

On the Use of Machine Learning in the Semiconductor Industry: Examples and Case Studies

Theresa L. Utlaut

Logic Technology Development
Intel Corporation
Hillsboro, OR
theresa.l.utlaut@intel.com

Kevin C. Anderson

F11 Yield Engineering Dept.
Intel Corporation
Rio Rancho, NM
kevin.c.anderson@intel.com

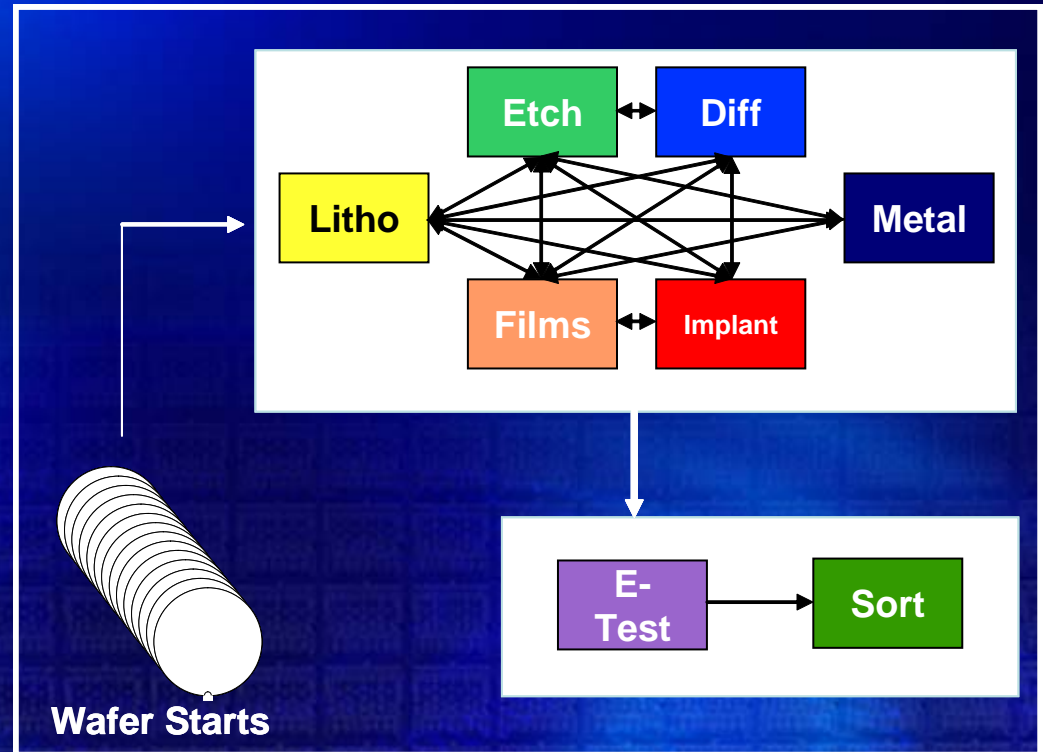
**Joint Statistical Meetings - Toronto
August 10th 2004**



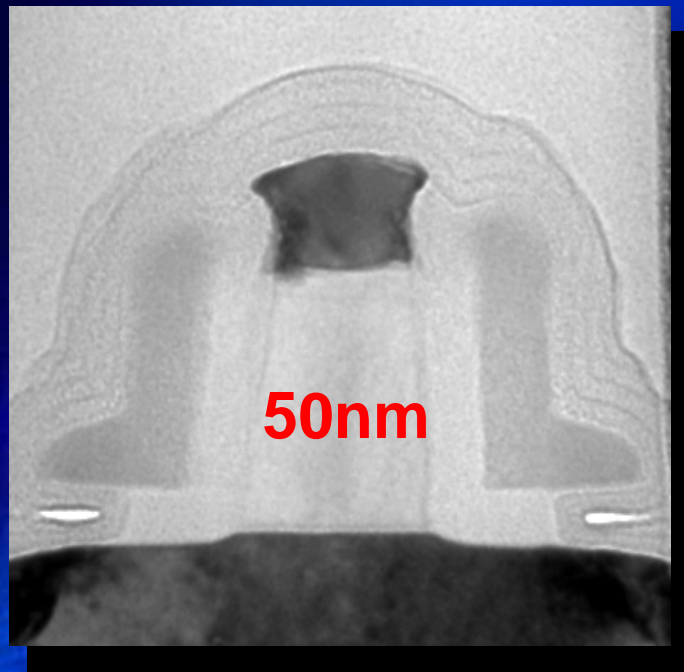
Semiconductor Manufacturing

Fun Fab Facts to Know and Tell

- Fabs are very expensive.
- Si Wafers are batched into Lots of wafers, and Lots are recursively queued in a lengthy manufacturing process.
- Fabs carry large inventories of wafers with long cycle times.
- 100-5,000 die, or “chip precursor”, may be made on a single wafer, depending on the device type and wafer size.
- Fabs may run hundreds of different device types on several technology families.

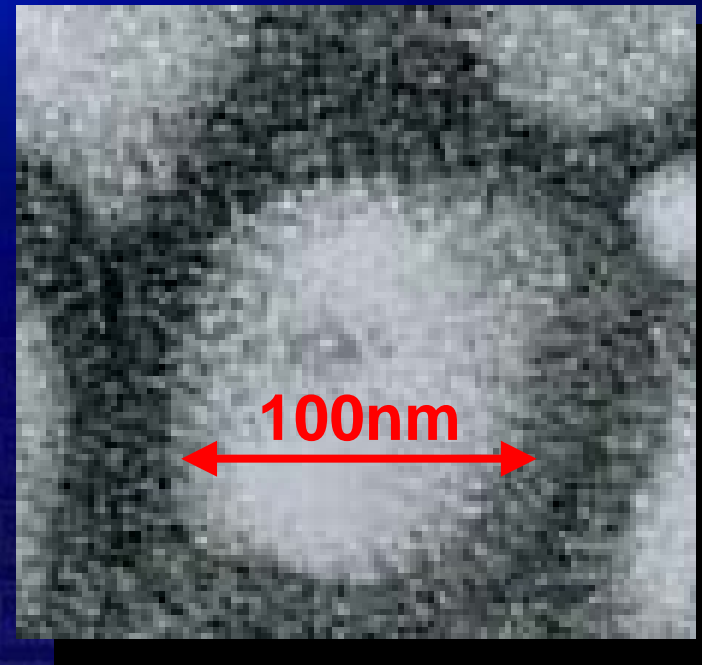


Silicon Devices Shrink Smaller than Virus Size



***Transistor for
90nm Process***

Source: Intel 2002



Influenza virus

Source: CDC

Semiconductor Manufacturing

One of the products of semiconductor manufacturing is a prodigious amount of data.

These data are generated and sampled at virtually every one of the ~500 process steps in the process flow.

Most of these data are happenstance in nature and the subsequent analyses are plagued with the attendant problems. (BHH, 1978)

Complex structure; non-normality; sparsity.

Despite it's problems, it would be powerful to use the information in the data for description or improvement...

The Problems

Some of the issues with data and analysis...

- An over-abundance of data with not enough people looking at it
- More variables than observations
- Outliers and missing data
- Unknown/unsuspected structure in the data
- Weird distributions (non-normal, mixed)
- Mixed data types

The ability to quickly detect problems and accurately predict results can have a huge positive impact on the bottom line.

The utility of Machine Learning will be demonstrated with case studies of simulated data.

Machine Learning

Machine Learning: An area of artificial intelligence involving developing techniques to allow computers to “learn”. More specifically, machine learning is a method for creating computer programs by the analysis of data sets, rather than the intuition of engineers.

T. Mitchell (1997). **Machine Learning**

Three Types of “learning” in Machine Learning -

- Supervised Learning: build a model between X and Y variables (classification and regression)
- Unsupervised Learning: model based on X variables only (cluster analysis)
- Learning to Learn: the algorithm learns its own inductive bias based on previous experience (adaptive learning)

Machine Learning using IDEAL

Interactive Data Exploration And Learning (IDEAL) is a suite of Intel Architecture optimized, state of the art, supervised machine learning methods for exploring and modeling relationships in complex data sets.

IDEAL placed in the top 5 of distinct learners in the NIPS 2003 conference – one of the leading conferences on machine learning.

The algorithms contained within IDEAL are

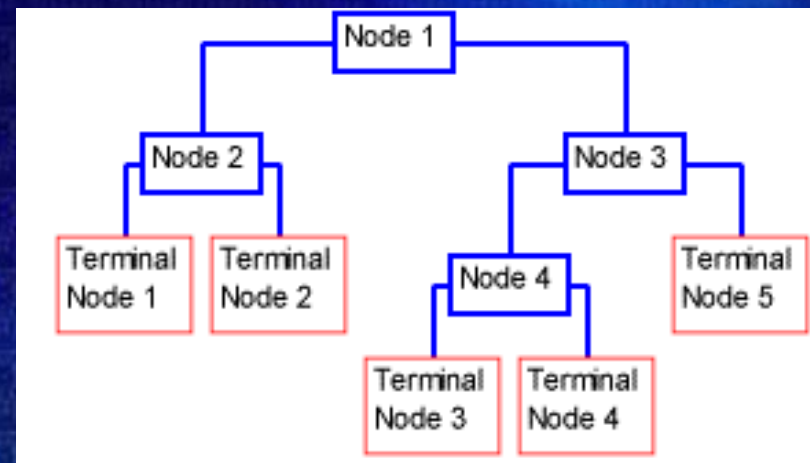
- Decision Trees / Classification and Regression
- Gradient Boosting Trees
- Random Forests

Decision Trees

Decision Tree analysis is one of the primary machine learning techniques.

A decision tree is a graphical display of relationships that may be in the data. It consists of a series of nodes where each node contains information of the data at that level and each level splits the previous level into distinct groups.

If the response is discrete, it is called a classification tree; and if the response is continuous, it is called a regression tree.



Binary Recursive Partitioning

Breiman, et al. (1984) developed Binary Recursive Partitioning to perform Classification and Regression Trees.

- Binary – each node is split into two and only two child nodes.
- Recursive – the same input variable can be used multiple times and each child node can be treated as a parent node.
- Partitioning – the data is divided into distinct subsets in each split.

Binary Recursive Partitioning splits can be performed by several algorithms. (See references for details.) All algorithms attempt to maximize the difference between the splits and minimize the difference within the splits.

Why Decision Trees?

Why Decision Trees?

- Handles large data sets well
- Nonparametric, robust technique
- Handles missing data and outliers well
- Builds models when there are more variables than observations.
- The results can be easily interpreted and explained to non-statisticians.

Cautions:

- Data are collected passively so all the warnings regarding happenstance data apply (Box, Hunter, Hunter (1978)).
- Decision Trees are best applied as Hypothesis Generators for further studies.

Gradient Boosting Trees

Gradient Boosting Trees build predictive models by generating a weighted combination of a large number of small trees.

The method generates an additive model by sequentially fitting small trees to pseudo-residuals from a regression at each iteration...

1. The first tree is fit to the data using a given loss function
2. A second tree is built on the residuals of the first.
3. A third tree is built on the residuals of the previous step (a tree that includes information from steps 1 and 2)
4. These steps are repeated through a number of trees until a given number of iterations is reached.

Stochastic Gradient Boosting Trees

Breiman (1996) introduced the idea of injecting randomness into function estimates to improve their performance.

How well the random selection improves performance depends on the following: training sample size, target function, proportion sampled, and the distribution of the deviations from the target function.

Friedman (1999) showed over a variety of conditions that the accuracy of gradient boosting can be considerably improved by taking random samples of the training data.

Random Forests

Random Forests (Breiman, 2001) are an ensemble of tree predictors where random vectors are generated that govern the growth of each tree and each tree casts a vote for the most popular class (or average prediction for regression).

- The Strong Law of Large Numbers ensures convergence of the error term so that overfitting is not a problem.
- An upper bound on the error is dependant on the accuracy of the individual classifiers and the correlation between them.

Random Forests

To improve accuracy, the randomness minimizes the correlation between trees while keeping the strength of the predictions.

- Random Input Selection: Select at random, at each node, a small group of input variables on which to split. Grow the tree to maximum size and do not prune.
- Random Linear Combinations: At a given node, L variables are randomly selected and added together with coefficients that are uniform random numbers on $[-1,1]$. F linear combinations are generated and then a search is made for the best split.

Why GBTs and RFs?

Why Stochastic Gradient Boosting Trees and Random Forests?

- Improved accuracy of predictions relative to other methods.
- Almost all the same positive aspects of Decision Trees (large data sets, nonparametric, robust, handles missing data and outliers, builds models when there are more variables than observations).
- Robust to noise in the data.
- Generally, increasing the number of random inputs does not dramatically reduce the error.

Cautions:

- Interpretation of the functional form is currently impossible. Work is being done in this area.

Which to use: GBTs or RFs?

Which one to use and when...

- Generally, for a continuous response GBT outperforms RFs.
- GBTs are more difficult to model and require a lot of attention to the tuning parameters.
- GBTs can handle large datasets with a moderate number of X variables. If the number of X variables is large then GBTs can be slow.
- RFs can handle a large number of X variables since it selects a random subset of all the X variables.
- RFs will not overfit the data. GBTs can overfit the data but the risk of this can be reduced by using random subsets to fit the model.

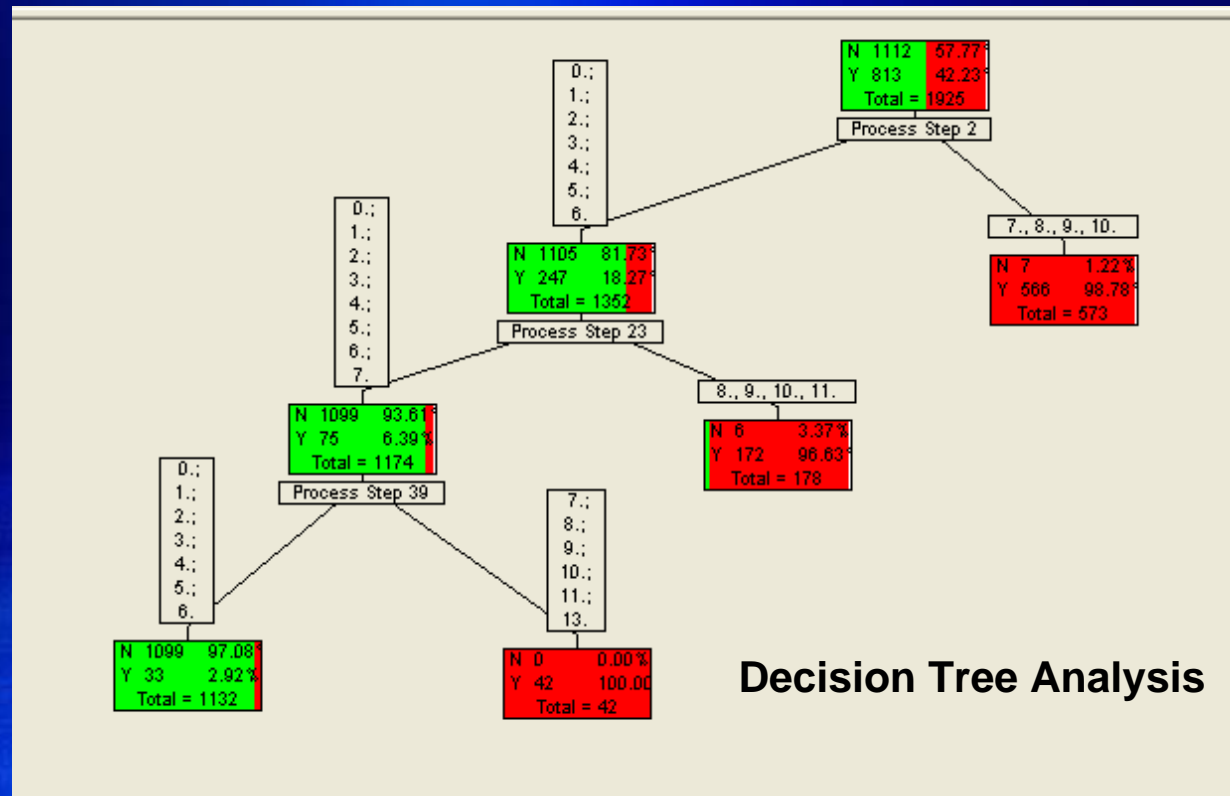
Case Study #1: Equipment Commonality

Equipment Commonality studies are challenging to analyze, especially if equipment performance interacts across multiple process steps.

When a defect is found, one of the first things considered is if the defective units were produced on the same equipment set.

In this case, there are 1,925 lots and approximately 42% of these have the defect. The equipment used on each of 39 process steps was analyzed.

Equip Commonality – Optimal Tree



The optimal tree in the case indicates that an interaction of equipment from Process Steps 2, 23, and 39 are the likely culprits of the defect. The equipment at these process steps would be investigated to determine the root cause of the problem.

Case Study #1: Equipment Commonality

Due to the long cycle time of material in a fab a large number of lots can be processed at an operation before the first signal of a problem is discovered. In this scenario, assume that prior to implementing the fix at those three process steps, an additional 75 lots were processed through those steps. It is of great interest to management to predict whether the lots will have the defect or not.

Since classification prediction is the primary objective, RFs will be used to address this issue. A RF model was fit using the information from the 1,925 lots and then predictions were performed on the 75 lots.

Case Study #1: Equipment Commonality

RFs predicted that 27 of the 75 lots would have the defect.

In this simulated example, it is known that 29 of the 75 lots did have the defect. The classification using a RF model is as follows:

		Predicted Defect	
		Classification	
True Defect	N	RF N	RF Y
	Y	46	0
		2	27

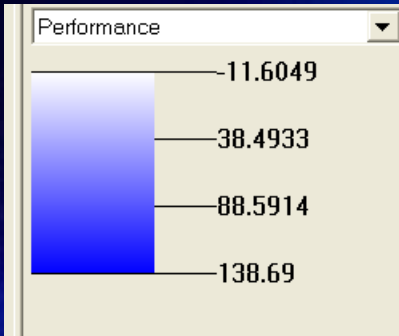
Case Study #2: Prediction of Performance

The three key items of focus in the semiconductor industry are reliability, performance, and yield. The ability to accurately predict each of these as early as possible is critical.

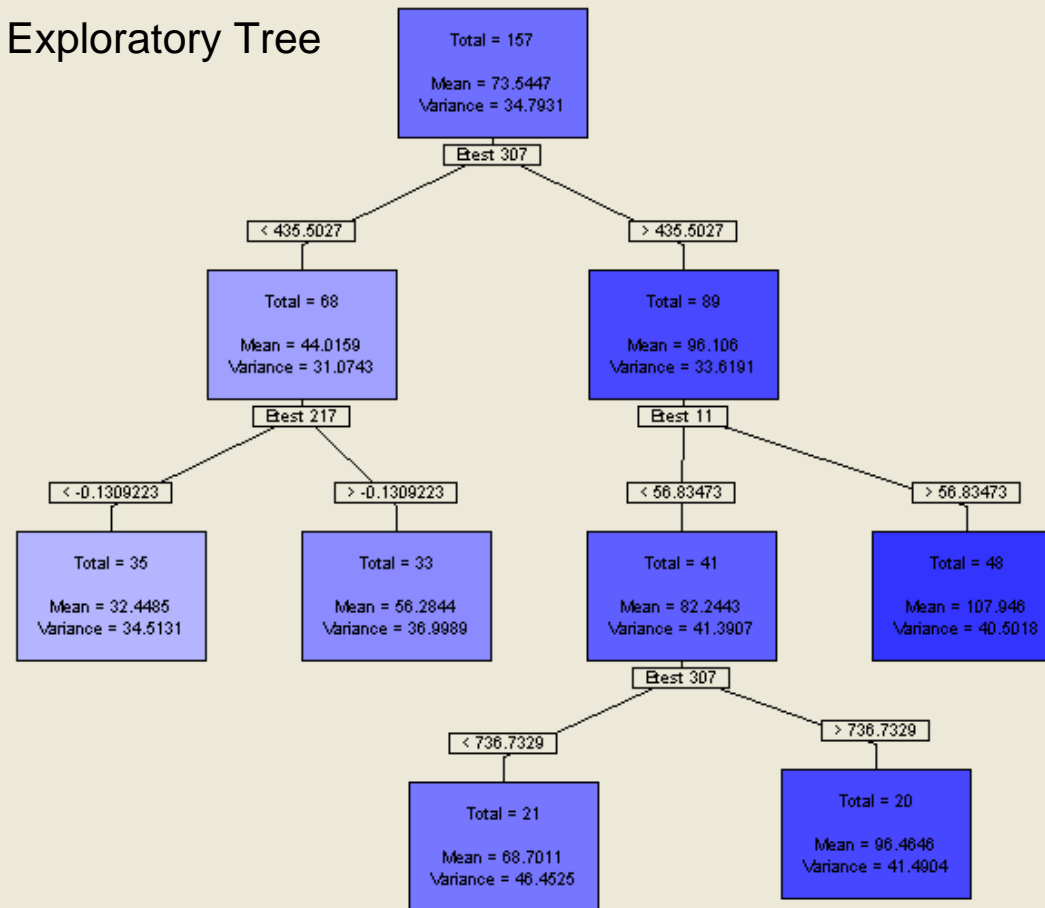
Consider the case where interest is in a Performance parameter. There are 157 lots and 307 inline and electrical test parameters.

The objective of the analysis is to determine which parameters have the greatest influence on the response and to develop a model to predict future performance values.

Case Study #2: Prediction of Performance



Exploratory Tree



Case Study #2: Prediction of Performance from Electrical Test

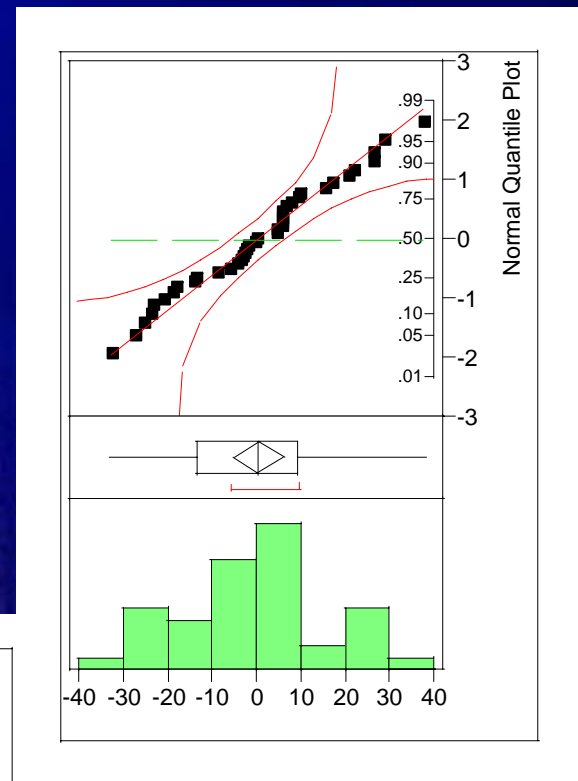
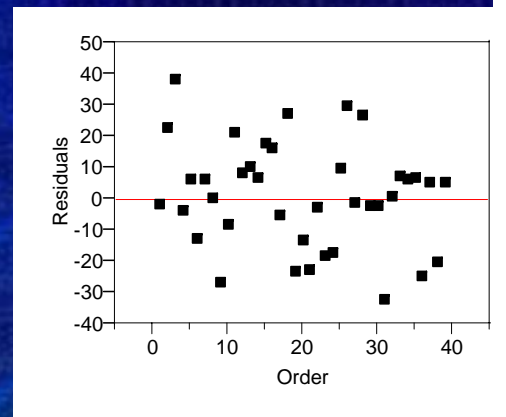
The first objective is to determine which variables have the largest influence. From the Regression Tree it can be seen that Electrical Test variables 307, 217, and 11 have the largest influence.

The second objective is to develop a prediction model for the Performance parameter. Since the response is continuous, this was accomplished using GBTs.

An additional 39 lots were simulated that were not used in building the model to determine the prediction accuracy.

Case Study #2: Prediction of Performance from Electrical Test

These predictions could be used to prioritize high-performance material, reduce test times, and improve monitoring of important steps.



Moments	
Mean	0.4893821
Std Dev	16.883651
Std Err Mean	2.7035478
upper 95% Mean	5.9624285
lower 95% Mean	-4.983664
N	39

Conclusions and Cautions

Machine Learning algorithms are powerful techniques that take advantage of the available computational power.

These are among the only techniques capable of predictive statistics when the number of variables exceeds the number of observations...this happens in the semiconductor industry frequently!

Some Statisticians feel threatened by these methods. These methods should not replace traditional statistical methods, but can enhance the statistician's ability to explore data and make predictions.

“...using fancy tools like neural nets, boosting and support vector machines without understanding basic statistics is like doing brain surgery before knowing how to use a band-aid”

Larry Wasserman (2004). *All of Statistics*

References & Acknowledgements

1. Box, G.E.P., Hunter, W.G., Hunter, J.S.(1978); *Statistics for Experimenters*; John Wiley and Sons; New York.
2. Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984); *Classification and Regression Trees*; Chapman & Hall; New York.
3. Breiman, L. (1996). Bagging Predictors. *Machine Learning* 26, 123-140.
4. Breiman, L. (2001). Random Forests. Technical Report, University of California. Berkeley.
5. Friedman, J.H. (1999). Stochastic Gradient Boosting. *Technical Report*, Stanford University.
6. Hand, D., Manila, H., Smyth, P. (2001); *Principles of Data Mining*; MIT Press.
7. Hastie, T., Tibshirani, R., Friedman, J. (2001); *The Elements of Statistical Learning*; Springer; New York.
8. Mitchell, T. (1997); *Machine Learning*; McGraw-Hill.
9. Salford Systems (2002); "Salford Systems White Paper Series." Website: <http://www.salford-systems.com/whitepaper.html>.
10. Wasserman, L. (2004); *All of Statistics*; Springer-Verlag; New York.

Special thanks to Eugene Tuv, Somnath Shahapurkar, Randall Goodwin, & Russell Miller!