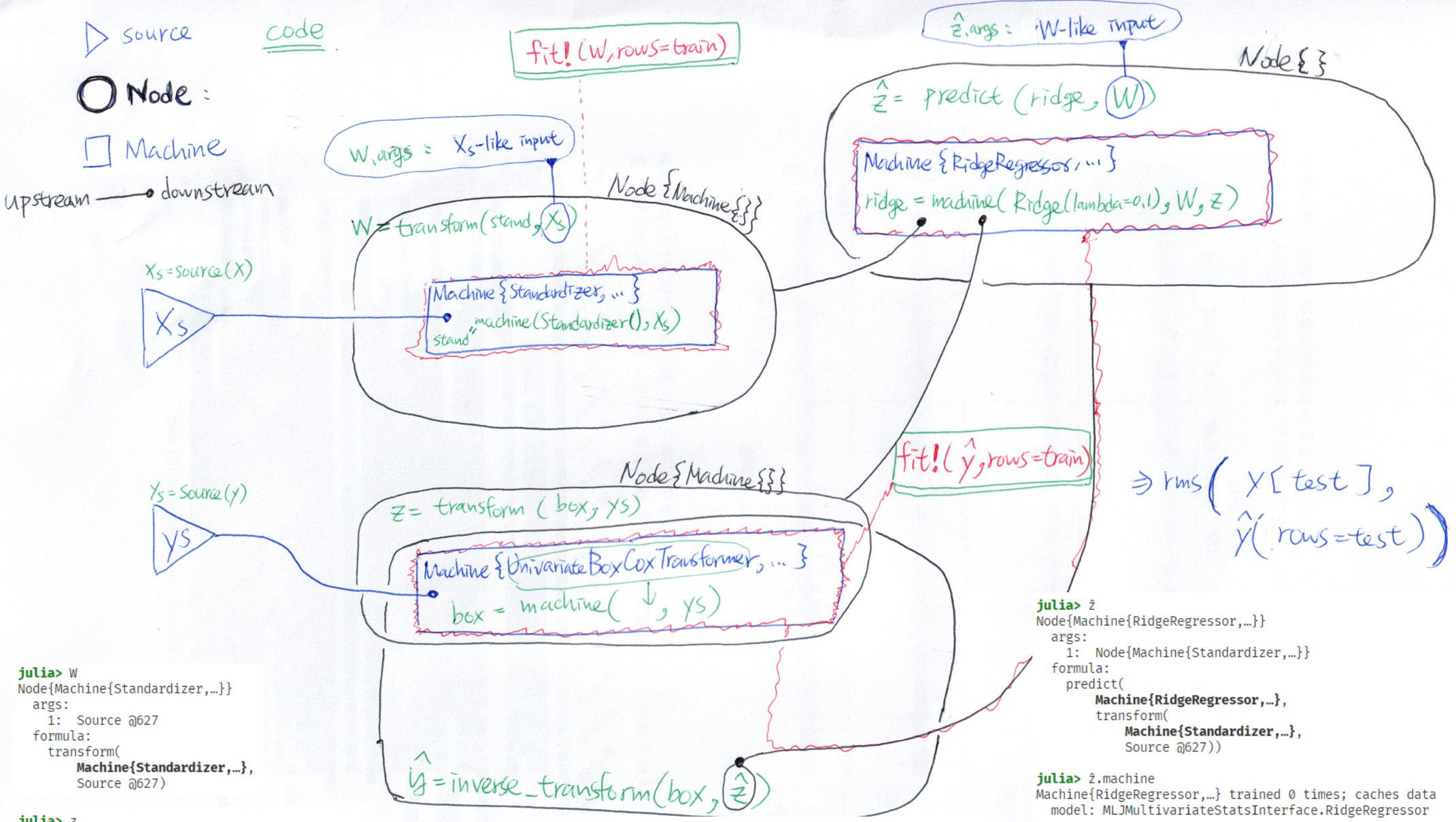


```
julia> W
Node{Machine{Standardizer,…}}
  args:
    1:  Source @627
  formula:
    transform(
        Machine{Standardizer,…},
        Source @627)
```

```
julia> z
Node{Machine{UnivariateBoxCoxTransformer,…}}
  args:
    1:  Source @198
  formula:
    transform(
        Machine{UnivariateBoxCoxTransformer,…},
        Source @198)
```

```
julia> box
Machine{UnivariateBoxCoxTransformer,…} trained 1 time; caches
data
  model: UnivariateBoxCoxTransformer
  args:
    1:  Source @198 ⏎ `AbstractVector{Continuous}`
```

```
julia> ridge
Machine{RidgeRegressor,…} trained 1 time; caches data
  model: MLJMultivariateStatsInterface.RidgeRegressor
  args:
    1:  Node{Machine{Standardizer,…}}
    2:  Node{Machine{UnivariateBoxCoxTransformer,…}}
```

```
julia> ẑ
Node{Machine{RidgeRegressor,…}}
  args:
    1:  Node{Machine{Standardizer,…}}
  formula:
    predict(
        Machine{RidgeRegressor,…},
        transform(
            Machine{Standardizer,…},
            Source @627))
```

```
julia> ẑ.machine
Machine{RidgeRegressor,…} trained 0 times; caches data
  model: MLJMultivariateStatsInterface.RidgeRegressor
  args:
    1:  Node{Machine{Standardizer,…}}
    2:  Node{Machine{UnivariateBoxCoxTransformer,…}}
```

```
julia> ŷ
Node{Machine{UnivariateBoxCoxTransformer,…}}
  args:
    1:  Node{Machine{RidgeRegressor,…}}
  formula:
    inverse_transform(
        Machine{UnivariateBoxCoxTransformer,…},
        predict(
            Machine{RidgeRegressor,…},
            transform(
                Machine{Standardizer,…},
                Source @627)))
```

Node: "A function" that uses trained machine

Reference:
https://juliaai.github.io/DataScienceTutorials.jl/getting-started/learning-networks/
https://juliaai.github.io/DataScienceTutorials.jl/getting-started/learning-networks-2/