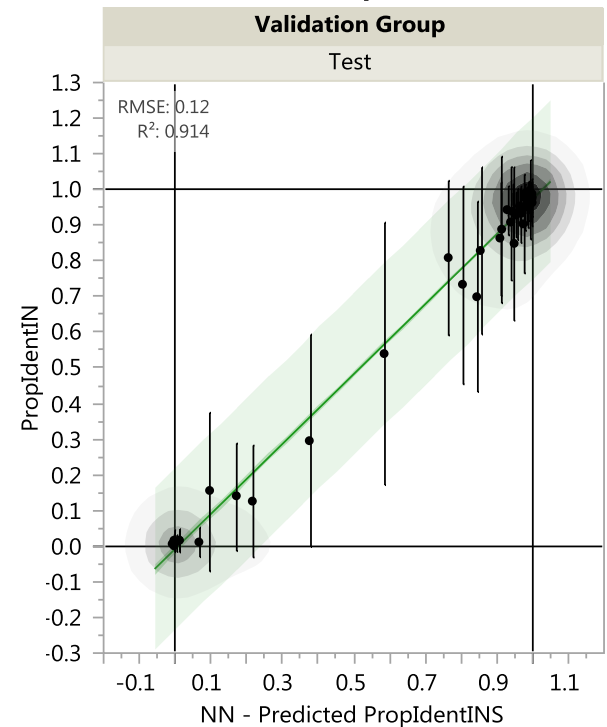
PropIdentINS & Mean(PropIdentINS) vs. GLM train set - Pred Formula PropIdentIN



PropIdentINS & Mean(PropIdentINS) vs. BF - PropIdentINS Predictor



PropIdentINS & Mean(PropIdentINS) vs. NN - Predicted PropIdentINS



Outline

- Case Study 1 Preview
- Introduction to Modeling
- Honest Assessment Method to *Prevent Overfitting*
- Regression and Model Selection
- Case Study 2
- Decision Trees
- Neural Models
- Model Comparison



THE
POWER
TO KNOW.

Thanks.
Questions or comments?

tom.donnelly@jmp.com