Machine Learning

Stephen O'Connell

Capacity and Performance Analysis

Overview

- Problem
- How Machine Learning can help
- Status:
  - Training Dataset
  - MART – Mike Bowles
  - SVM – R e1071 package
- Plans
- Demo

Problem Space

- Many Many Servers to run the business
- Functional organized, not a lot of cloning
- Different Workloads: Database, transactional, business analytics, Other (?????)
- How do you do capacity planning, forecast utilization, identify problems?

Problem Space – The old way

- Capture Metrics all day long, CPU, Memory, Disk, Network
- Consolidate data to hourly, daily, monthly avg, max, min, p95, etc.
- Monthly use the historical data to generate a forecasted utilization
  - 180 days, Mon-Fri, approx 130 days into the model, simple regression.
- Looking servers “out of gas”, broken, declining, etc.

Problem Space - Old Way

	A	O	P	Q	R	S	T	U	V	W	X
1	*Host Capacity Forecast Report*		<0%	>80%	>90%						
2	server_name	days	avg_30_days	avg_60_days	avg_90_da	avg_180_c	p95_30_da	p95_60_da	p95_90_da	p95_180_c	avgm
3		129	145.44	169.73	194.02	266.89	140.25	161.71	183.18	247.56	5
4		129	131.69	146.18	160.68	204.16	132.62	147.04	161.46	204.73	3
5		126	117.36	132.64	147.91	193.73	117.97	133.18	148.4	194.05	5
6		129	112.19	135.61	159.03	229.29	116.24	139.45	162.66	232.3	3
7		129	98.66	116.56	134.47	188.17	103.55	121.31	139.06	192.33	6
8		129	93.27	110.22	127.17	178.01	107.51	122.72	137.94	183.57	8
9		129	93.1	110.49	127.87	180.02	103.05	122.55	142.05	200.54	3
10		129	89.35	92.05	94.75	102.85	95.98	97.89	99.79	105.52	3
11		129	87.83	100.52	113.21	151.27	105.92	117.33	128.75	162.99	2
12		42	75.69	106.86	138.02	231.53	85.12	118.05	150.97	249.75	5
13		129	75.61	76.75	77.89	81.31	78.41	79.53	80.66	84.05	5
14		129	75.29	75.96	76.62	78.62	100.03	100.06	100.09	100.19	1
15		129	75.27	90.63	105.99	152.07	89.8	104.27	118.74	162.15	2
16		129	74.61	85.59	96.57	129.52	102.6	119.14	135.68	185.31	4
17		129	73.91	83.64	93.37	122.54	97.05	107.96	118.88	151.62	7
18		129	73.87	92.05	110.24	164.79	88.42	110.11	131.79	196.83	
19		115	73.36	83.59	93.82	124.51	73.87	83.81	93.74	123.56	
20		129	73.24	82.98	92.72	121.94	74.12	83.93	93.75	123.2	4
21		129	72.43	74.81	77.19	84.34	94.57	97.9	101.24	111.25	6
22		125	71.97	84.55	97.13	134.87	90.55	102.53	114.5	150.42	2
23		129	71.85	78.5	85.15	105.11	88.94	95.47	101.99	121.57	7
24		85	71.08	90.73	110.38	169.34	76.11	97.01	117.92	180.64	5
25		26	70.72	105.89	141.06	246.58	87.74	127.98	168.21	288.92	6

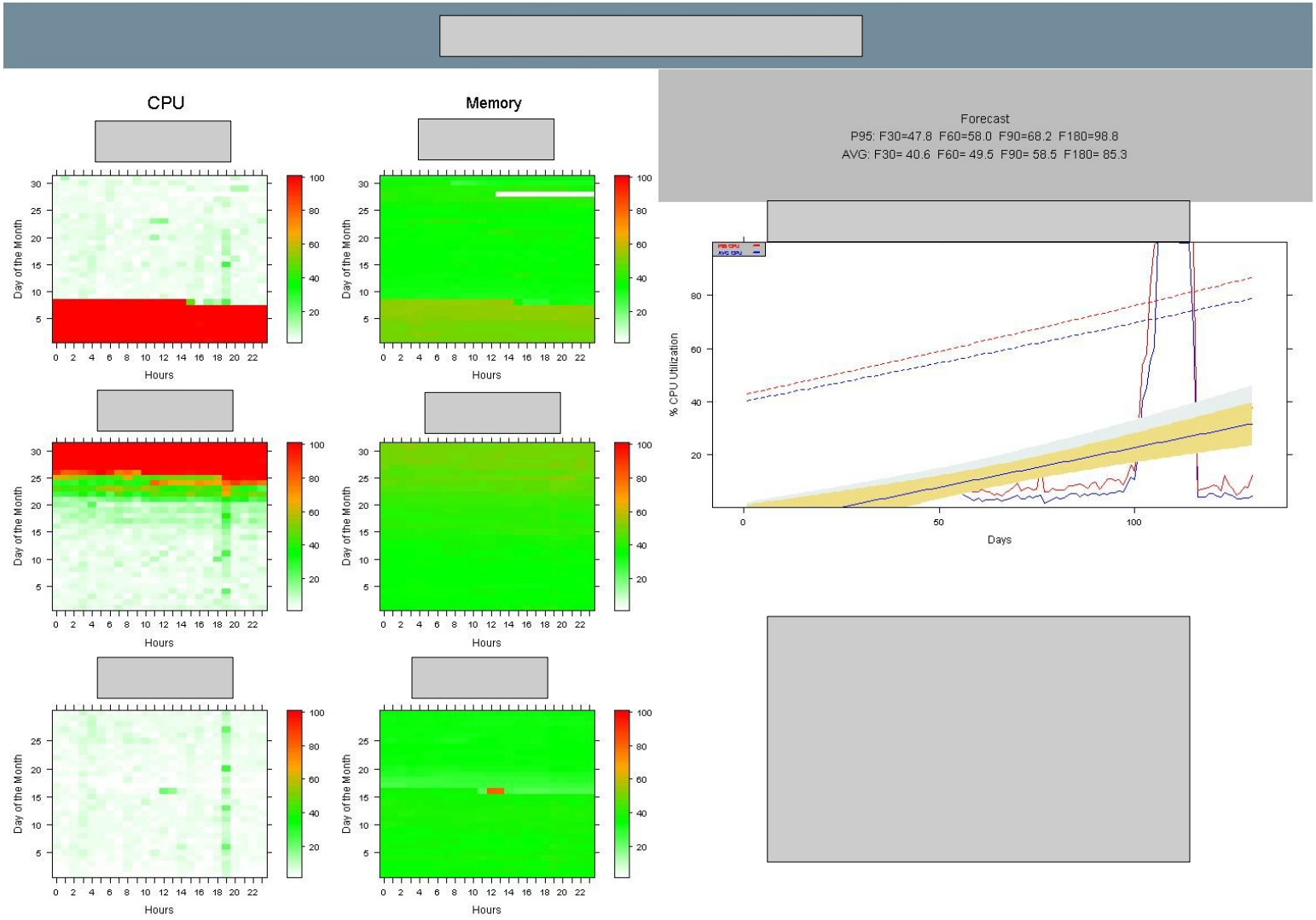
Produce a spreadsheet of all the numbers

Sort by the busiest Server

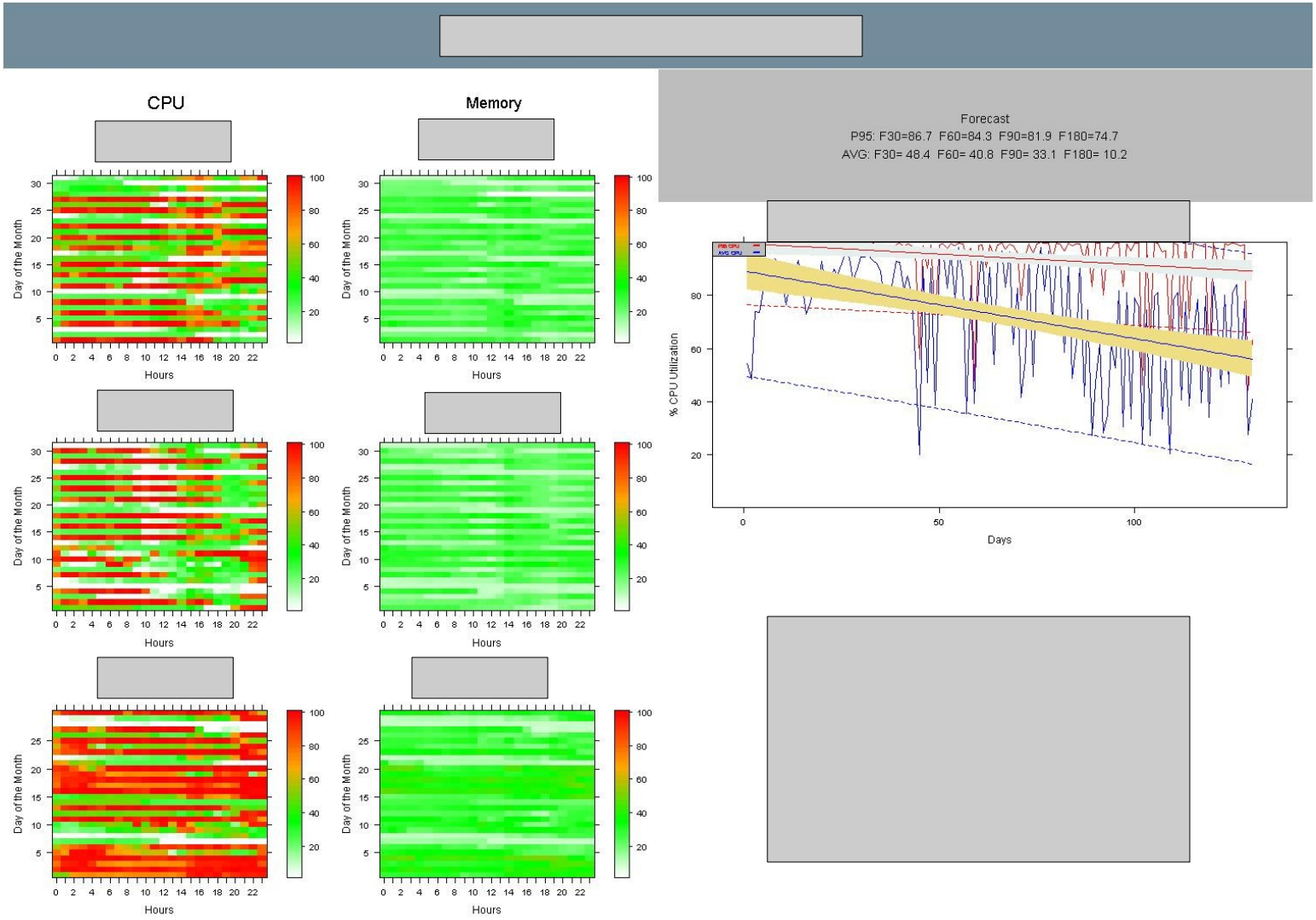
Top 10 - Bottom 10

Doesn't work for 1800+ servers, really won't work for 4000 servers

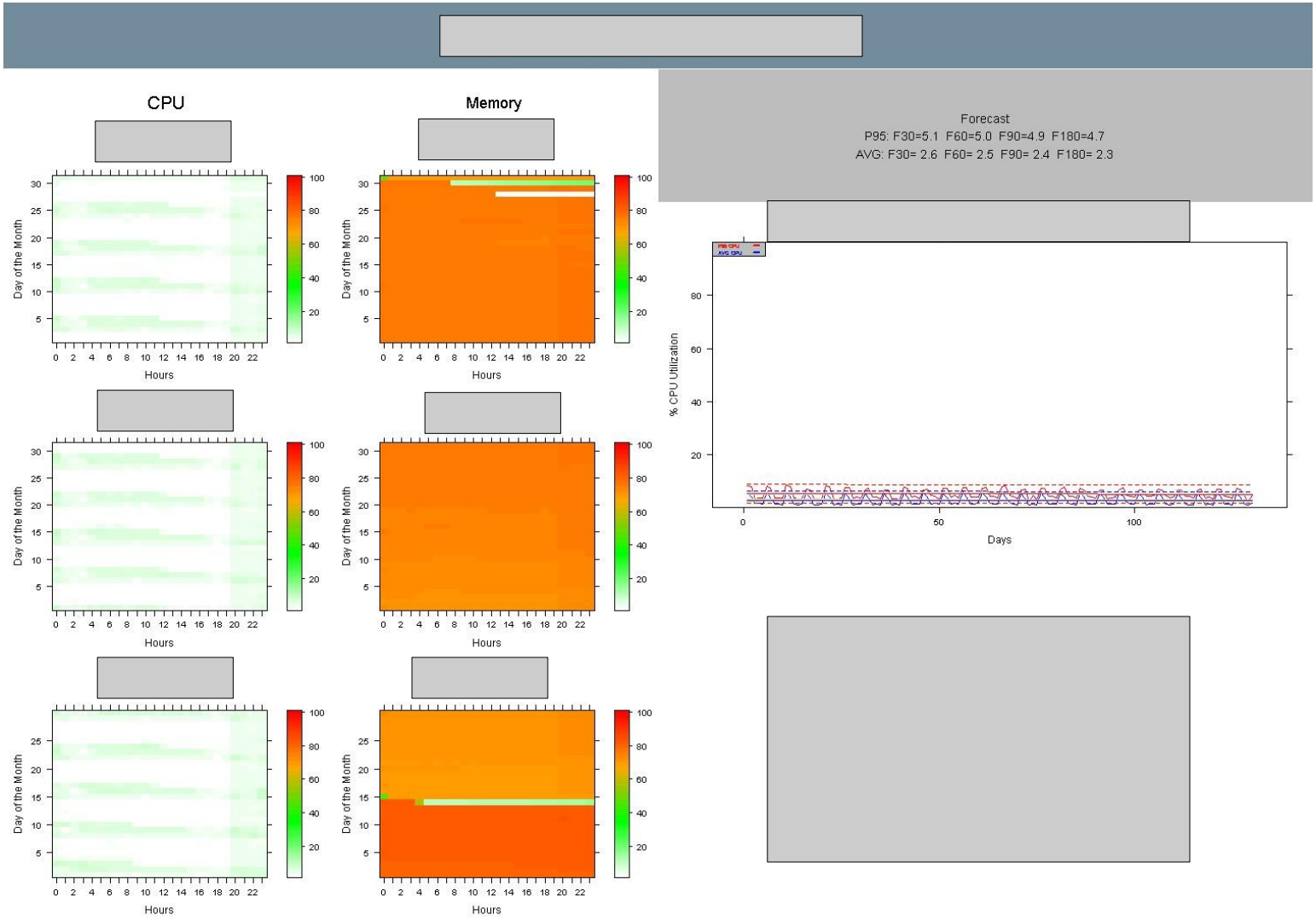
Problem Space – Types of Servers – Broken - Fixed



Problem Space – Types of Servers – Highly Variable - Cycle



Problem Space – Types of Servers – Low



Problem Space – Types of Servers – What I am looking for

How Machine Learning Can Help

- Tons of servers with similar “patterns”, with LOTS of noise
- ML is great at finding the patterns in the data that are re-occurring
- Training can be straight-forward, after the first pass
- Can programmatically incorporate into the monthly forecasting process, tag the servers with the “guessed” pattern.
- Can be retrained as needed.

Status

- Created a training dataset - Manually went through 1800 Servers and classified them as:
    - 1 - Low
    - 2 - Broken
    - 3 - Variable
    - 4 - Monitor
    - 5 - Fixed
    - 6 – Declining
- Created Matrix from the Raw Data
- Evaluated R Package - E1071 – SVM Model developed as an initial pass at classification of the data

Status – e1071 SVM

- Have developed initial set of models using the training data
- Have done some tuning of the model
- Have created visualizations of the data to help understand the “patterns” in the data, and can be used to tune the training data.

Untuned Model Results: Not very good

NUMBER IN EACH CLASS

```
  1   2   3   4   5   6
529  26 486 119  64  91
```

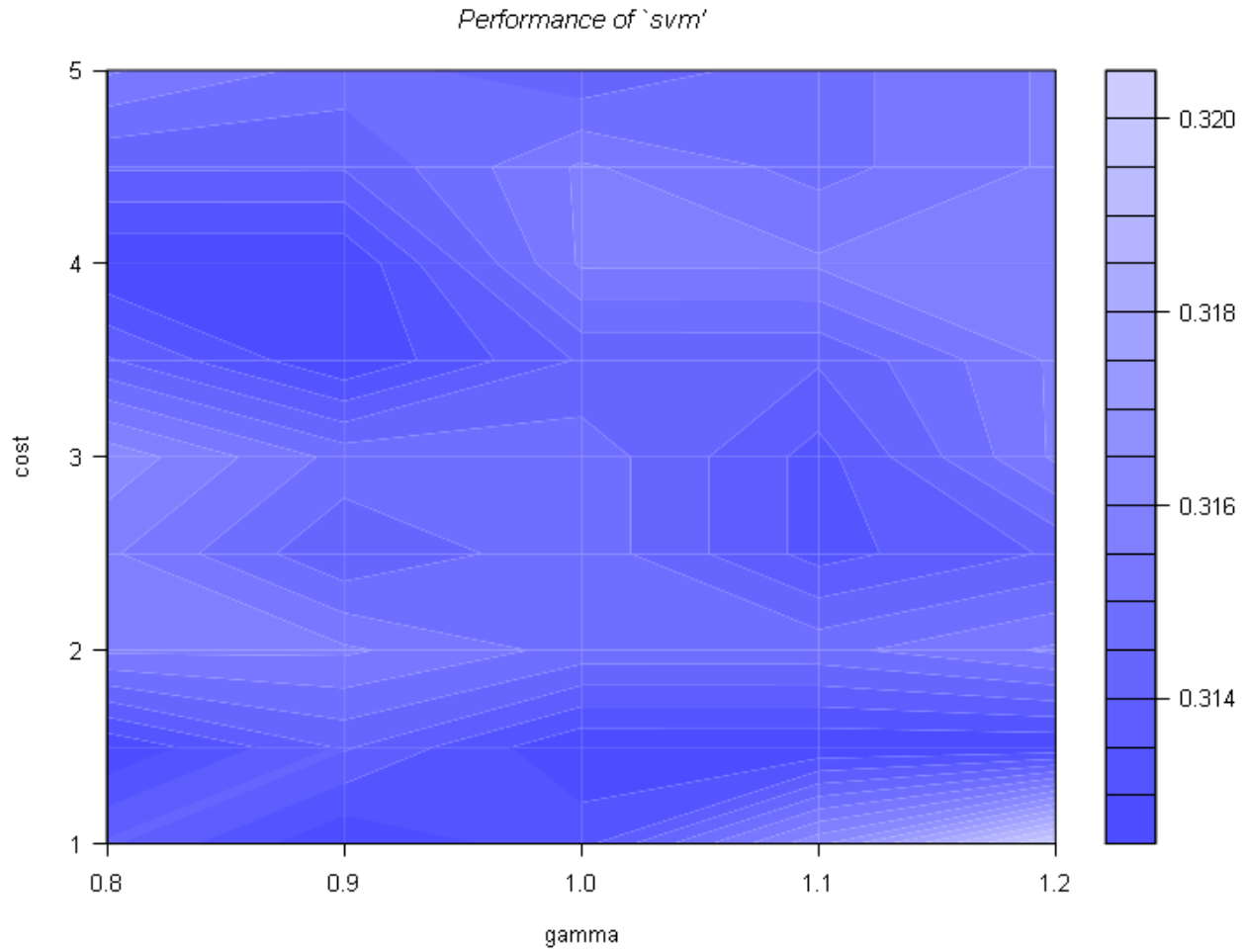
CONFUSION MATRIX

```
    y
pred   1   2   3   4   5   6
   1 478   1 121   0   1   9
   2   0  15   0   0   0   0
   3  50   3 357  17  10  35
   4   0   7   6 101   1   2
   5   0   0   1   1  49   0
   6   1   0   1   0   3  45
```

PERCENTAGE OF PROPERLY CLASSIED SAMPLES

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 90.4 57.7 73.5 84.9 76.6 49.5
```

Tuning Grid: Visual of the grid search for “best” cost and gamma



Tuned the svm with a range of Cost and gamma using grid search
Sweet Spot:
Cost = 3.5
Gamma = .9

Tuned Model Results: Improved Classification – over fit?

TUNNING RESULTS

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 gamma cost
   0.9  3.5

- best performance: 0.3125318
```

CONFUSION MATRIX

```
     y
c_new   1   2   3   4   5   6
    1 529   1   4   0   0   0
    2   0  25   0   0   0   0
    3   0   0 482   0   0   0
    4   0   0   0 119   0   0
    5   0   0   0   0  64   0
    6   0   0   0   0   0  91
```

PERCENTAGE OF PROPERLY CLASSIED SAMPLES

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]  100 96.2 99.2  100  100  100
```

Misclassification analysis: What specific samples were misclassified.

SAMPLE NUMBER AND TRAINING CLASSIFICATION

```
199 850 857 868 998
  1   1   1   1   1
Levels: 1 2 3 4 5 6
```

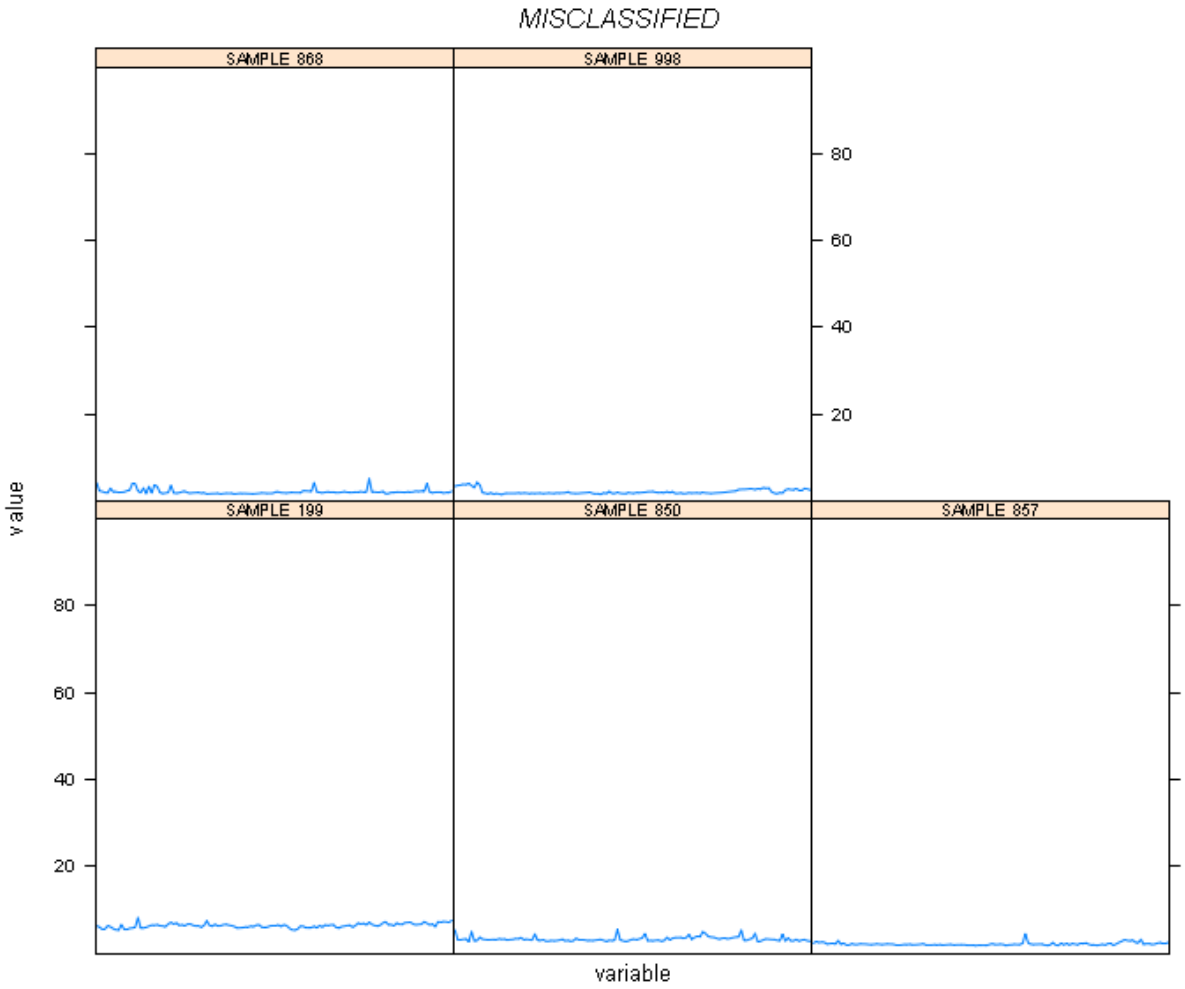
WHAT THEY WERE 'MISCLASSIFIED' AS BASED ON THE SVM MODEL

```
[1] "2" "3" "3" "3" "3"
```

MISCLASSIFIED SAMPLE NAMES - INCLUDED IN THE FOLLOWING CHARTS

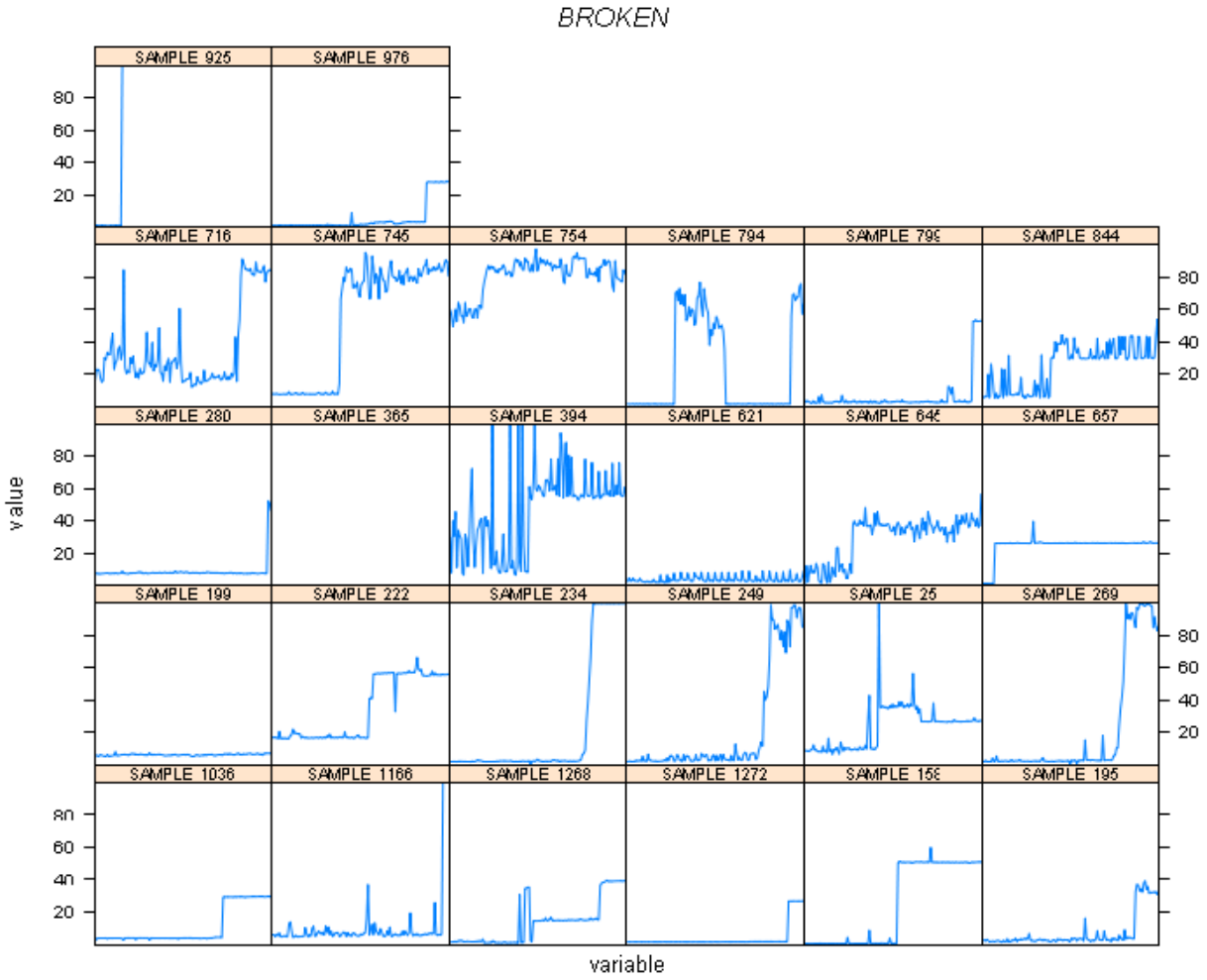
```
[1] "SAMPLE_199" "SAMPLE_850" "SAMPLE_857" "SAMPLE_868" "SAMPLE_998"
```

Training Data Visualization: Misclassified Samples



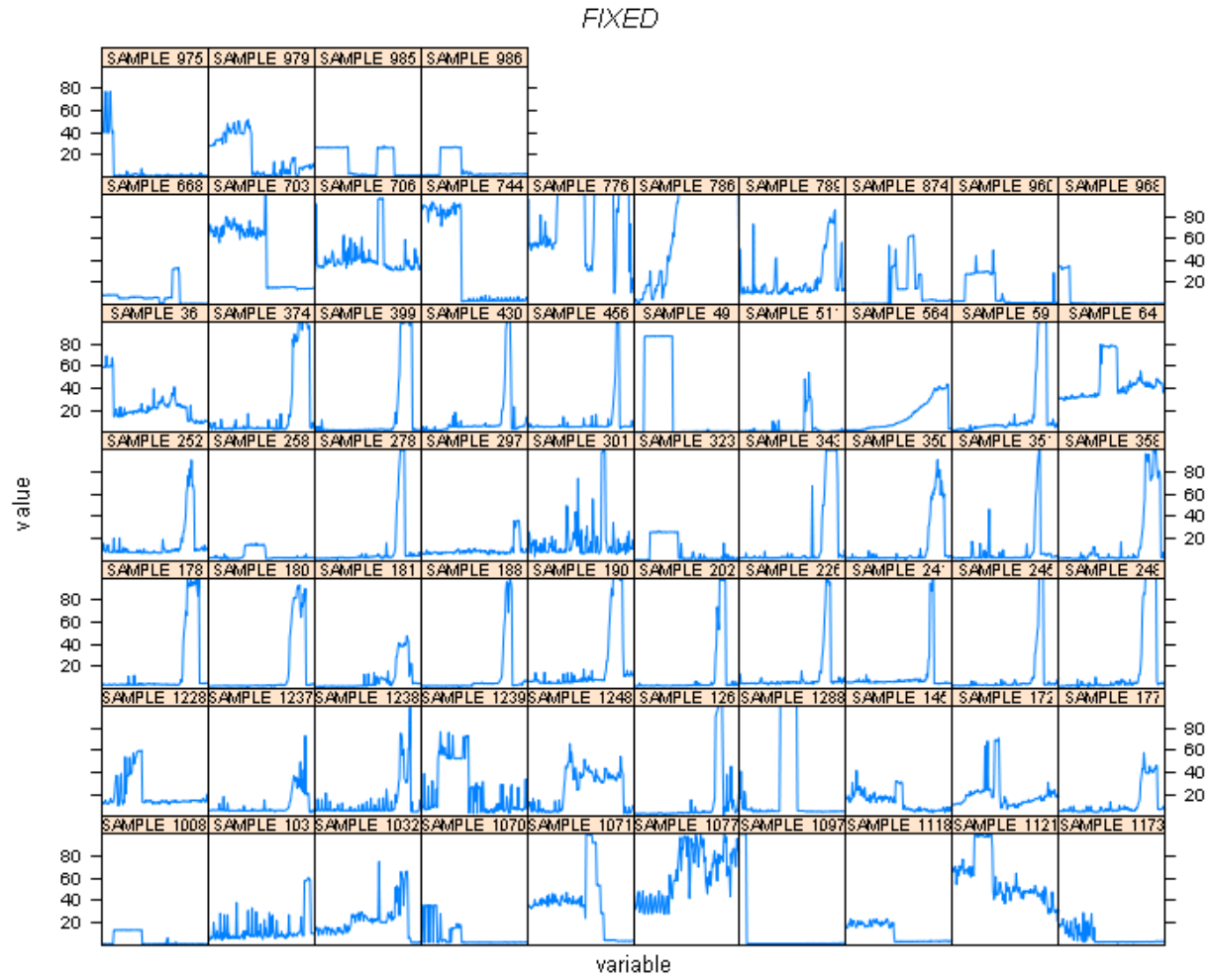
Misclassified samples, visual can be used to evaluate why the sample was misclassified. Should it be removed? Should be classified differently in the training data?

Training Data Visualization: CLASS = 2 – BROKEN SAMPLES



CLASS 2 – BROKEN SAMPLES have a pattern with a "step up" at the end of the feature set, i.e feature (utilization) has stepped up to a consistent level indicating a possible problem.

Training Data Visualization: CLASS = 5 - FIXED SAMPLES



CLASS 5 - FIXED SAMPLES are samples with a pattern of "step down" at some point in the feature set indicating utilization has returned to a stable state.

Plans

- Experiment more with MART, e1071, Shugon, other classification techniques
- Measure results of classification on “real” data – working on prior month data...
- Incorporate ML techniques into monthly reporting process to enhance the capacity, forecast, and problem resolution process.
- Will need to build a 're-training' tool/technique to improve the patterns used in the model to make classification predictions.

Demonstration

- Review 'basic' examples of R package e1071 – svm_tutorial.R
- Review 'CPU classification' code – stephen_cpu.R
- All data and code are included in demo.zip on machinelearning site
  - Includes R console output if you don't have R available to run the code