Fermilab | U.S. DEPARTMENT OF ENERGY | Office of Science

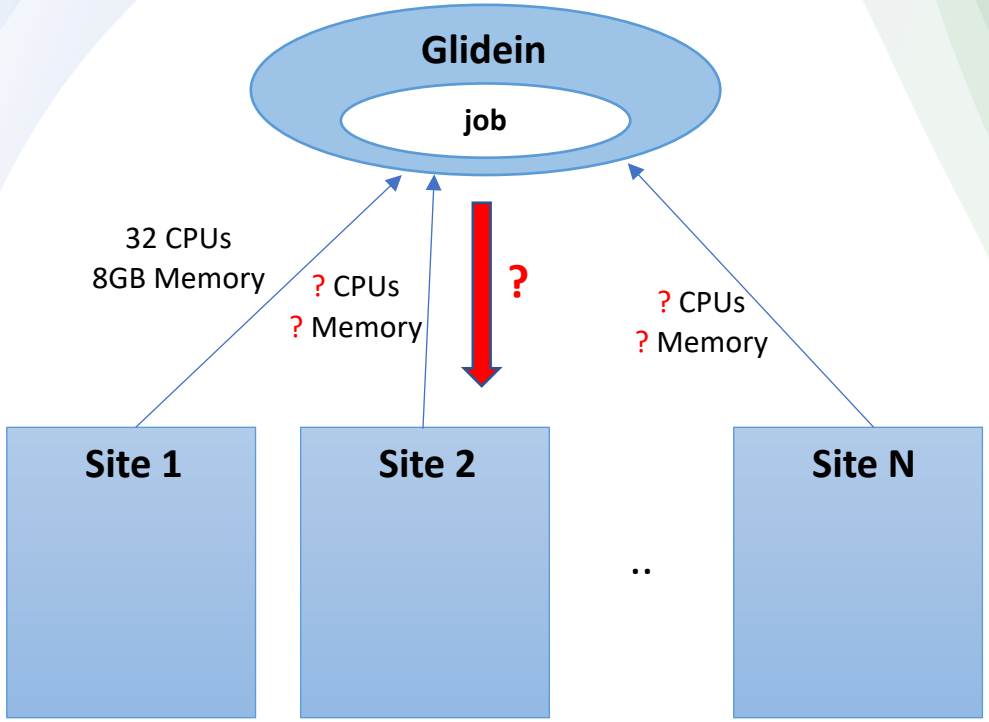
Flexible Pilot Jobs Framework for Distributed High Throughput Computing

Candidate: Franco Terranova
Supervisor: Marco Mambelli

Final-Term Presentation

Fermilab Italian Summer School 2022

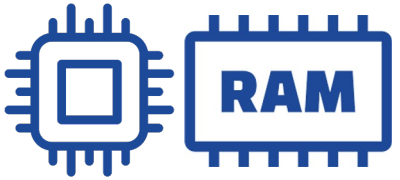
IN SUPREMÆ DIGNITATIS 1343



Job/Site Matching problem

What if we can use AI to predict what is the best site where the glidein should be spawned?
Some sites give information about the number of resources that are going to provide for the job, while some others don't.
We want to allocate the glidein to the site that provide the largest amount of resources while minimizing the probability of failure.

$\max Resources$



CPU Memory

$\min Failure$

Failure

CPU, Memory and Probability of Failure

- Predict the amount of CPU and Memory that is going to be provided from each site

$$CPUProvided,\ MemoryProvided$$

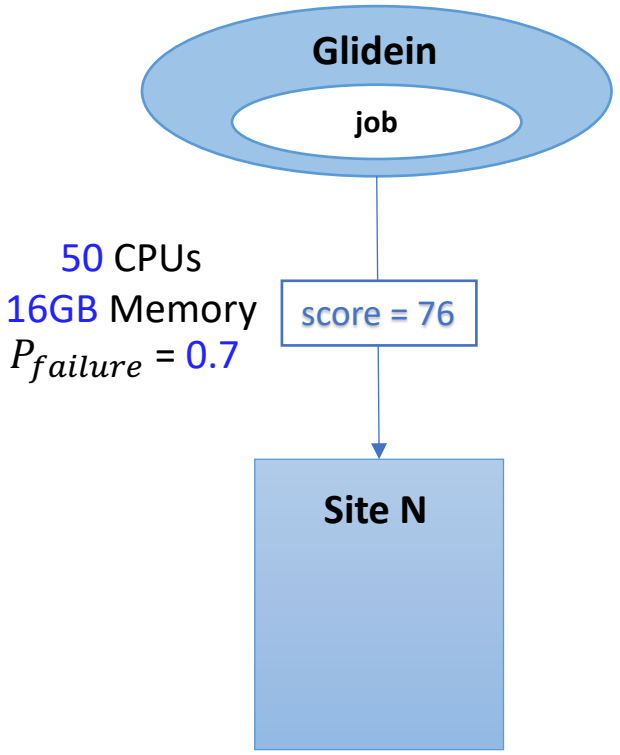
- Consider the sites for which the resources provided are enough for our job

$$CPURequested \leq CPUProvided$$
$$MemoryRequested \leq MemoryProvided$$

- Calculate the probability of failure of each site

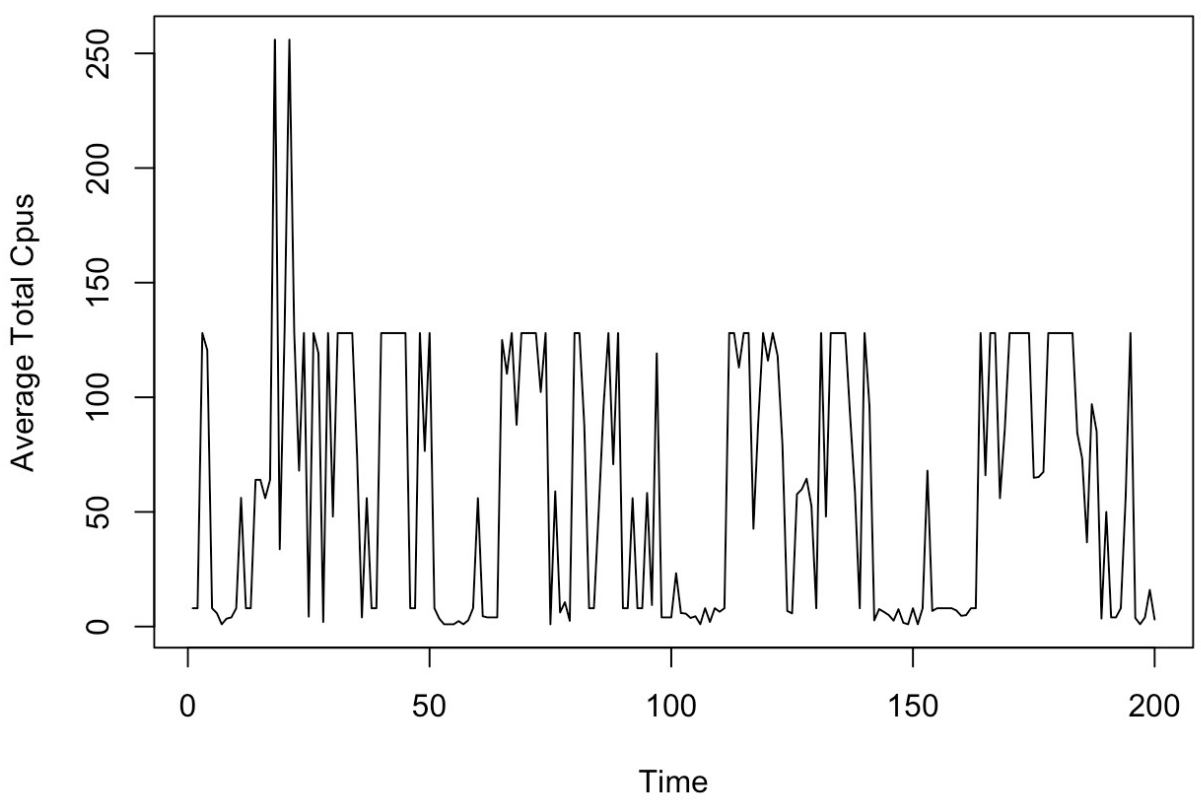
$$P_{failure}$$

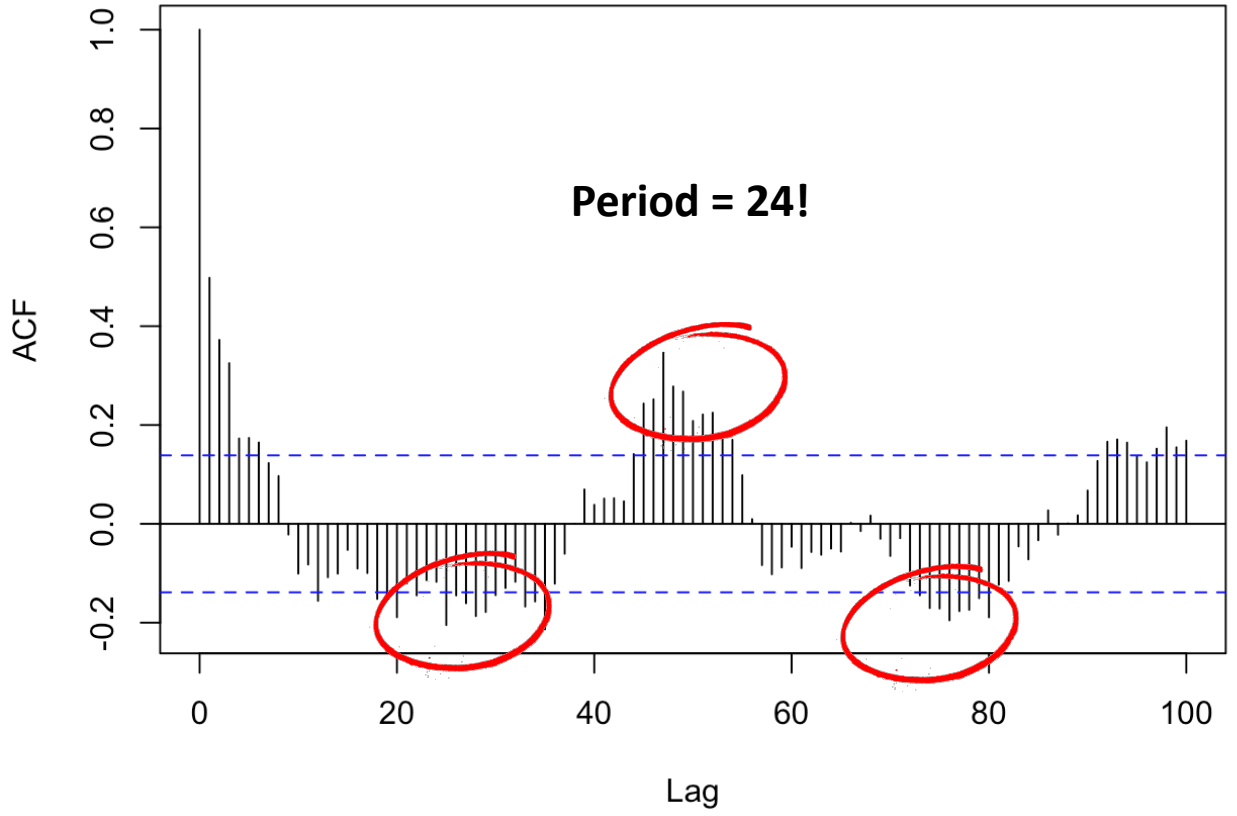
- Calculate a cumulative score that allows us to take this decision



Time Series Analysis - CPU

Predict the average total hourly CPU that a certain site can provide, in order to select the best one.



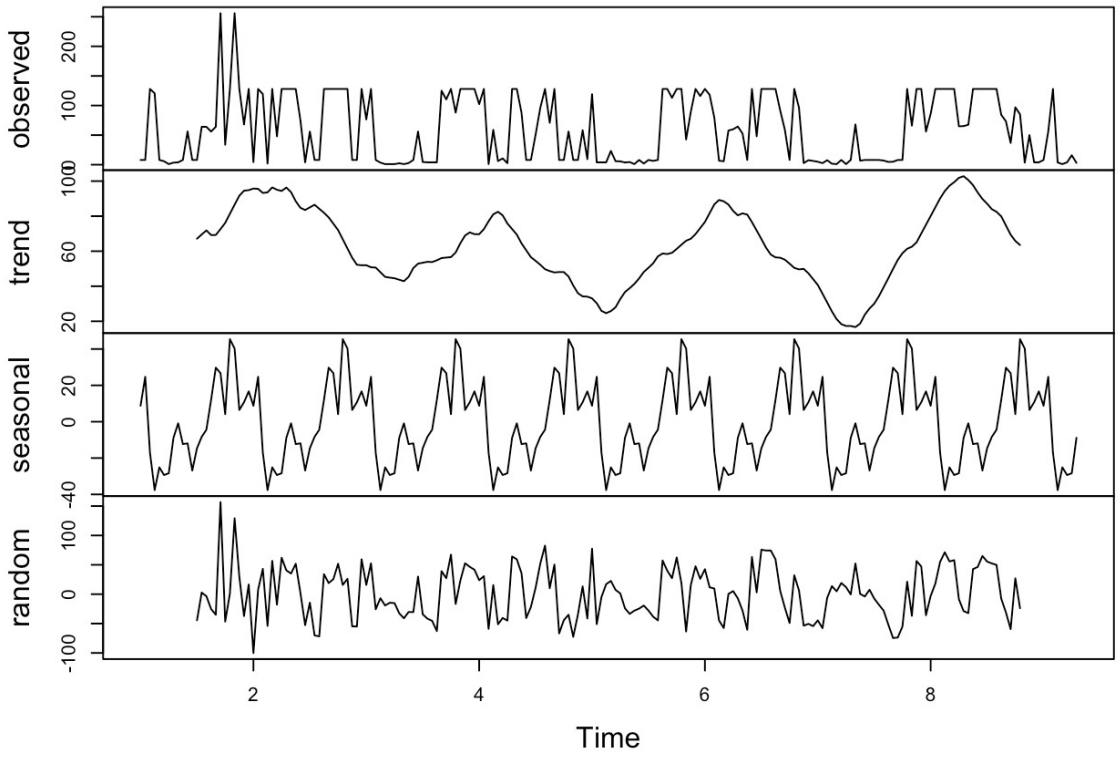


Series Decomposition

The decomposition choice will depend on the nature of the series.

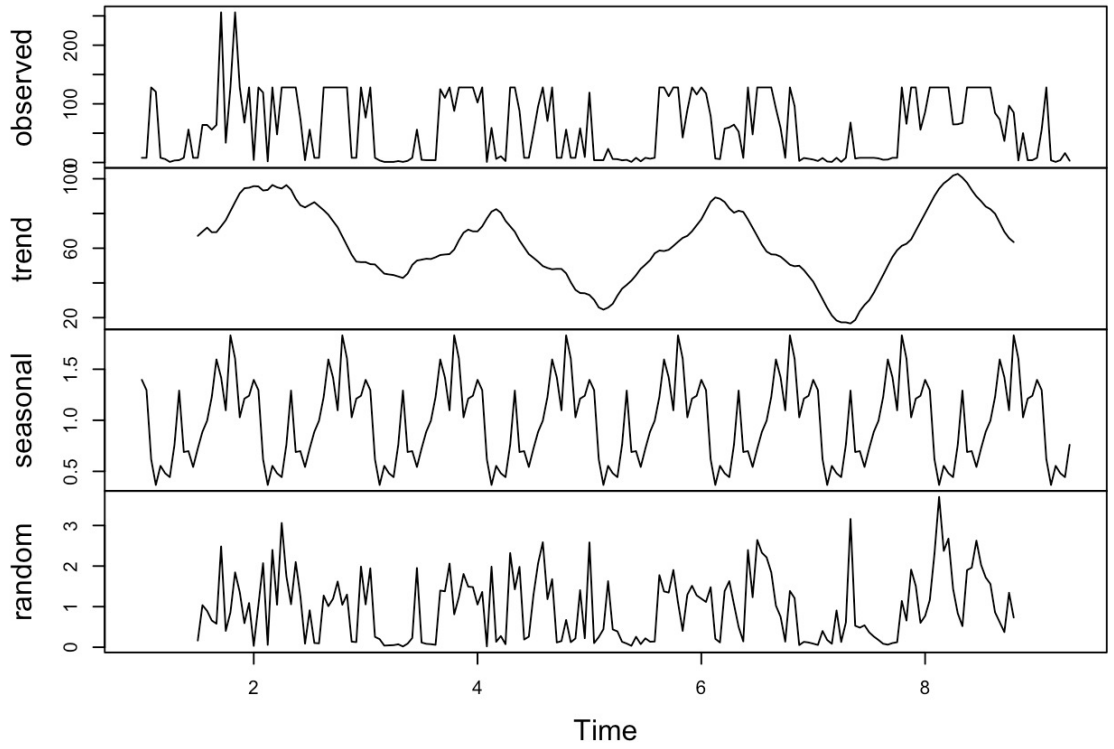
$$X_i = T_i + S_i + E_i$$

Decomposition of additive time series

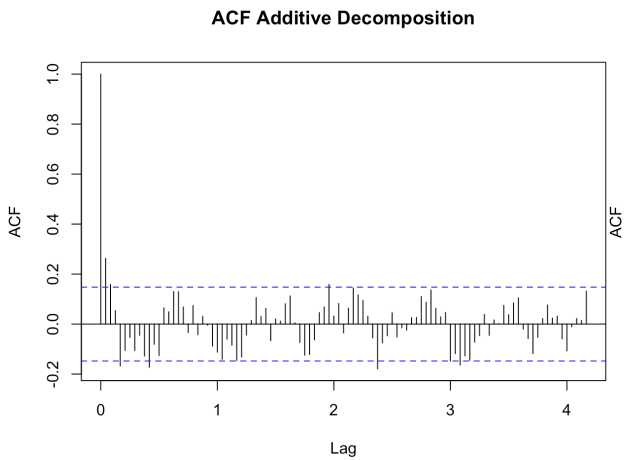


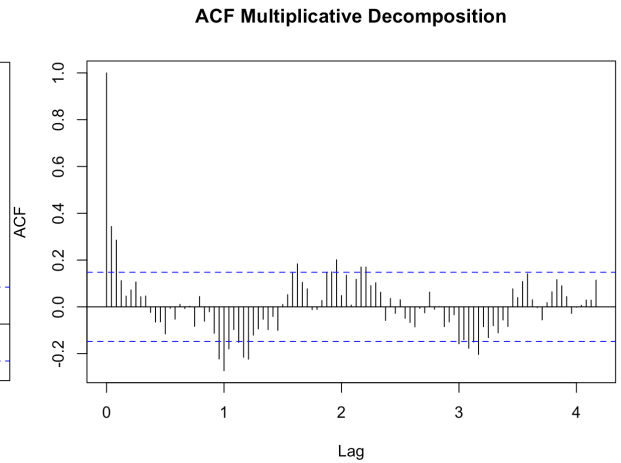
$$X_i = T_i * S_i * E_i$$

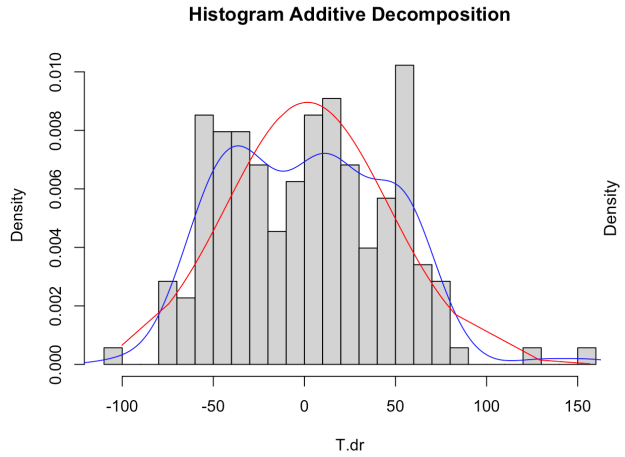
Decomposition of multiplicative time series

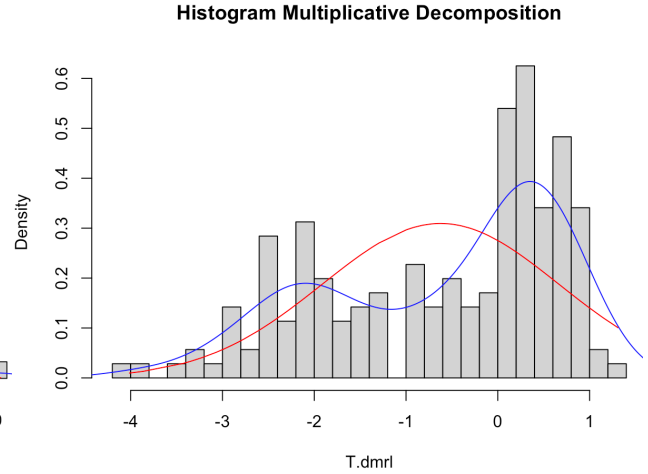


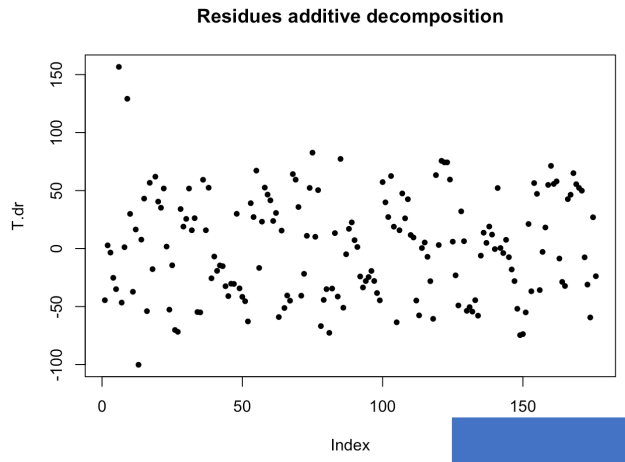
Series Decomposition Residues

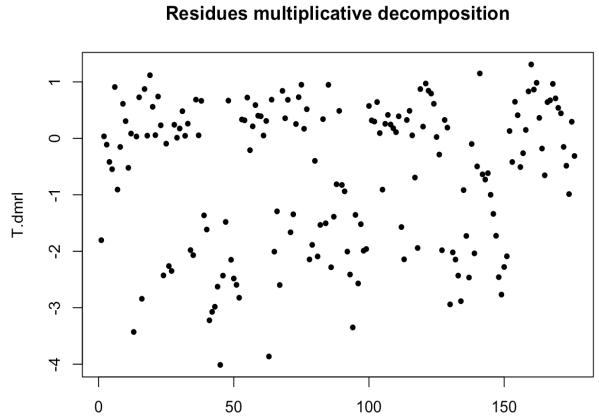


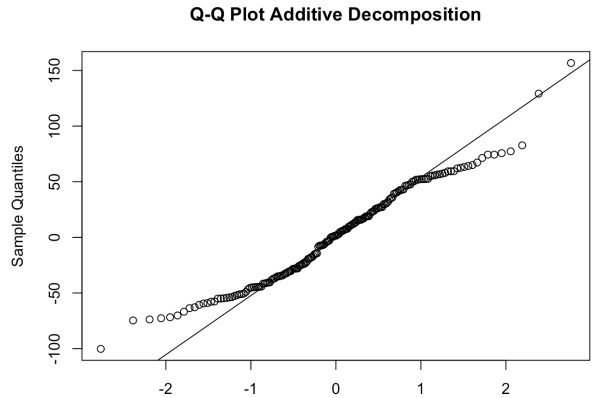


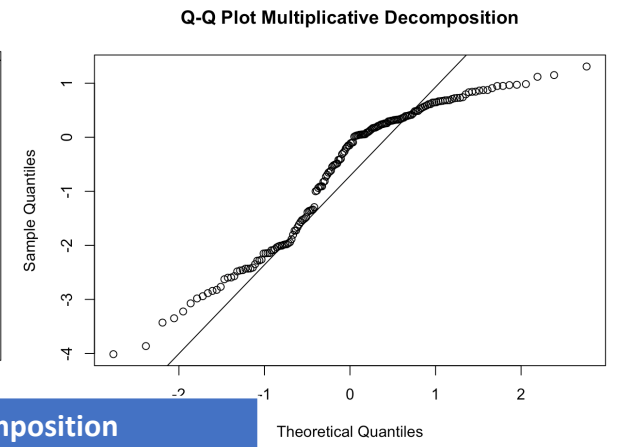












	Additive Decomposition	Multiplicative Decomposition
Non-explained variance	0.62	0.64
Shapiro-Wilk normality test p-value	0.002	1e^-08
Autocorrelation function's variability	0.24	0.23

milab

Holt-Winters Method (SETS)

(Level) $L_t = \alpha * (Y_t - S_{t-s}) + (1 - \alpha) * (L_{t-1} + b_{t-1})$

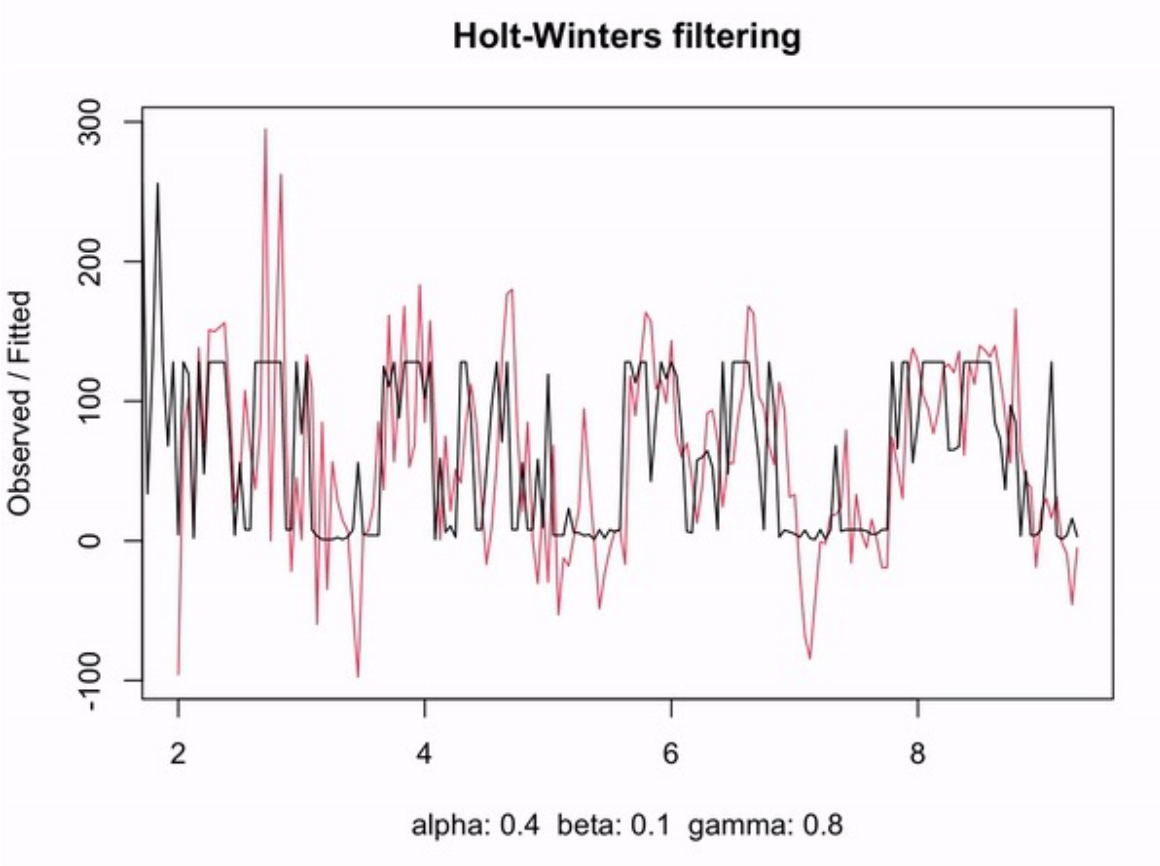
(Trend) $b_t = \beta * (L_t - L_{t-1}) + (1 - \beta) * b_{t-1}$

(Seasonal) $S_t = \gamma * (Y_t - L_t) + (1 - \gamma) * S_{t-s}$

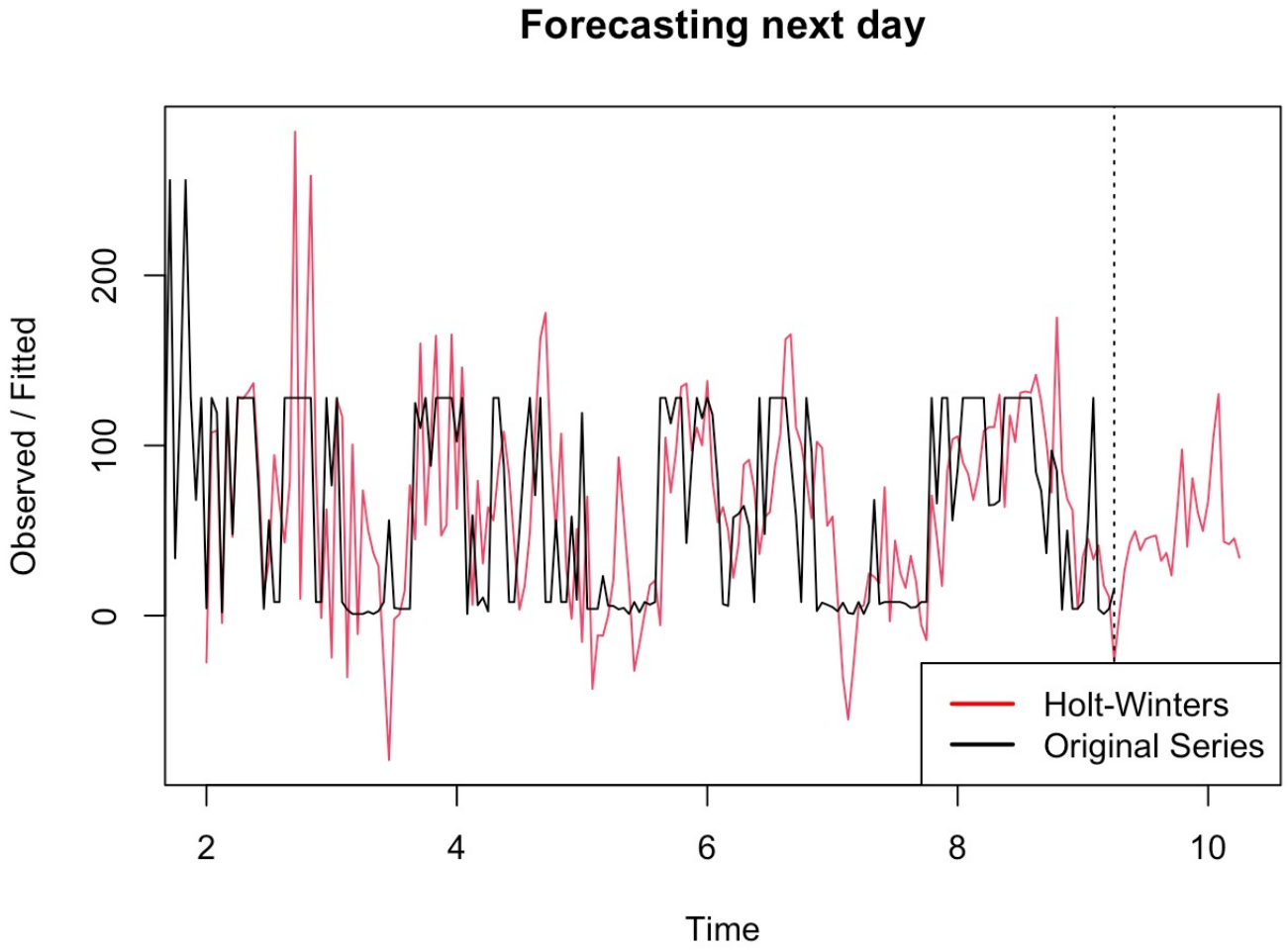
(Forecast for period m) $F_{t+m} = L_t + m*b_t + S_{t+m-s}$

	Model Choice
α	0.32
β	0
γ	0.72
Initial Intercept	-0.74
Initial Slope	5.43

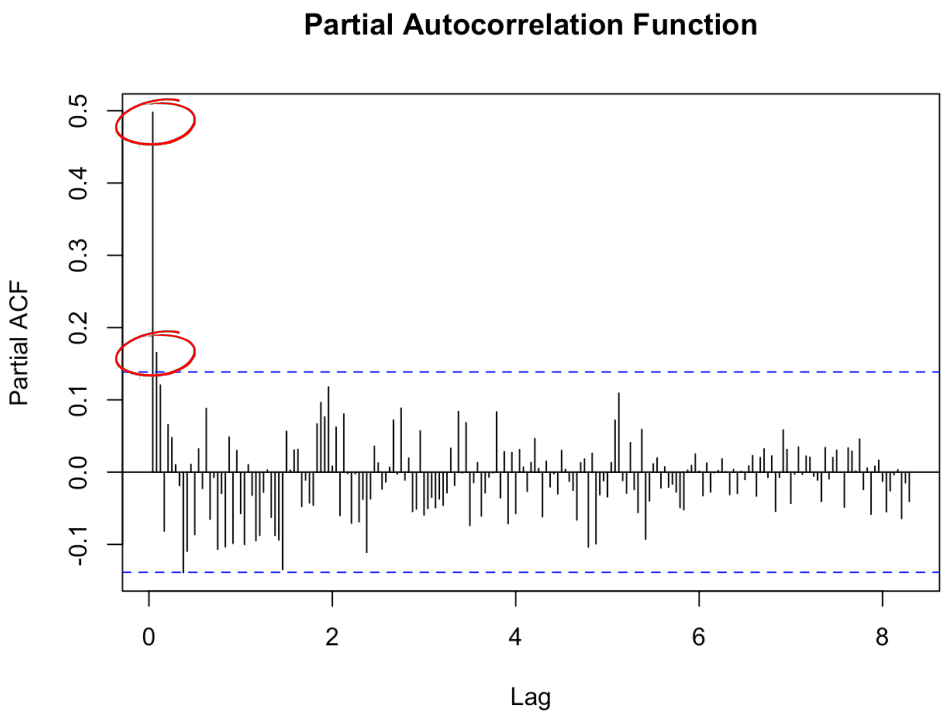
} obtained with a linear regression



Holt-Winters Additive Model Forecasting



Regression Methods for Time Series



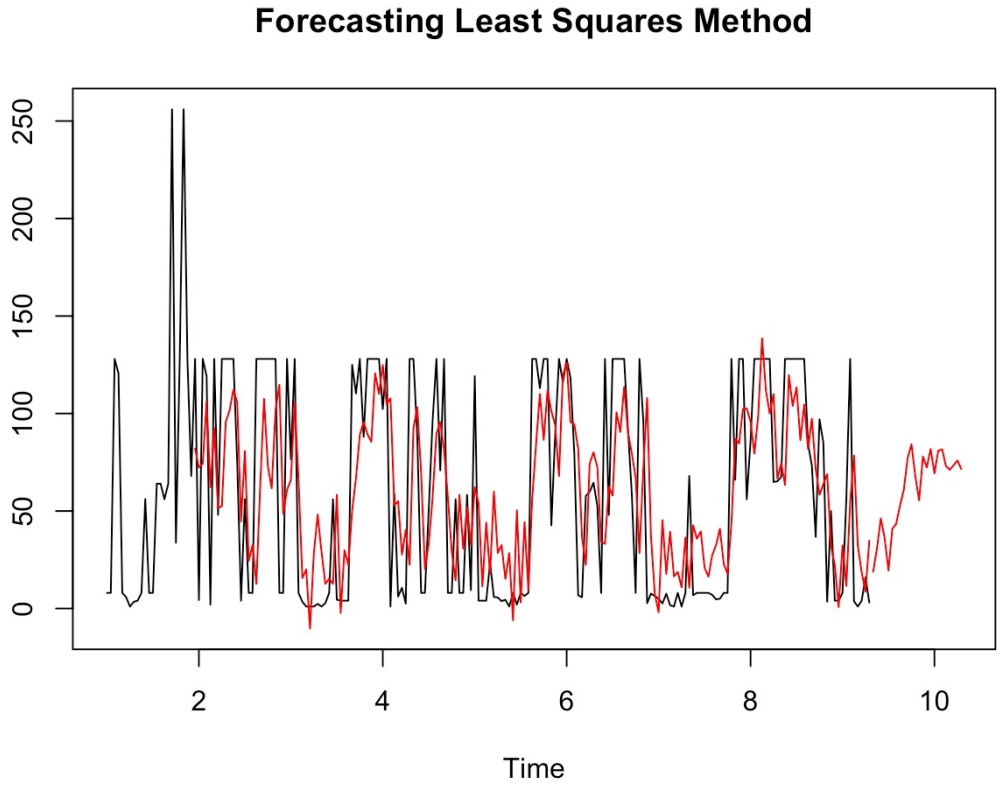
2 lags!

```
Call:
lm(formula = X3 ~ ., data = mnt)

Residuals:
    Min      1Q  Median      3Q     Max
-108.46  -28.32  -14.71   28.11  194.62

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 25.42046    5.52481   4.601 7.56e-06 ***
X1           0.16929    0.07058   2.399   0.0174 *
X2           0.41379    0.07062   5.859 1.95e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 48.57 on 195 degrees of freedom
Multiple R-squared:  0.2695,    Adjusted R-squared:  0.262
F-statistic: 35.97 on 2 and 195 DF,  p-value: 5.032e-14
```

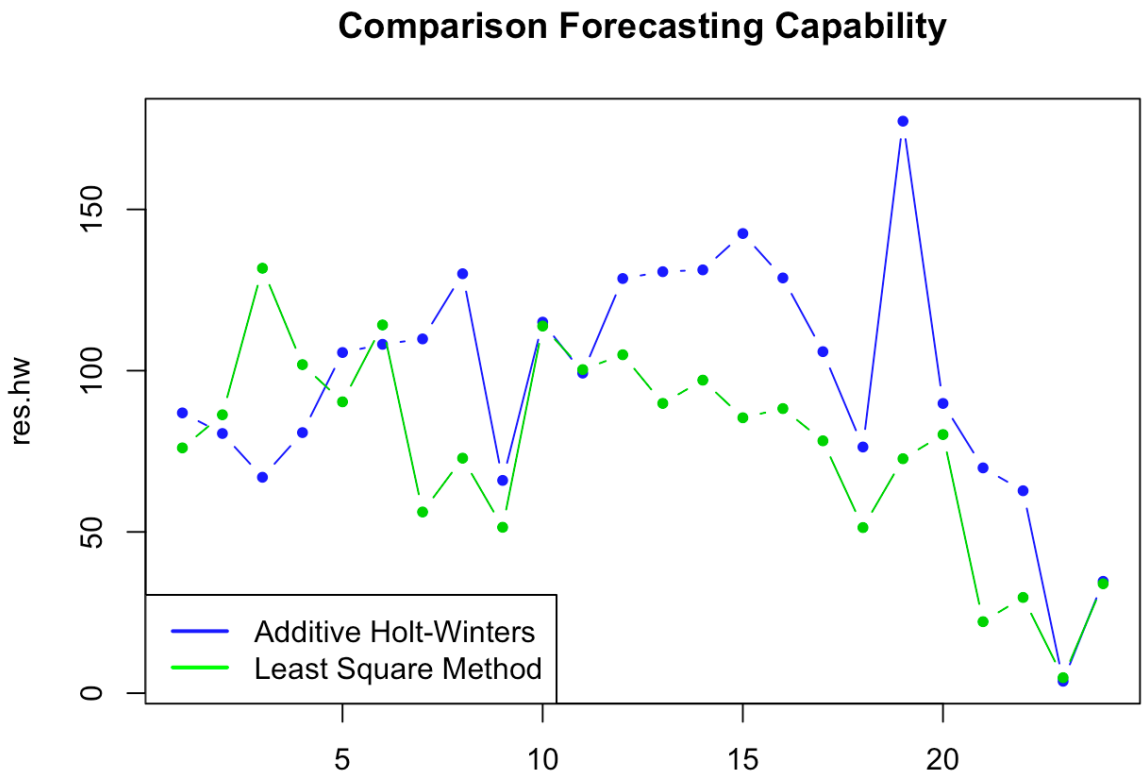
	X1	X2	X3
1	8.000000	8.000000	128.000000
2	8.000000	128.000000	120.653061
3	128.000000	120.653061	8.000000
4	120.653061	8.000000	5.833333
5	8.000000	5.833333	1.000000
6	5.833333	1.000000	3.454545
7	1.000000	3.454545	4.000000
8	3.454545	4.000000	8.000000
9	4.000000	8.000000	56.155844
10	8.000000	56.155844	8.000000
11	56.155844	8.000000	8.000000
12	8.000000	8.000000	64.000000
13	8.000000	64.000000	64.000000
14	64.000000	64.000000	56.000000
15	64.000000	56.000000	64.000000
16	56.000000	64.000000	256.000000
17	64.000000	256.000000	33.728938
18	256.000000	33.728938	128.000000
19	33.728938	128.000000	256.000000
20	128.000000	256.000000	128.000000
21	256.000000	128.000000	68.000000
22	128.000000	68.000000	128.000000
23	68.000000	128.000000	4.333333
24	128.000000	4.333333	128.000000
25	4.333333	128.000000	119.304348

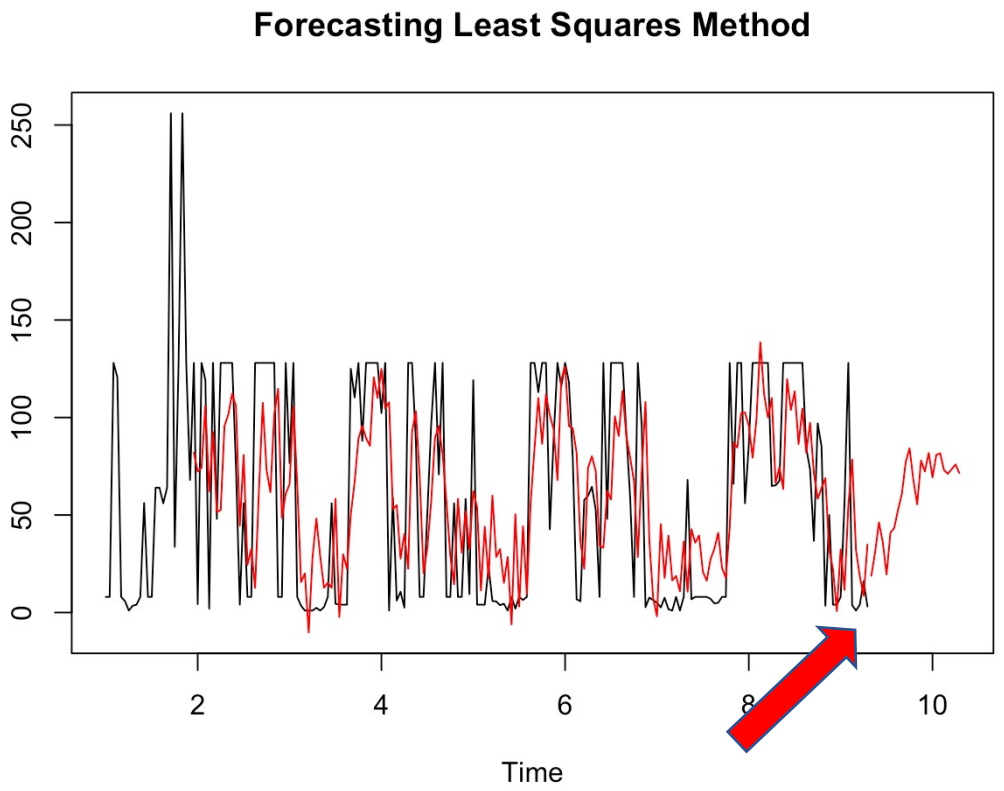
Least Square Regression Method

```
Coefficients:
      1        2        3        4        5        6        7        8        9       10       11
 0.4603   0.1552   0.0639  -0.1735   0.1635   0.0211   0.0147   0.0346  -0.0285  -0.1608   0.0681
     12       13       14       15       16       17       18       19       20       21       22
-0.0755   0.0530  -0.1062   0.1526   0.0072   0.0107  -0.1830   0.0859  -0.0693   0.0731  -0.1376
     23
 0.0295

Intercept: -0.1343 (3.092)

Order selected 23   sigma^2 estimated as  1666
```



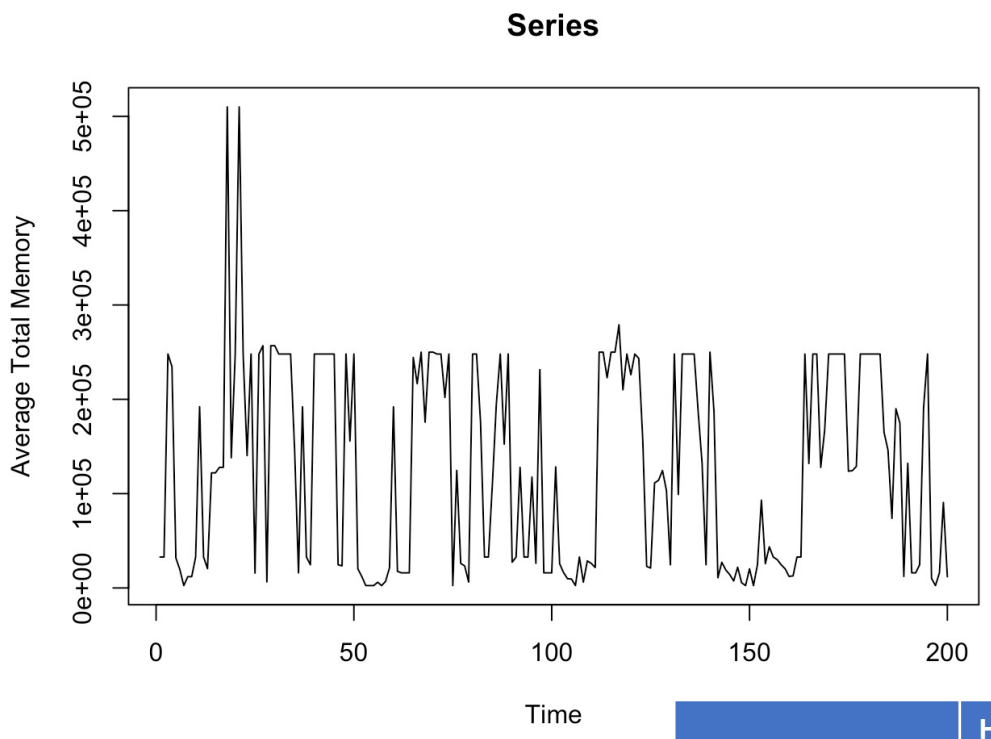
Holt-Winters Methods (SETS) Comparison

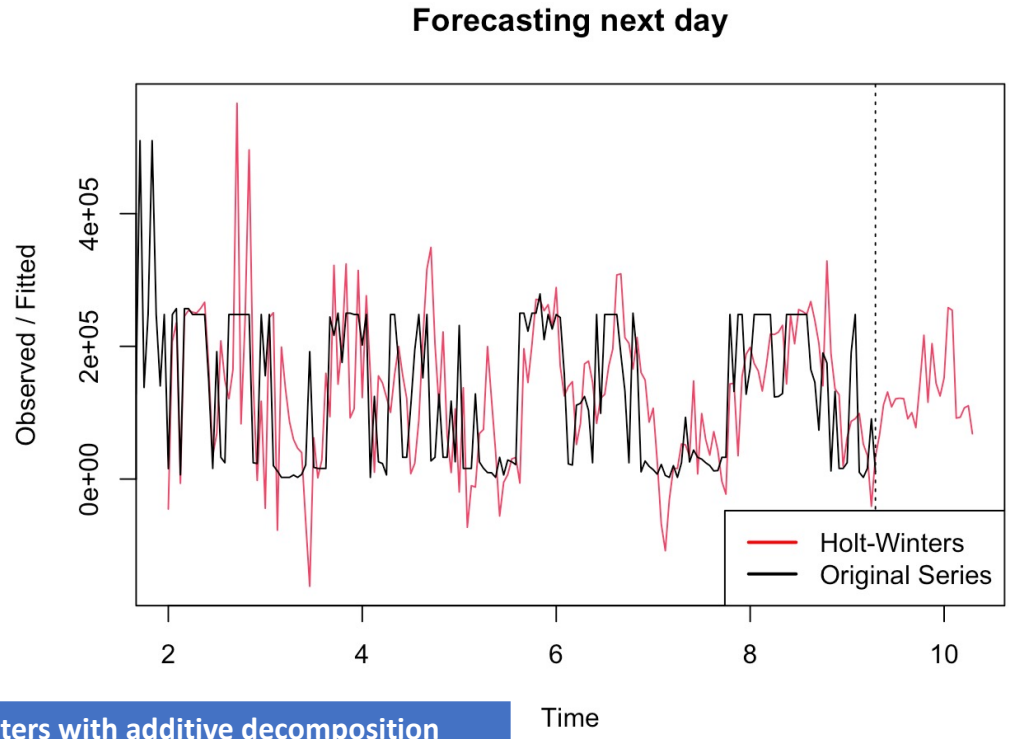




	Holt-Winters Additive Decomposition	**Least Square Method**
Mean Square Error	47.40	36.96

Time Series Analysis - Memory





	Holt-Winters with additive decomposition
α	0.30
β	0.03
γ	0.70
Initial Intercept	11352
Initial Slope	10580

Complete time series analysis on memory not reported in this presentation.

CPU, Memory and Probability of Failure

✓ Predict the amount of CPU and Memory that is going to be provided from each site

$$CPUProvided,\ MemoryProvided$$

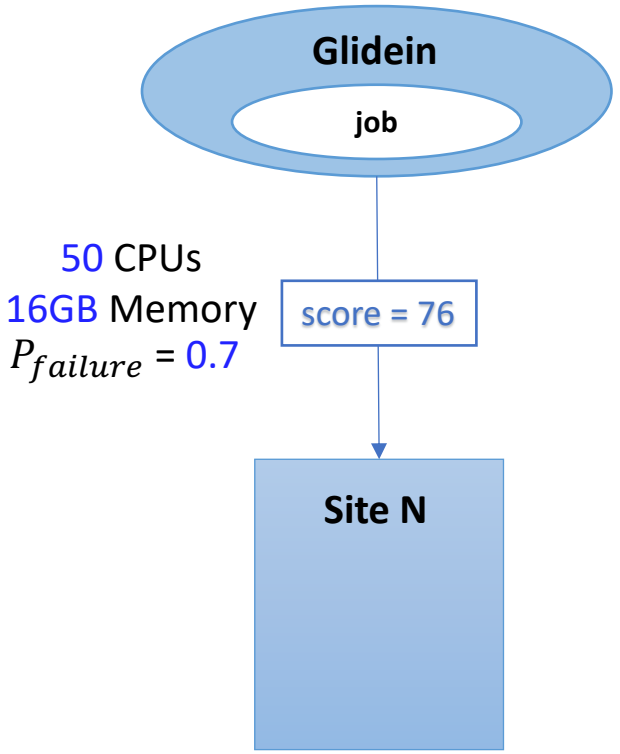
✓ Consider the sites for which the resources provided are enough for our job

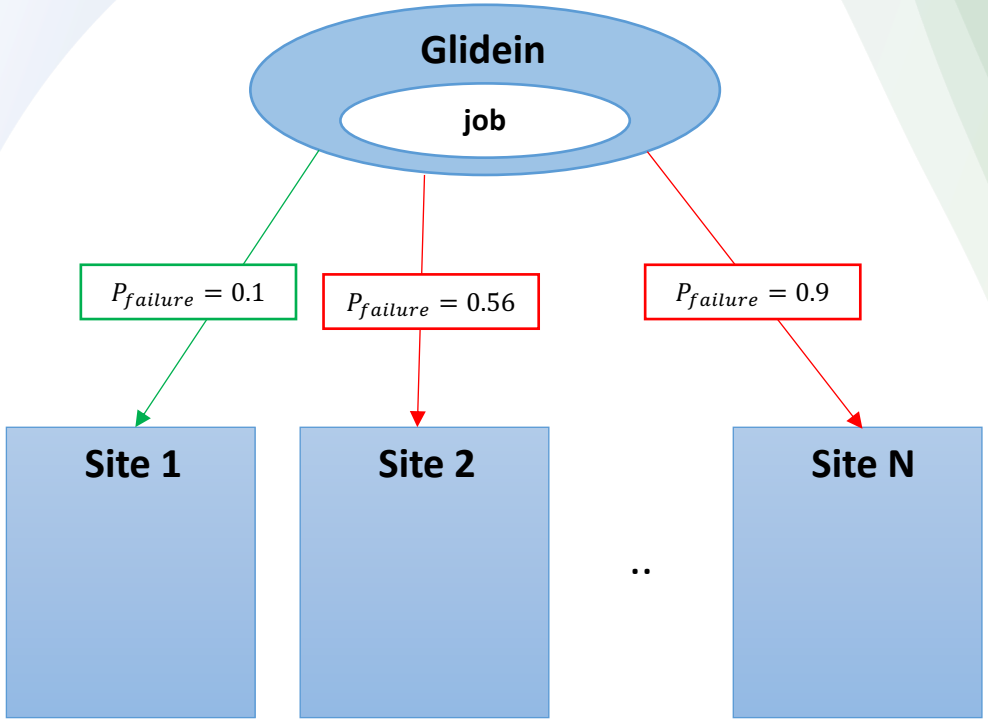
$$CPURequested \leq CPUProvided$$
$$MemoryRequested \leq MemoryProvided$$

- Calculate the probability of failure of each site

$$P_{failure}$$

- Calculate a cumulative score that allows us to take this decision





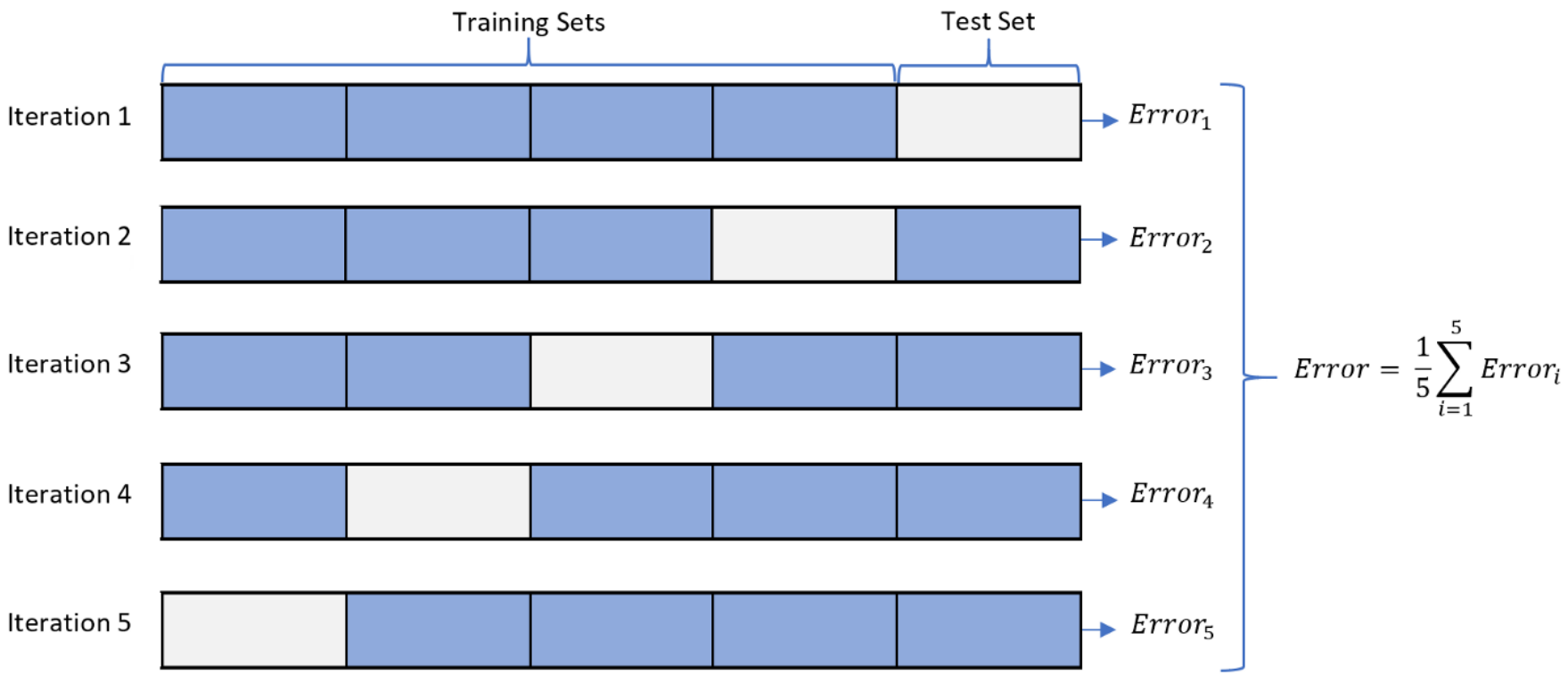
Failure Prediction

Use probabilistic classification methods to predict the failure of a node in a certain site and obtain the probability of failure.

- Linear Regression for Classification
- Logistic Regression
- Linear Discriminant Analysis
- Quadratic Discriminant Analysis

	DiskUsage	TotalCpus	Total Memory	TotalDisk	CpuCache Size	TotalVirtual Memory	JobMax Time	TotalSlots	CpuIsBusy	SlotType		Failure
Sample 1	…	…	…	…	…	…	…	…	…	…	→	…

Failure Prediction Cross-Validation Comparison

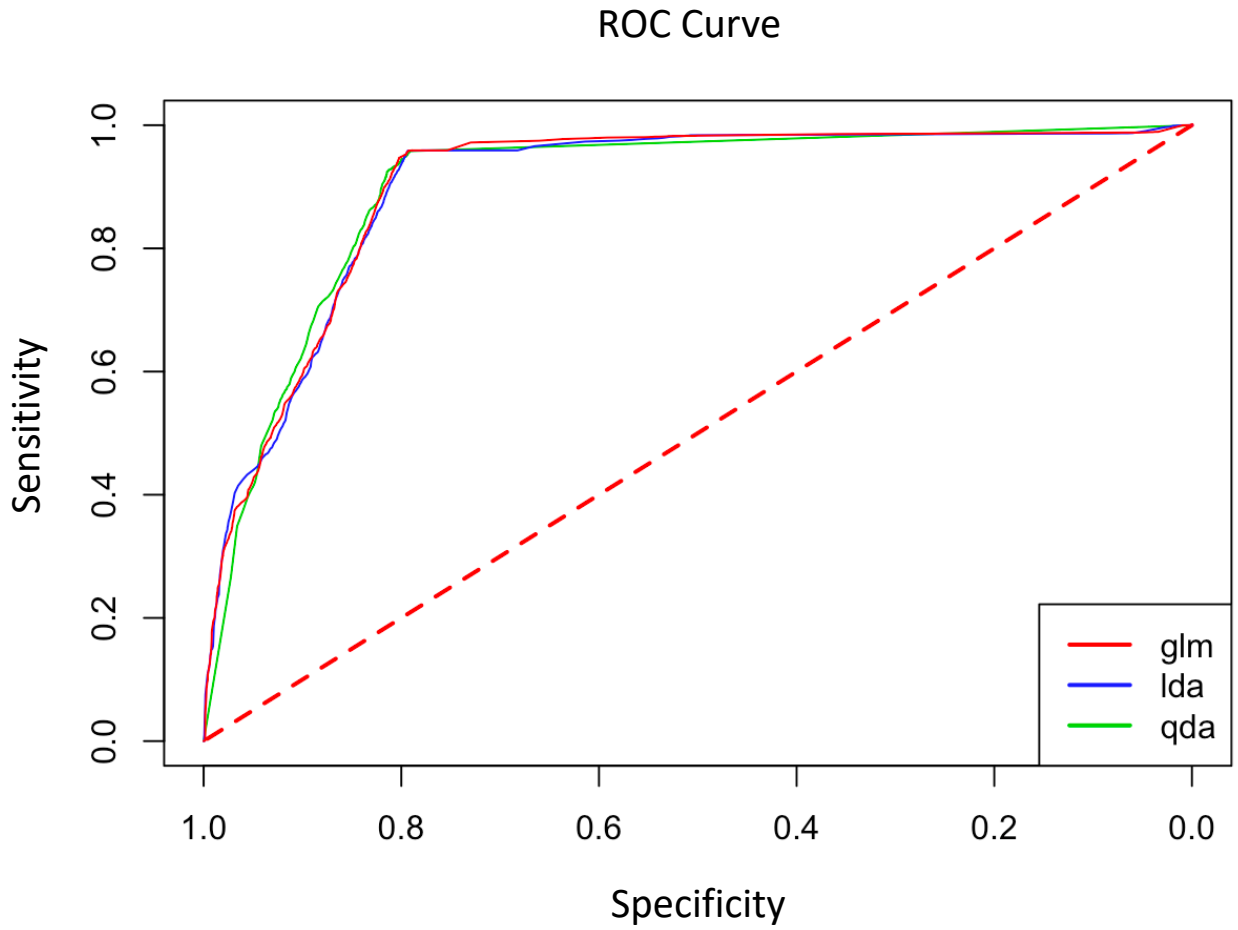


Failure Prediction Cross-Validation Comparison

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

	Accuracy's Mean	Accuracy's Std	AUC
Linear Regression	0.33	0.07	-
Logistic Regression	0.84	0.04	0.90
Linear Discriminant Analysis	0.83	0.04	0.91
Quadratic Discriminant Analysis	0.85	0.04	0.90

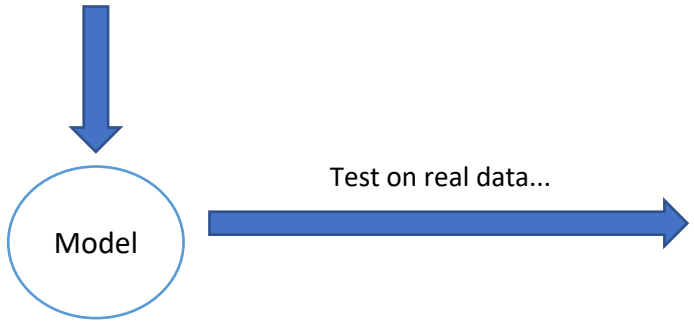
Site Cross Validation	Accuracy Mean	Accuracy Std
Logistic Regression	0.75	0.23

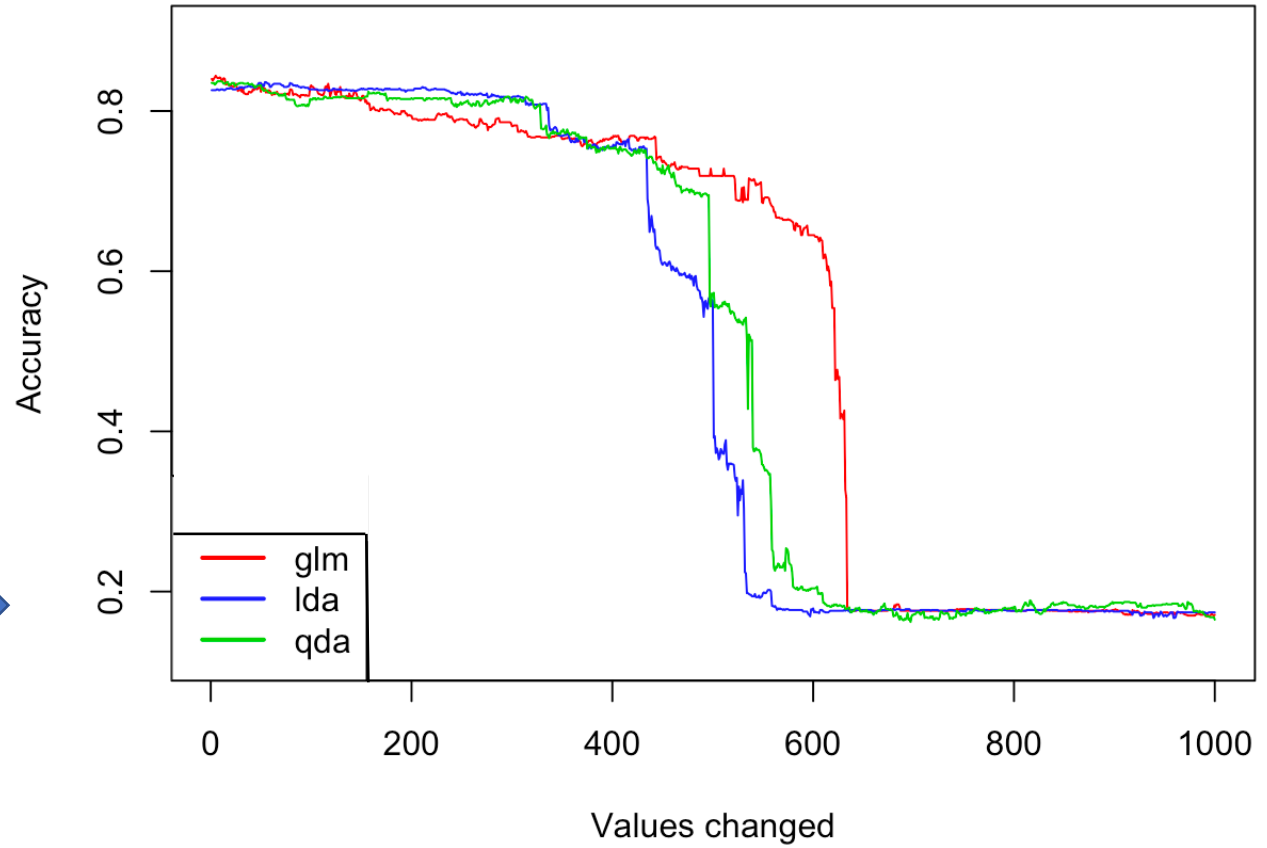


Models' Robustness Comparison

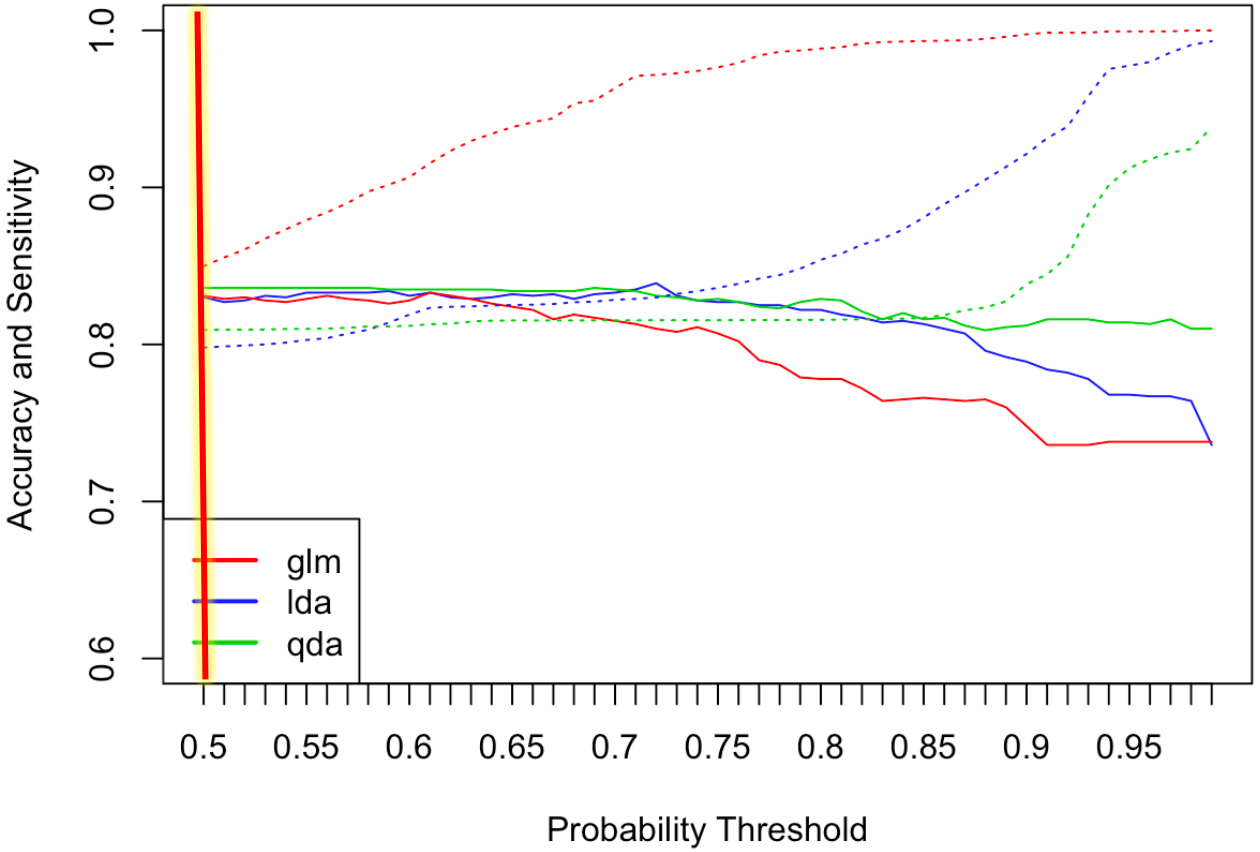
Robustness analysis is performed by introducing false information in order to determine how robust the model is.

	Attribute 1	….	Failure
Sample 1	….	….	~~1~~ 0
Sample 2	….	….	~~0~~ 1
….	….	….	….
Sample N	….	….	~~1~~ 0



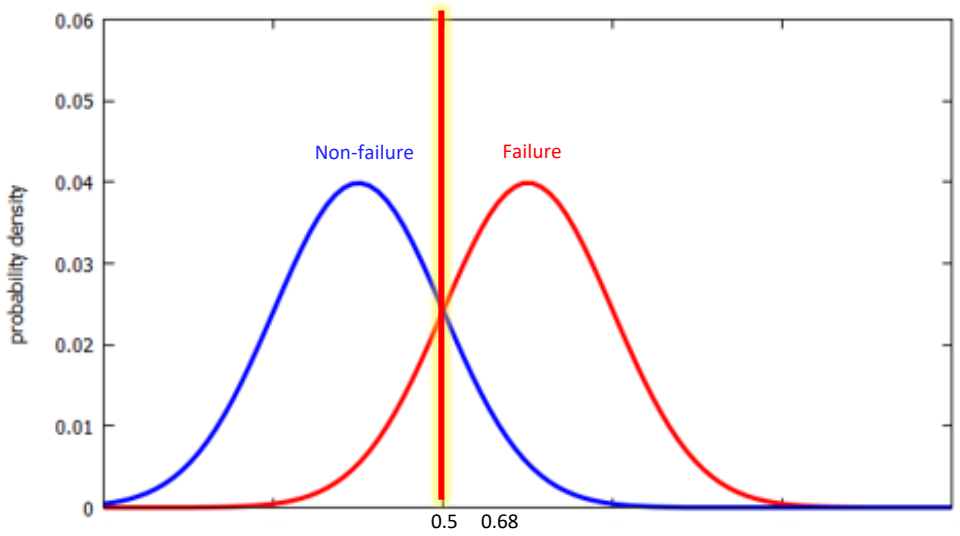


Probability Threshold Choice



Choice of a different threshold value than the one provided by the model (½) to improve the prediction and achieve a good trade-off between accuracy and sensitivity.

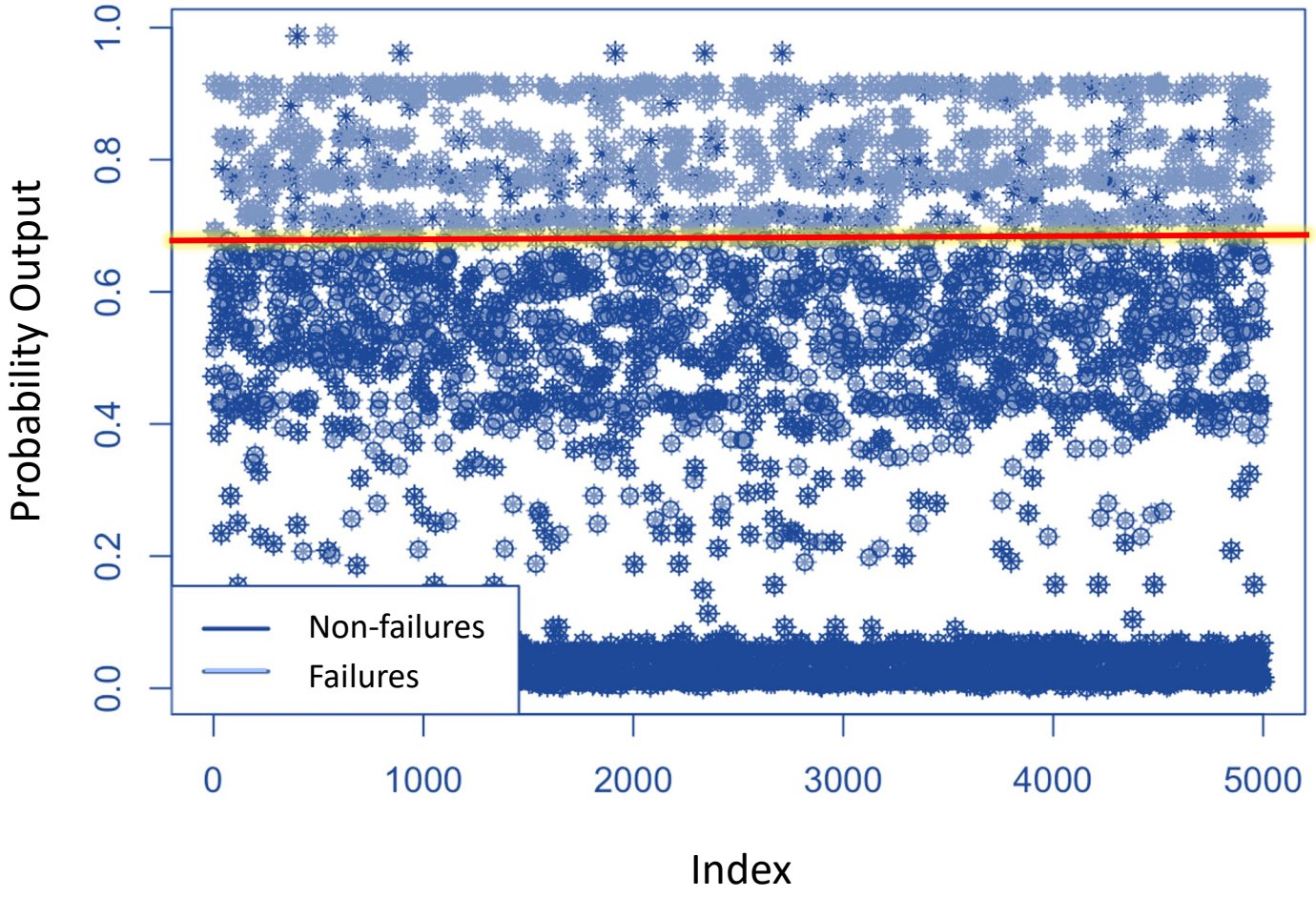
$$Sensitivity = \frac{TP}{P} \qquad Accuracy = \frac{TP + TN}{P + N}$$



Logistic Regression Model

Training set results	Accuracy	Sensitivity
Logistic Regression with $P_{threshold}$ = 0.68	0.81	0.93

	Actual Positive	Actual Negative
Predicted Positive	6554	3626
Predicted Negative	486	15117



Prediction Workflow

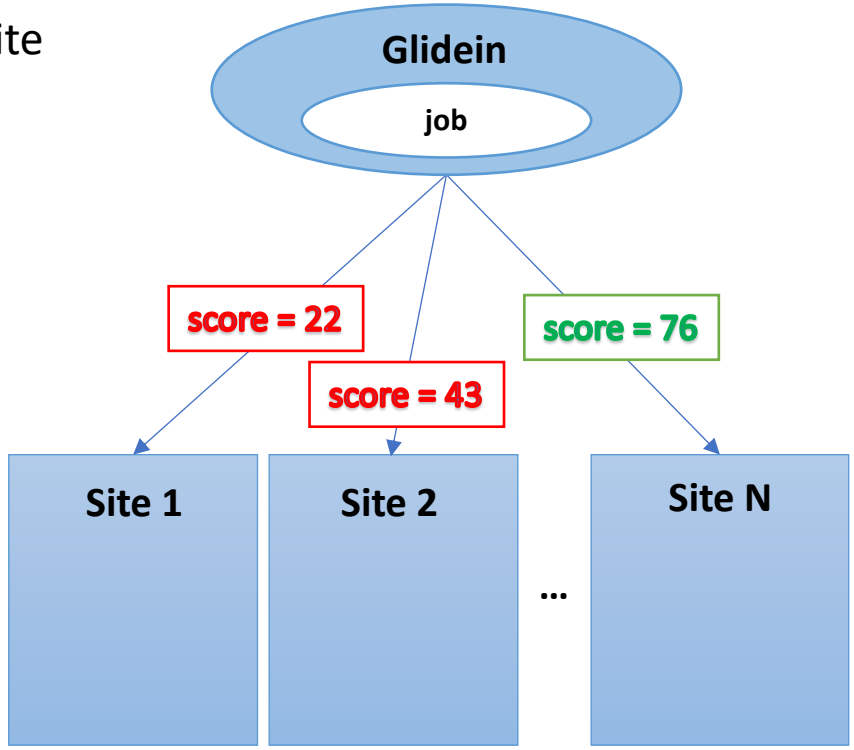
- Calculate a cumulative score that allows us to determine the best matching site

$$resources\_score = \frac{CPU - \min(CPUs)}{\max(CPUs) - \min(CPUs)} * 50$$

$$+ \frac{Memory - \min(Memories)}{\max(Memories) - \min(Memories)} * 50$$

$$score = resources_{score} * (1 - P_{failure})$$

$$score \in [0, 100]$$



- Choose the site with the highest score!