# Divided CPU and Node Aware Elastic Net

*L3P3*

*Thursday, November 13, 2014*

This document summarizes the first approach in order to expand the node aware elastic net regularized predictor v3 to include resource awareness. This second iteration adds two columns to each input matrix consisting on the mean cpu value for the predicted node for each training instance time and the variation of that resource over the mean. The whole model creation is described in the file "rnen2-whole.R", found in the private L3P3 Github repository. The models are exported in a variable called "models_sc", which is a list of all the combinations of nodes and events in the system, containing each one the best created model for each event.

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
load("data.Rdata")
```

Our objective is now to study the performance of this models. To do so, we're going to create a matrix where the rows are the system nodes and the columns are the events that happened on the system. The coefficients of the matrix will be the f-score of each model in each node.

Let's now extract some insightful analytics.

- First, let's see how many models were correctly created and how many models failed to create:

```
## [1] "Possible models:" "1335"
```

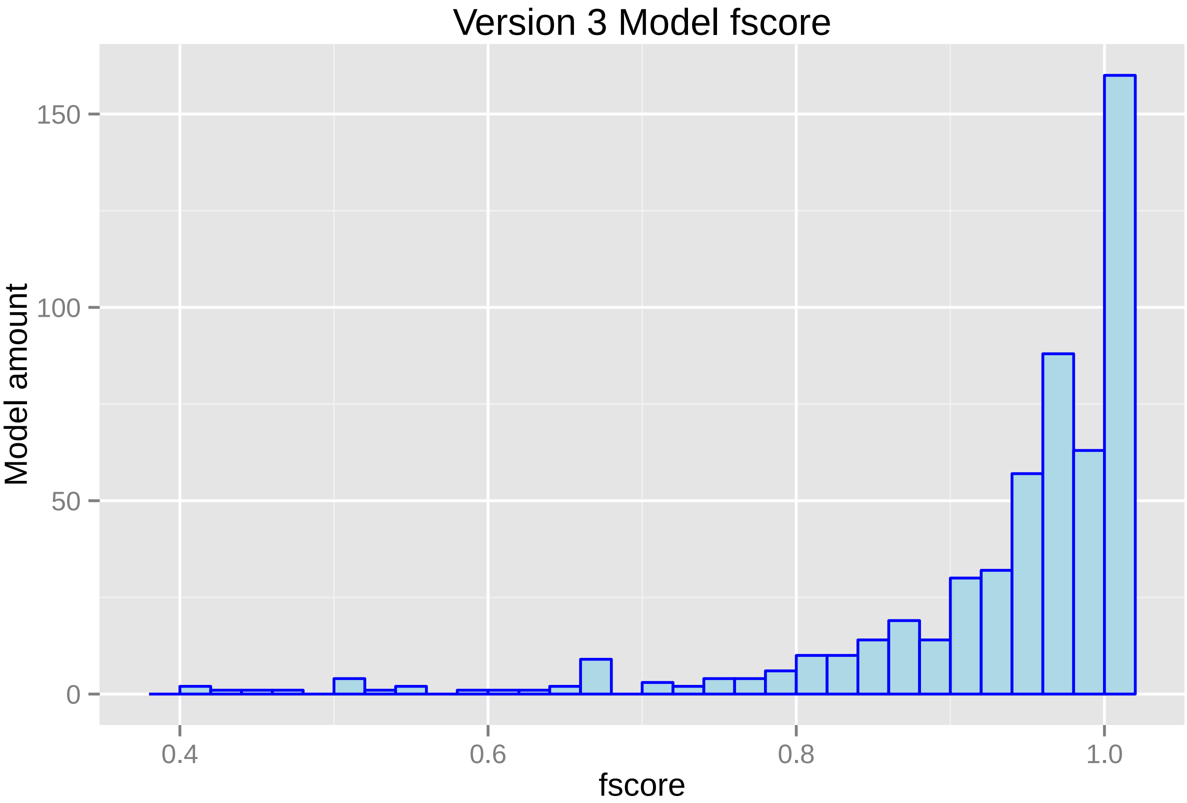
```
## [1] "Created models:" "542"
```

```
## [1] "Failed models:" "793"
```

The percentage of created models is a 40.5992509%. A logical result, as each model has now far fewer data to train.

- Now, we'll study obtained fscores in the 563 correctly created models:

```
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



The histogram shows good results for the models that actually work. A 79.3357934% (430 models) is at least 0.9 and a 91.697417% (497 models) is at least 0.8. The average obtained fscore of the created models is 0.9318644, a slightly lower amount compared to the one found in the previous iteration. These models are hugely simpler than those, though, which makes still a feasible option in deployment.

- Lastly, we will study which critical events were correctly modelled and their fscore. There are three different critical events on the system, which we will study separately:
- 96731154: this event indicates a critical threshold violation. It only appears in two nodes.

```
## fwpeuib02.sceu.corp fwpeuid02.sceu.corp
##                   0                   1
```

- 68917: this event's message points that a device hast stopped responding to external requests and/or polls. It is greatly associated with a change in the network on the 1st of June.

```
##        aswpeuib02.sceu.corp        aswpeuib04.sceu.corp
##                        0.00                        0.00
##        aswpeuid01.sceu.corp        aswpeuid02.sceu.corp
##                        0.00                        0.00
##        aswpeuid03.sceu.corp        aswpeuid04.sceu.corp
##                        0.00                        1.00
##        aswpeuin01.sceu.corp        aswpeuin02.sceu.corp
##                        0.00                        0.96
##        aswpeuin03.sceu.corp        aswpeuin06.sceu.corp
##                        0.00                        0.00
##        aswpeuin08.sceu.corp        aswpeuin09.sceu.corp
##                        0.00                        0.00
##        aswpeuin10.sceu.corp        aswpeuin12.sceu.corp
##                        0.00                        0.00
##        ASWPEUIN13.sceu.corp        ASWPEUIN19.sceu.corp
##                        0.00                        0.00
## dnsrespeuin02-adm.sceu.corp        dswpeuin01.sceu.corp
##                        0.00                        0.00
##        dswpeuin02.sceu.corp        dswpeuin03.sceu.corp
##                        1.00                        0.00
##        dswpeuin04.sceu.corp        dswpeuwt01.sceu.corp
##                        0.00                        1.00
##        dswpeuwt02.sceu.corp        dswpeuwt03.sceu.corp
##                        0.00                        0.00
##        dswpeuwt04.sceu.corp         fwpeuid02.sceu.corp
##                        0.00                        0.00
```

A total of 26 models are created, out of which 4 were correctly created, this is, a 15.3846154%, with an average fscore of 0.99 and a standard deviation of 0.01.

```
* 69481:this event's message poitns that a whole chassis has stopped responding to polls. While it is g
```

```
## aswpeuib02.sceu.corp aswpeuib04.sceu.corp aswpeuid01.sceu.corp
##            0.0000000            0.0000000            0.0000000
## aswpeuid02.sceu.corp aswpeuid03.sceu.corp aswpeuid04.sceu.corp
##            1.0000000            0.0000000            0.0000000
## aswpeuin01.sceu.corp aswpeuin02.sceu.corp aswpeuin03.sceu.corp
##            0.0000000            0.0000000            0.0000000
## aswpeuin06.sceu.corp aswpeuin08.sceu.corp aswpeuin10.sceu.corp
##            1.0000000            0.0000000            0.0000000
## aswpeuin12.sceu.corp ASWPEUIN13.sceu.corp ASWPEUIN19.sceu.corp
##            0.0000000            0.0000000            1.0000000
## dswpeuin01.sceu.corp dswpeuin02.sceu.corp dswpeuin03.sceu.corp
##            1.0000000            0.9615385            0.0000000
## dswpeuin04.sceu.corp dswpeuwt01.sceu.corp dswpeuwt02.sceu.corp
##            1.0000000            0.9787234            0.9787234
## dswpeuwt03.sceu.corp dswpeuwt04.sceu.corp
##            0.0000000            0.0000000
```

A total of 23 models are created, out of which 8 were correctly created, this is, a 34.7826087%, with an average fscore of 0.9898732 and a standard deviation of 0.0074744.

## Conclusions

Adding the CPU consumption has increased the amount of created models and also increased (though in a really small amount) the average fscore, but has decreased the performance of the critical events models. In a model with 1400 input variables it is expected for a single variable not to cause a huge change in performance. It is probable to see a larger change with every resource consumption variable.