A TUTORIAL ON MACHINE LEARNING WITH WEKA

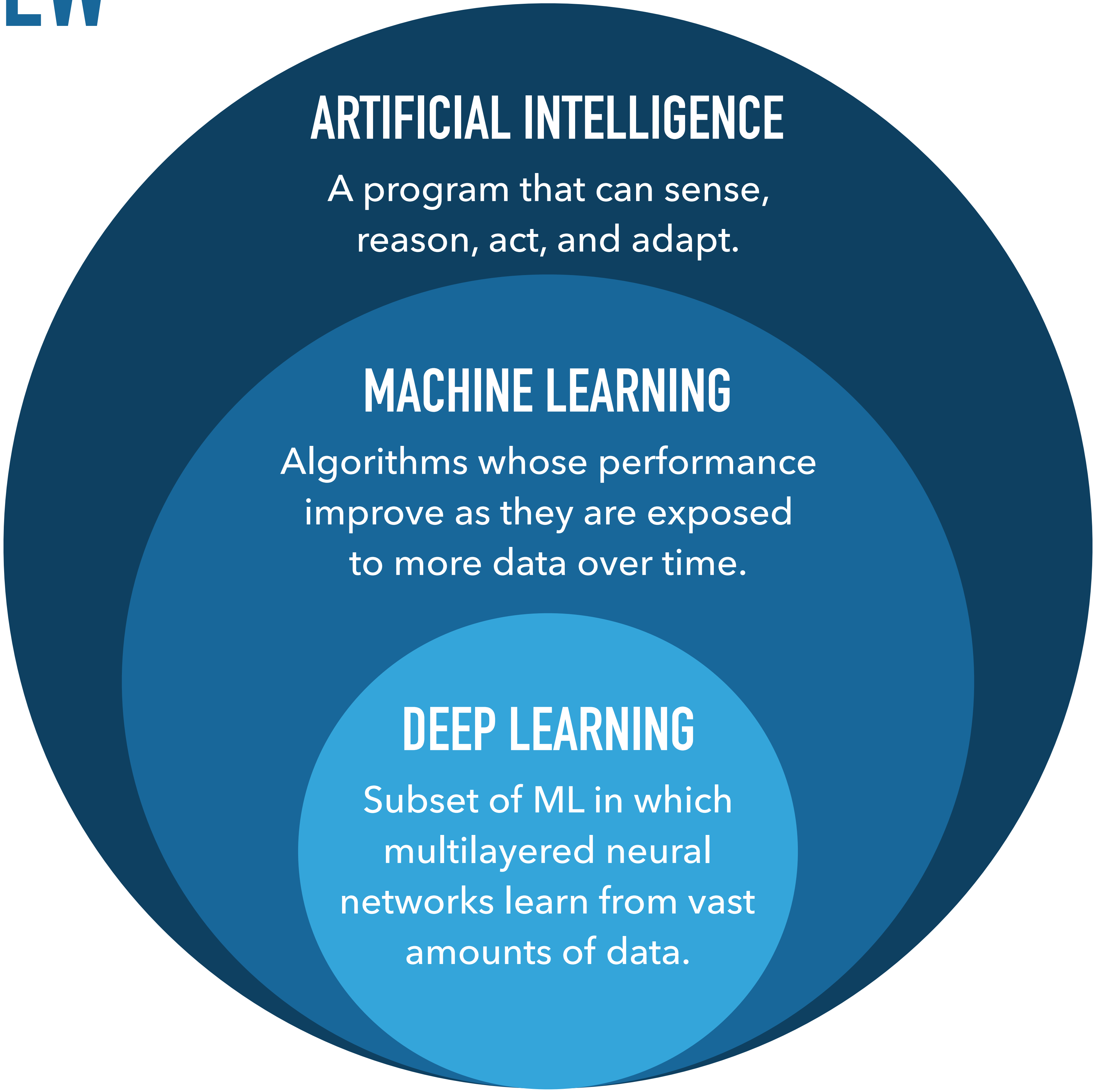
STEFANO PIO ZINGARO

PH.D. STUDENT IN COMPUTER SCIENCE AND ENGINEERING

STEFANOPIO.ZINGARO [AT] UNIBO.IT

CS.UNIBO.IT/~STEFANOPIO.ZINGARO

OVERVIEW



ARTIFICIAL INTELLIGENCE

A program that can sense, reason, act, and adapt.

MACHINE LEARNING

Algorithms whose performance improve as they are exposed to more data over time.

DEEP LEARNING

Subset of ML in which multilayered neural networks learn from vast amounts of data.

BASIC ASSUMPTIONS FOR THE TUTORIAL

SOME KNOWLEDGE ON…

- The concept of independent and dependent variables;
- The concept of error in statistic;
- The concepts of Supervised and Unsupervised Learning;
- Boolean variables (TRUE and FALSE).

BUT NO KNOWLEDGE ON…

- Programming Languages;
- Coding;
- Machine Learning techniques;
- Neural Networks;
- Minimisation or Maximisation of a function.

MACHINE LEARNING VS. DEEP LEARNING

CLASSIC MACHINE LEARNING

How do you engineer the best features?

FEATURES

- Roundness of face;
- Distance between eyes;
- Nose width;
- Eye socket depth;
- Cheek bone structure;
- Jaw line length…

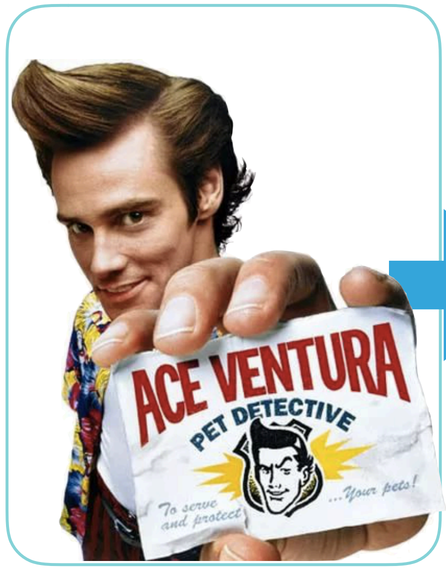
CLASSIFIER ALGORITHMS

- SVM;
- Random Forest;
- Naïve Bayes;
- Decision Tree;
- Regression
- …many more…

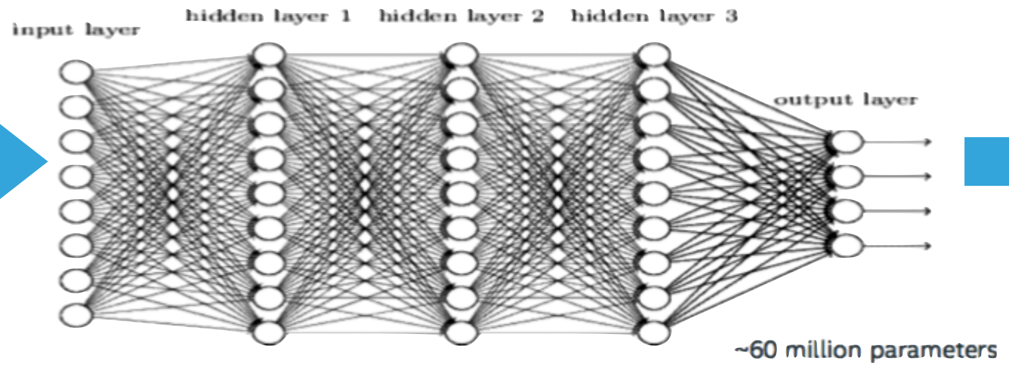
Jim Carrey

DEEP LEARNING

How do you guide the model to find the best features?



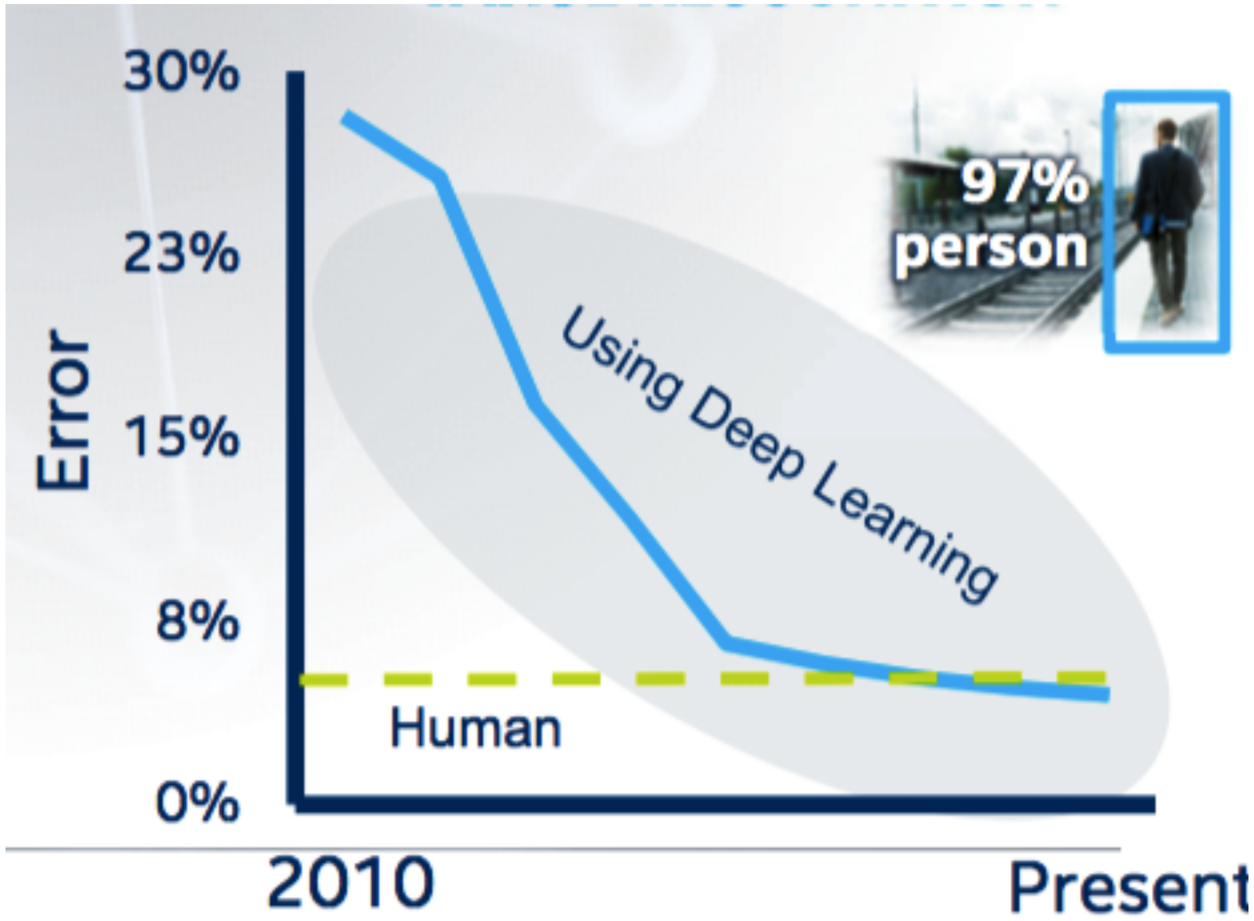
DEEP NEURAL NETWORK



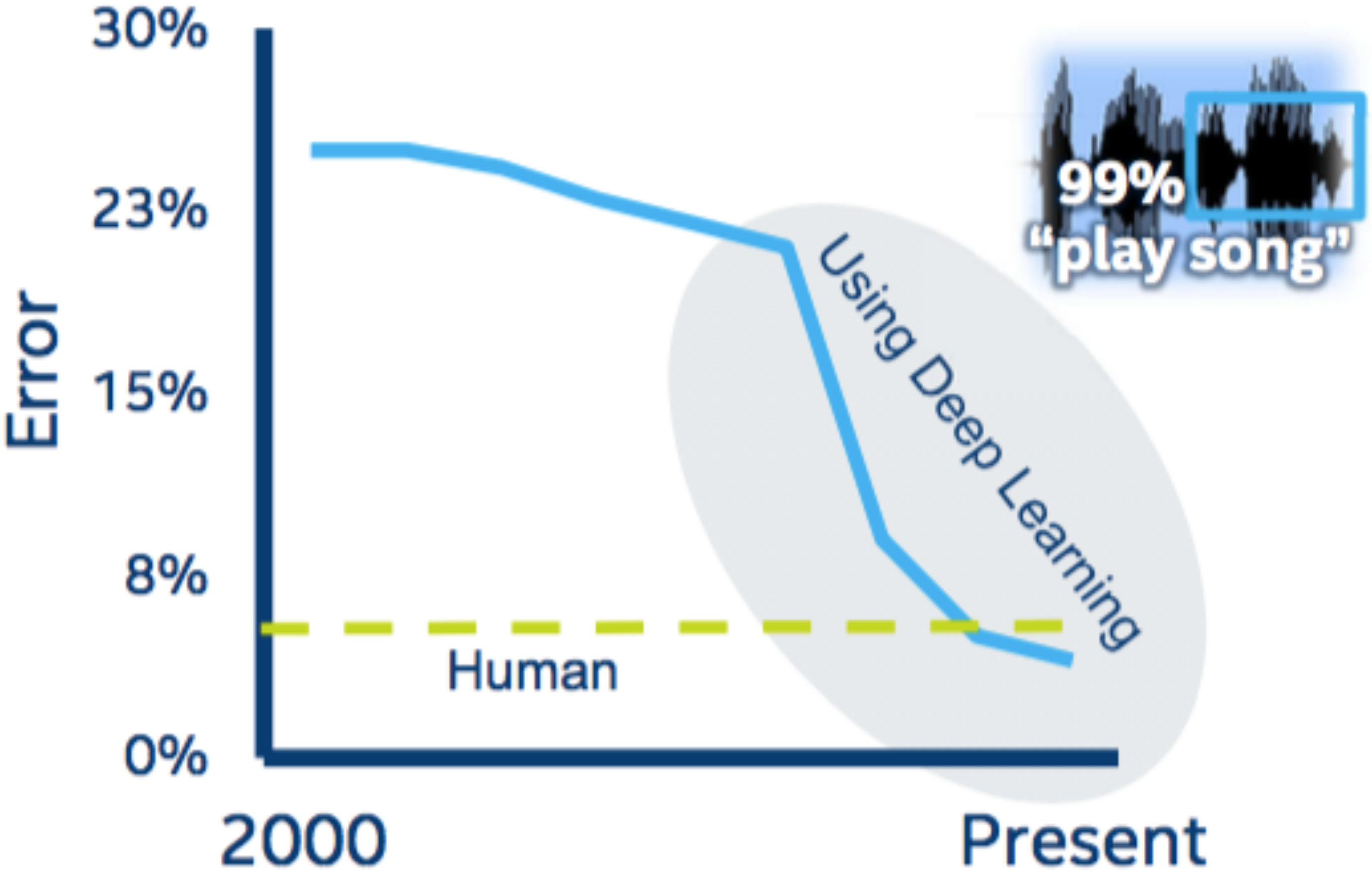
Jim Carrey

DEEP LEARNING BREAKTHROUGHS

IMAGE RECOGNITION



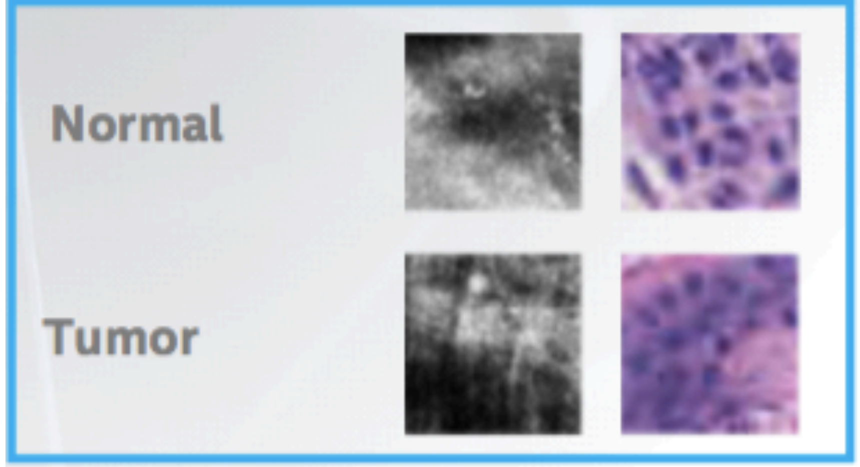
SPEECH RECOGNITION



MACHINES ABLE TO MEET OR EXCEED HUMAN IMAGE & SPEECH RECOGNITION (TO SOME EXTEND…)

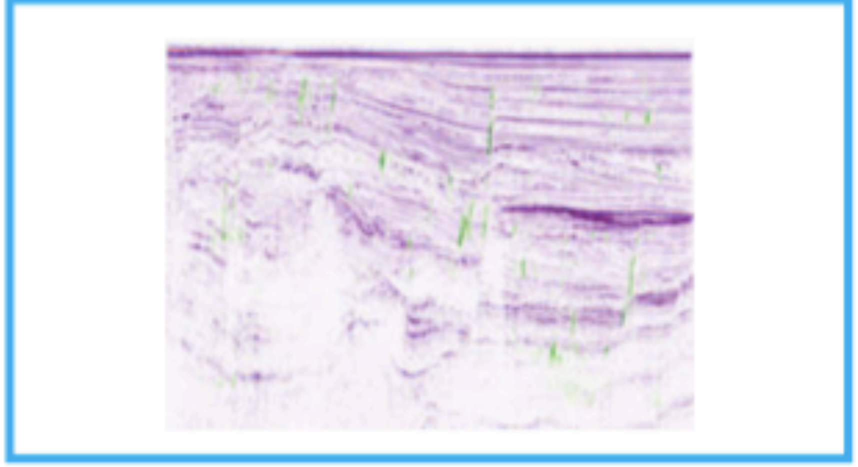
DEEP LEARNING IN PRACTICE

Healthcare: Tumor detection



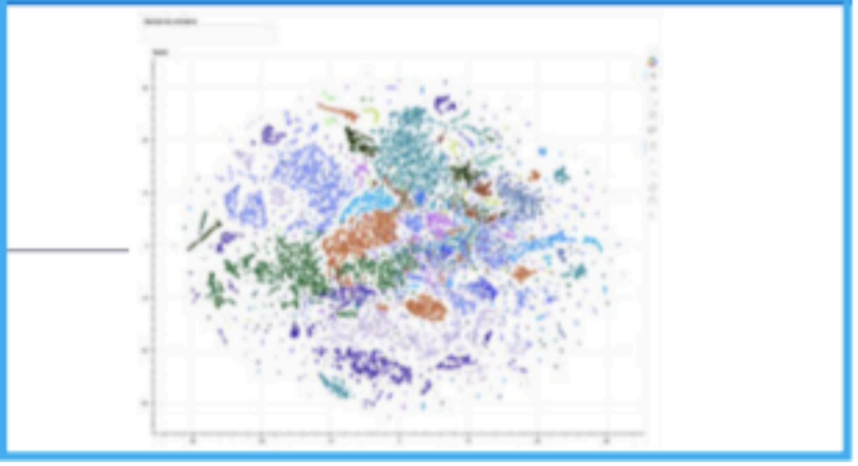
Industry: Agricultural Robotics

Energy: Oil & Gas

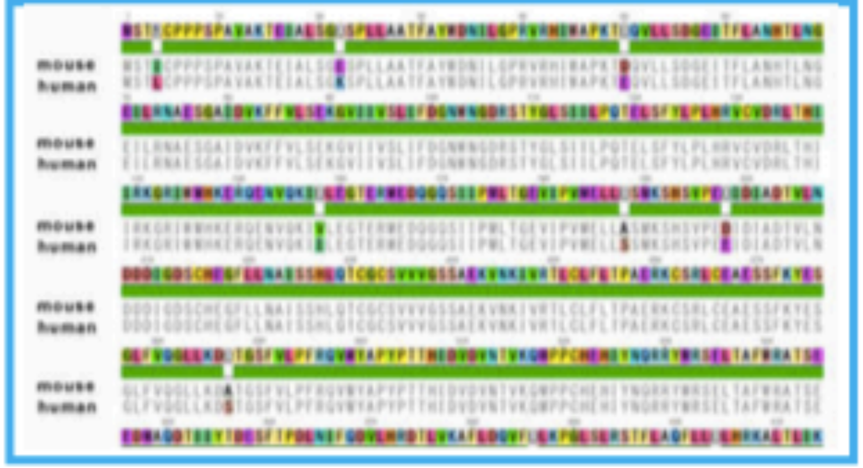


Automotive: Speech interfaces

Finance: Document Classification



Genomics: Sequence analysis



FEED FORWARD NEURAL NETWORKS

FEED FORWARD NEURAL NETWORKS

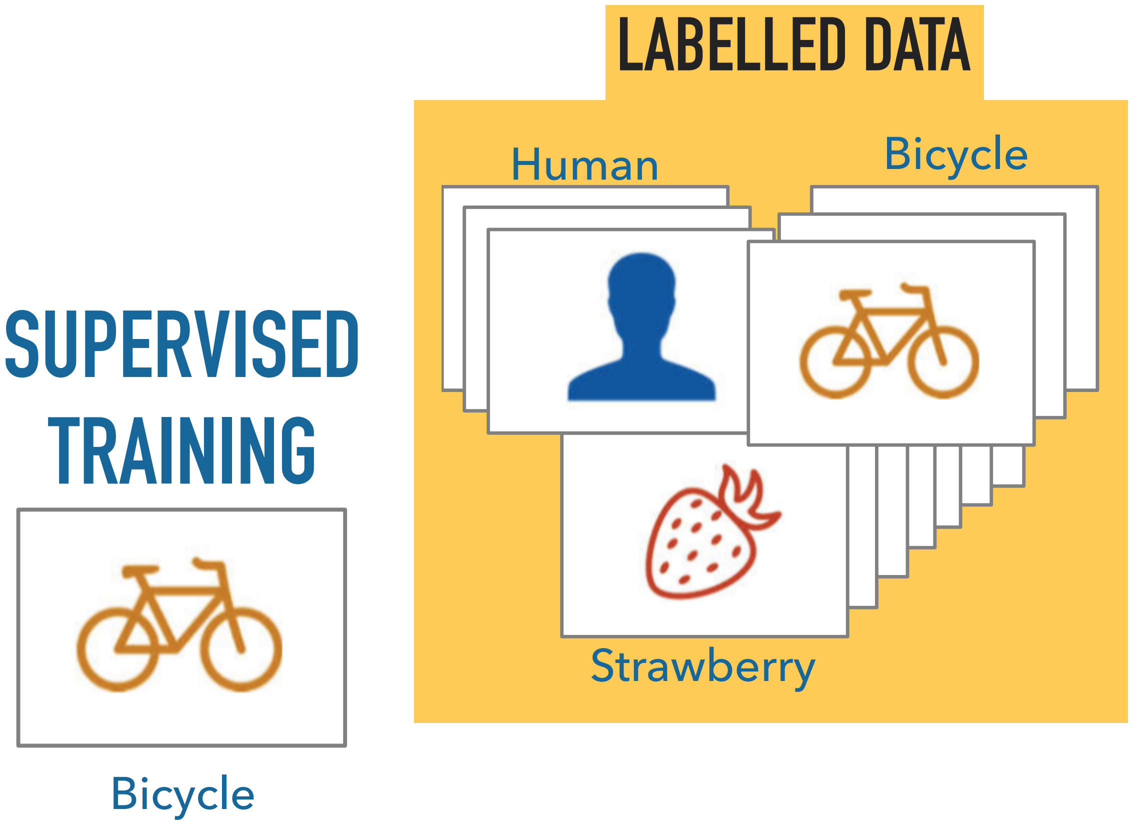
SUPERVISED TRAINING

FEED FORWARD NEURAL NETWORKS

SUPERVISED TRAINING

Bicycle

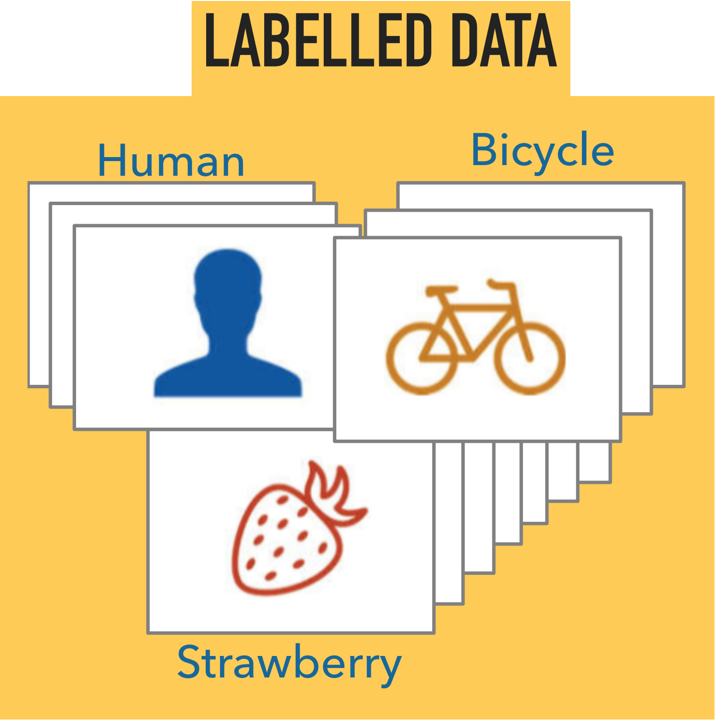
FEED FORWARD NEURAL NETWORKS



FEED FORWARD NEURAL NETWORKS

SUPERVISED TRAINING

Bicycle

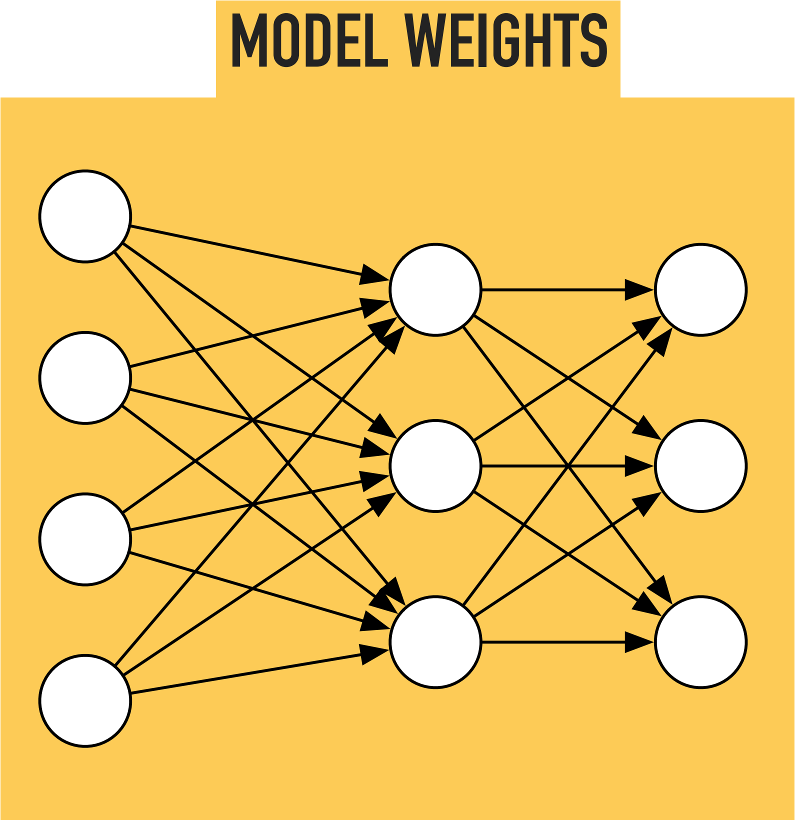


LABELLED DATA

Human

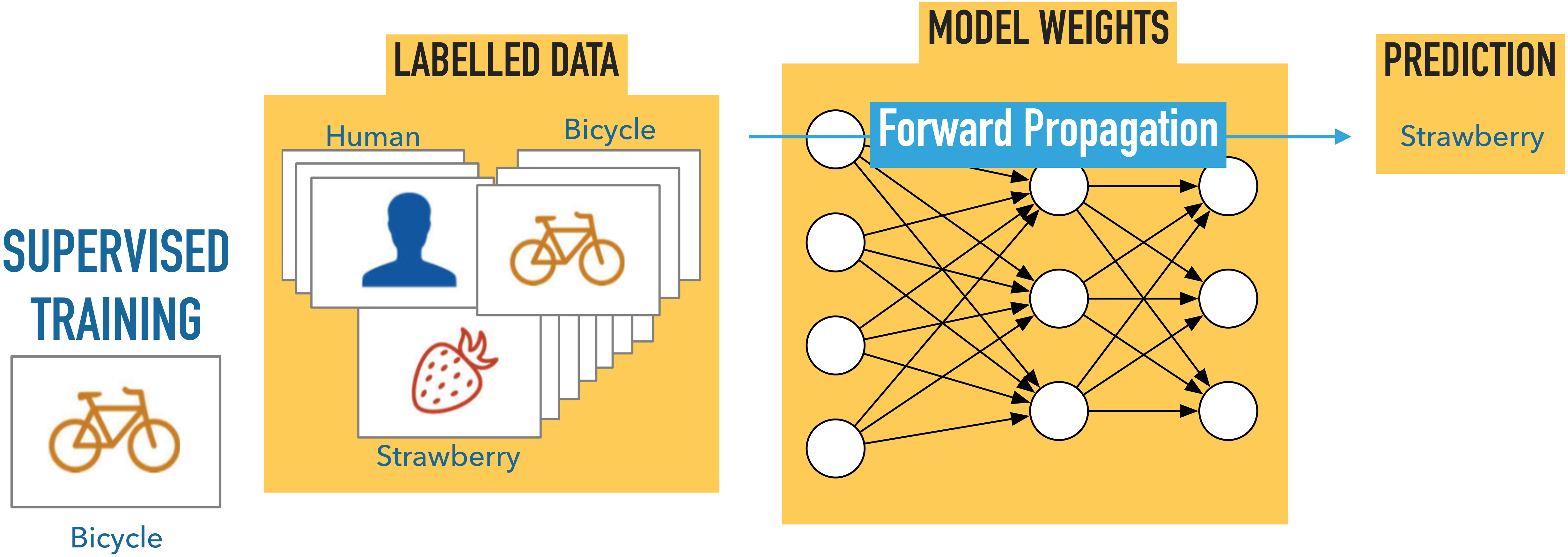
Bicycle

Strawberry



MODEL WEIGHTS

FEED FORWARD NEURAL NETWORKS



SUPERVISED TRAINING

Bicycle

LABELLED DATA

Human

Bicycle

Strawberry

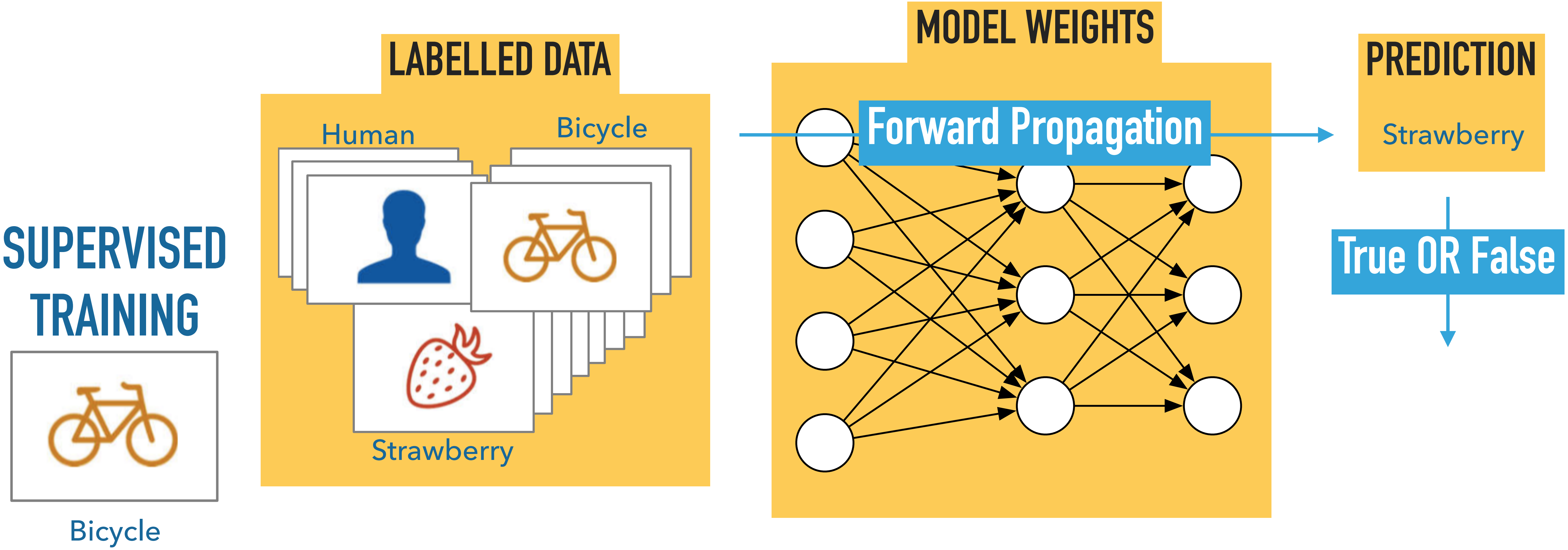
MODEL WEIGHTS

Forward Propagation

PREDICTION

Strawberry

FEED FORWARD NEURAL NETWORKS



SUPERVISED TRAINING

Bicycle

LABELLED DATA

Human

Bicycle

Strawberry

MODEL WEIGHTS

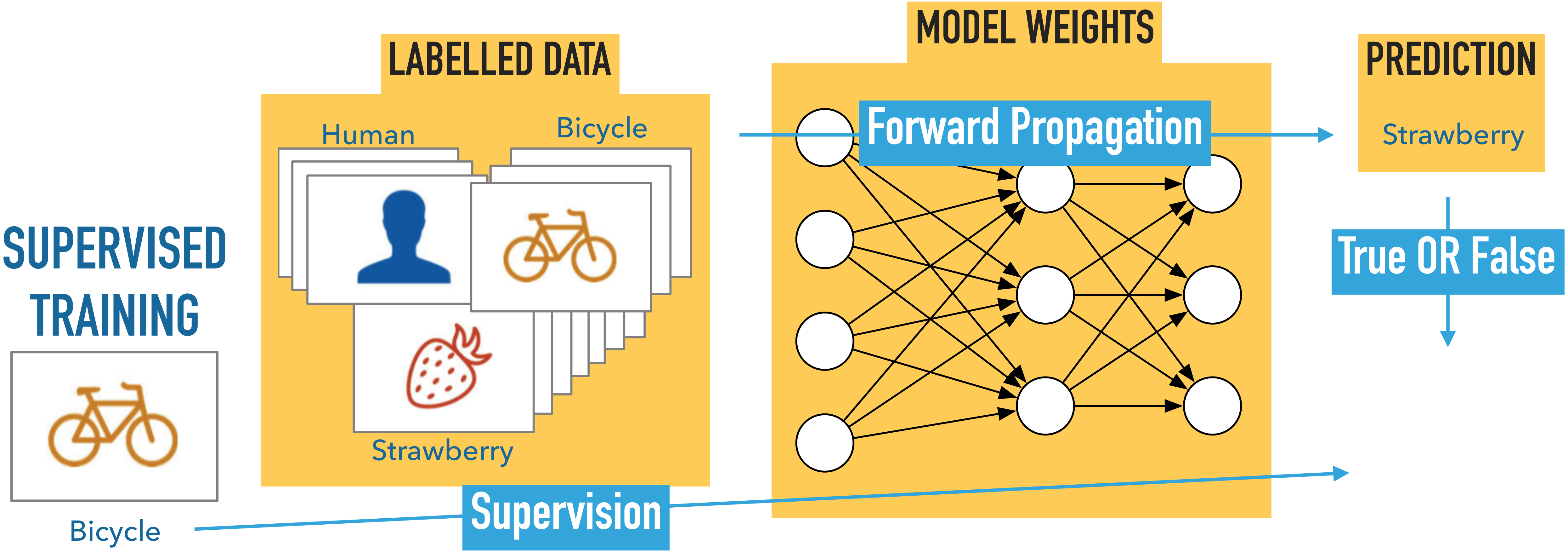
Forward Propagation

PREDICTION

Strawberry

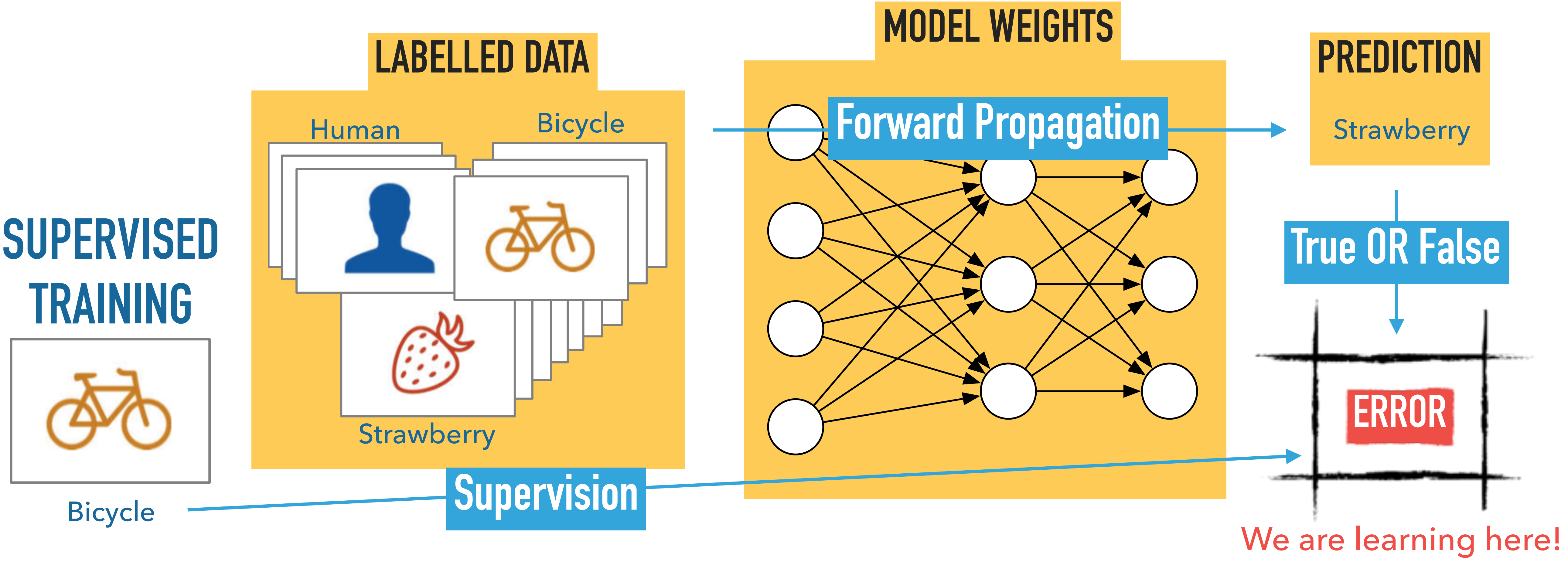
True OR False

FEED FORWARD NEURAL NETWORKS



SUPERVISED TRAINING

FEED FORWARD NEURAL NETWORKS



SUPERVISED TRAINING

Bicycle

LABELLED DATA

Human

Bicycle

Strawberry

Supervision

MODEL WEIGHTS

Forward Propagation

PREDICTION

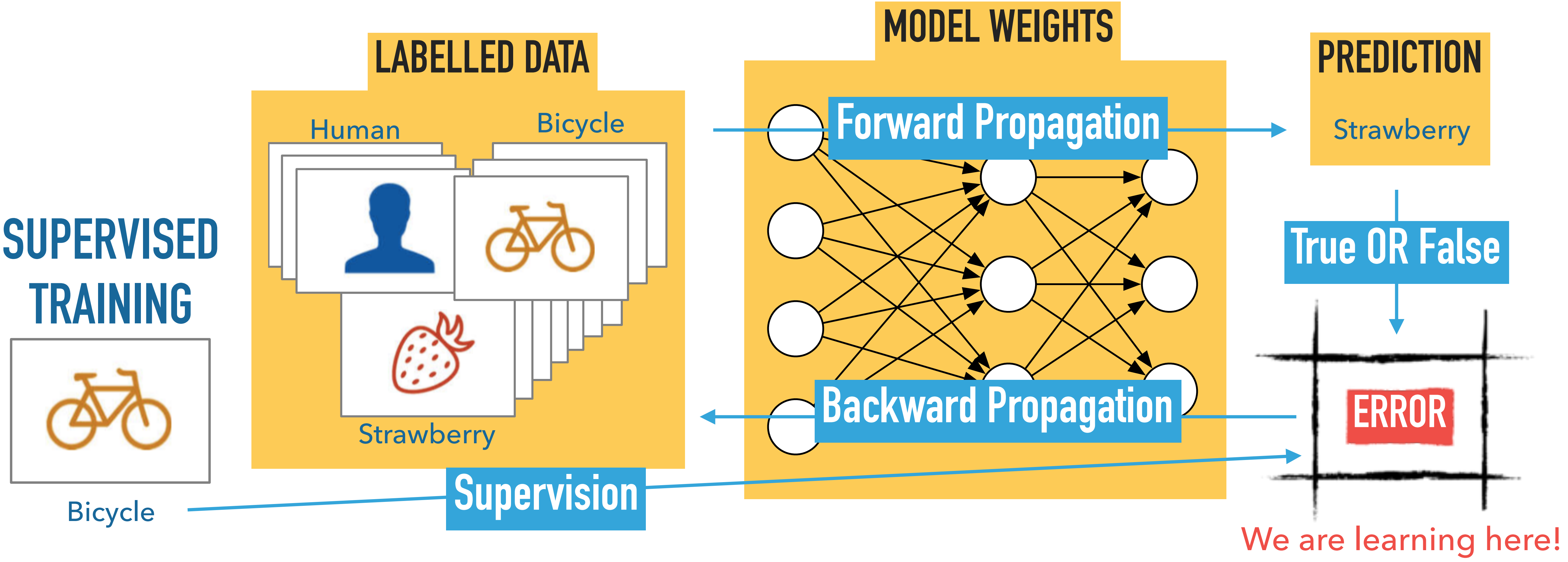
Strawberry

True OR False

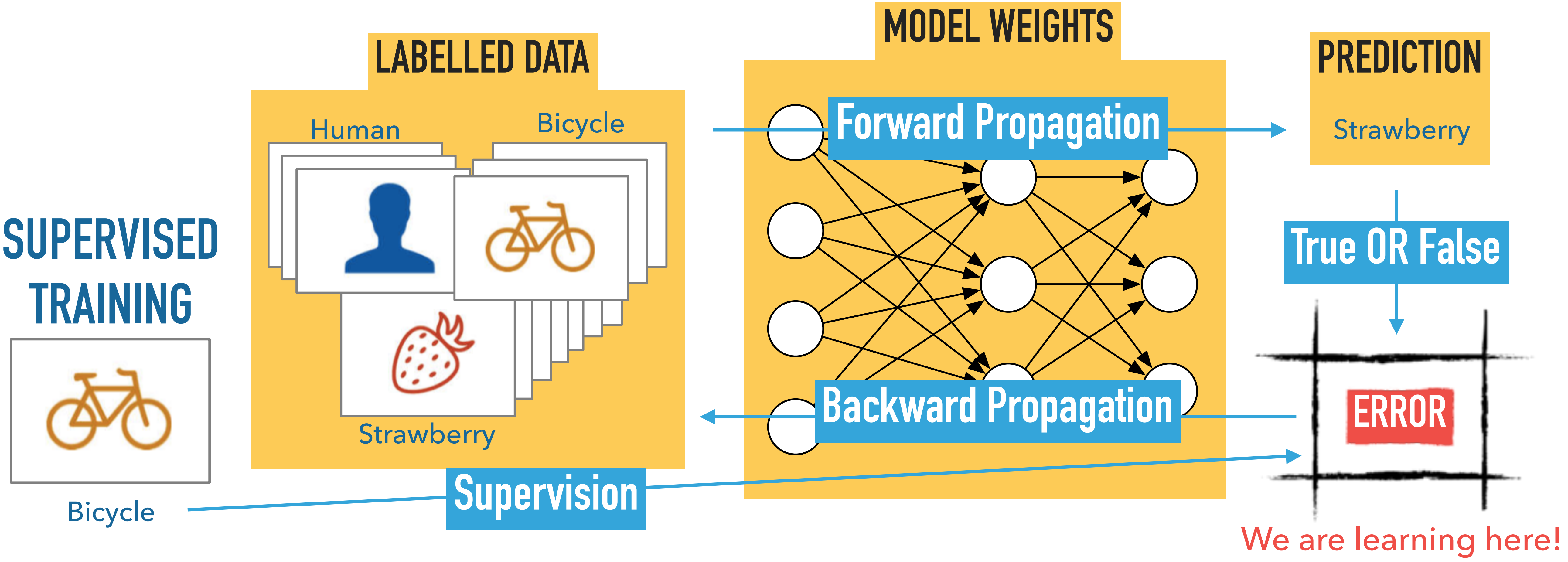
ERROR

We are learning here!

FEED FORWARD NEURAL NETWORKS



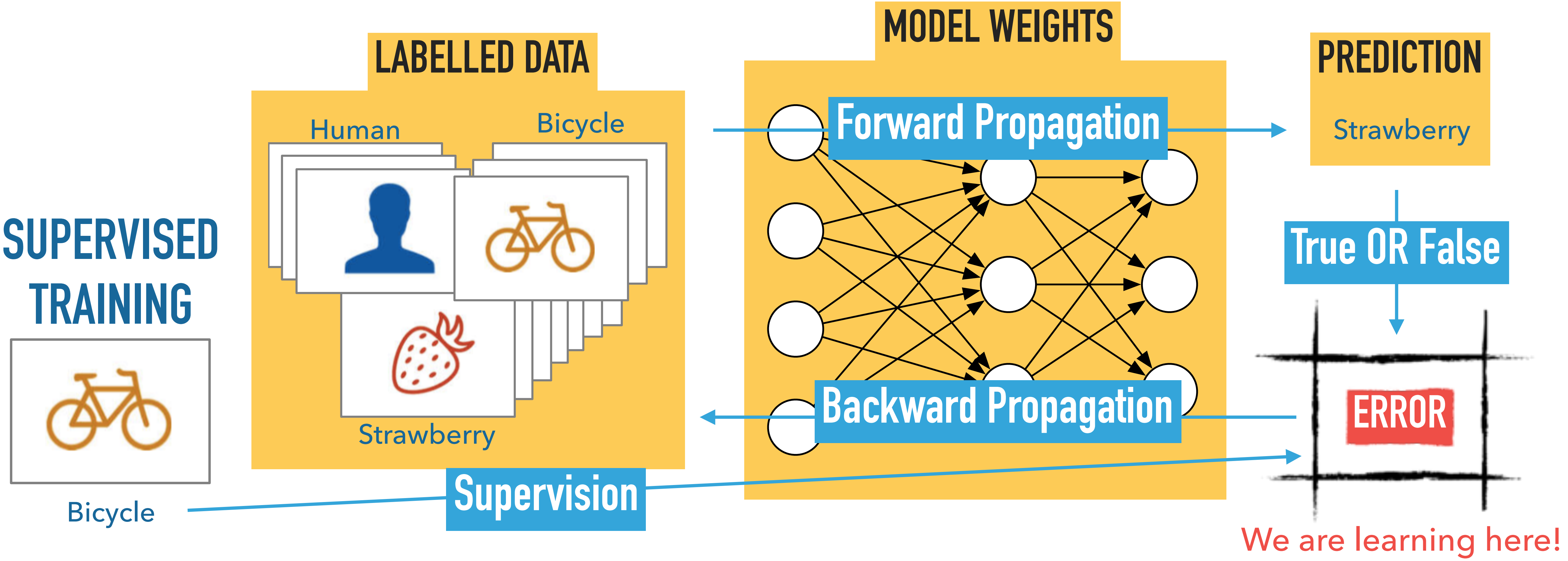
FEED FORWARD NEURAL NETWORKS



BLIND TEST

FEED FORWARD NEURAL NETWORKS

SUPERVISED TRAINING



LABELLED DATA

Human

Bicycle

Strawberry

MODEL WEIGHTS

Forward Propagation

Backward Propagation

PREDICTION

Strawberry

True OR False

ERROR

We are learning here!

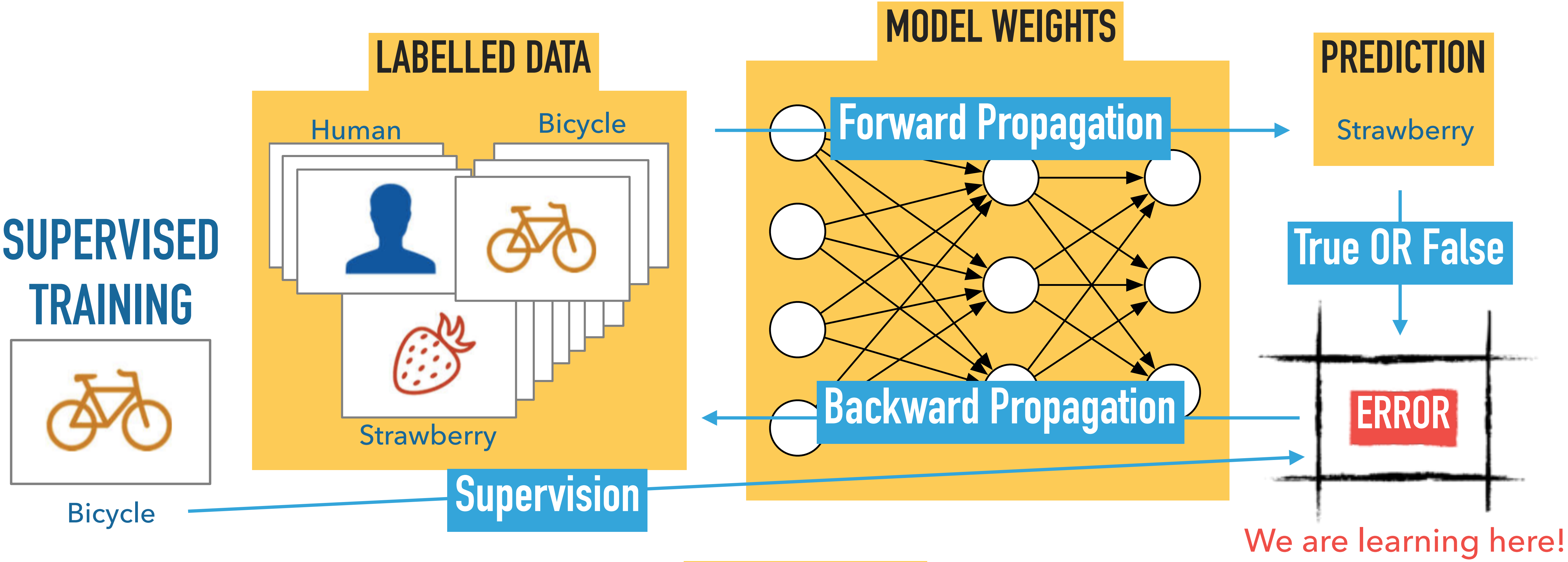
Bicycle

Supervision

BLIND TEST

????

FEED FORWARD NEURAL NETWORKS



SUPERVISED TRAINING

Bicycle

LABELLED DATA

Human

Bicycle

Strawberry

MODEL WEIGHTS

Forward Propagation

Backward Propagation

Supervision

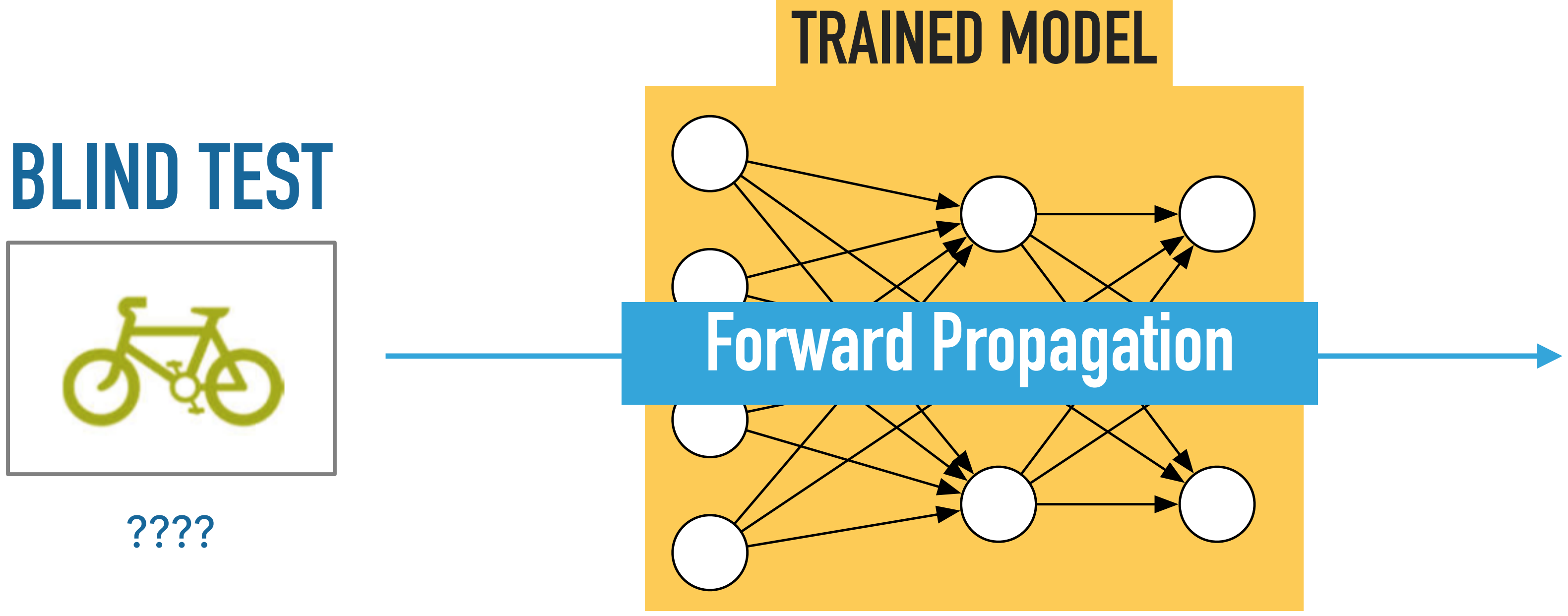
PREDICTION

Strawberry

True OR False

ERROR

We are learning here!



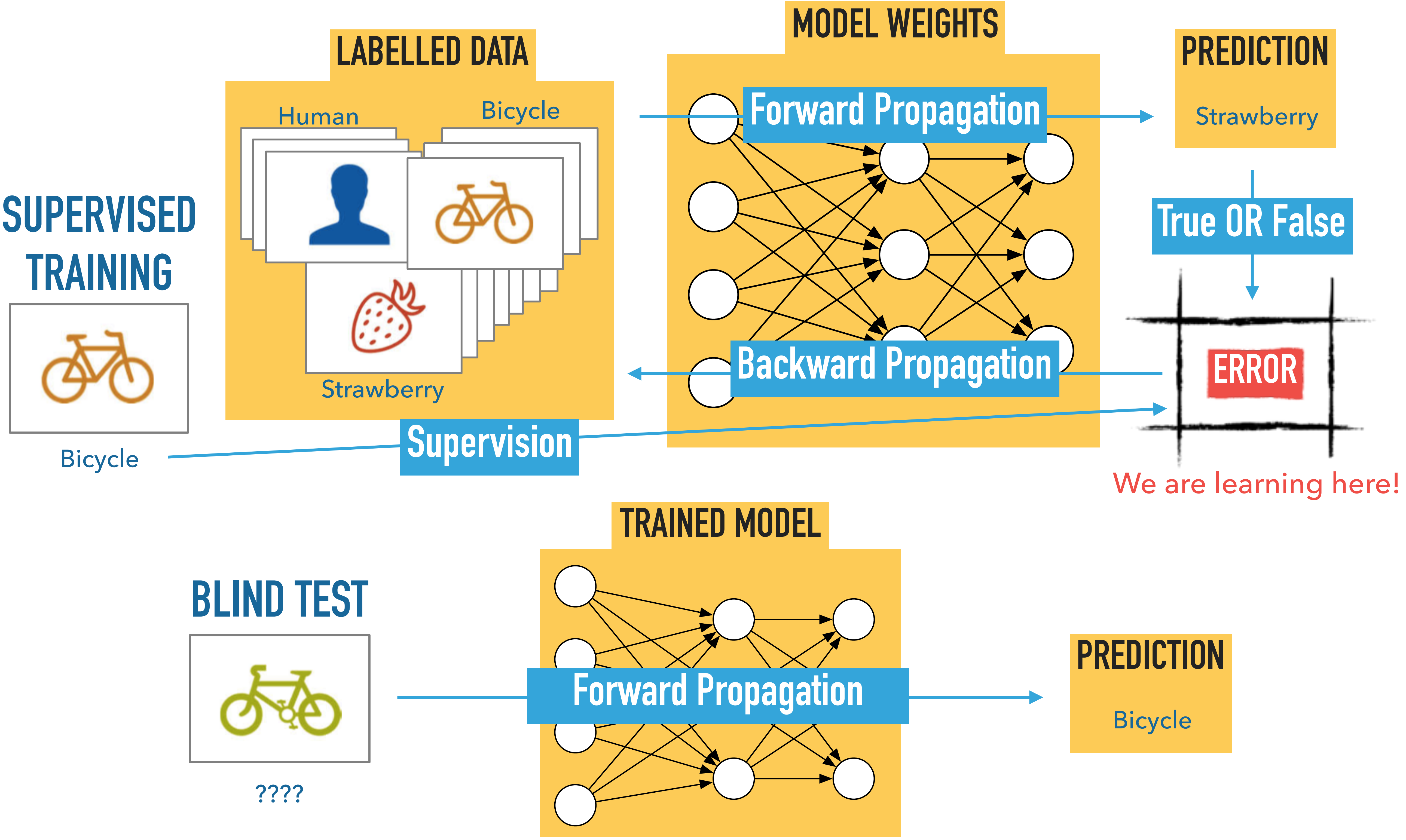
BLIND TEST

????

TRAINED MODEL

Forward Propagation

FEED FORWARD NEURAL NETWORKS



SUPERVISED TRAINING

Bicycle

LABELLED DATA

Human

Bicycle

Strawberry

Supervision

MODEL WEIGHTS

Forward Propagation

Backward Propagation

PREDICTION

Strawberry

True OR False

ERROR

We are learning here!

BLIND TEST

????

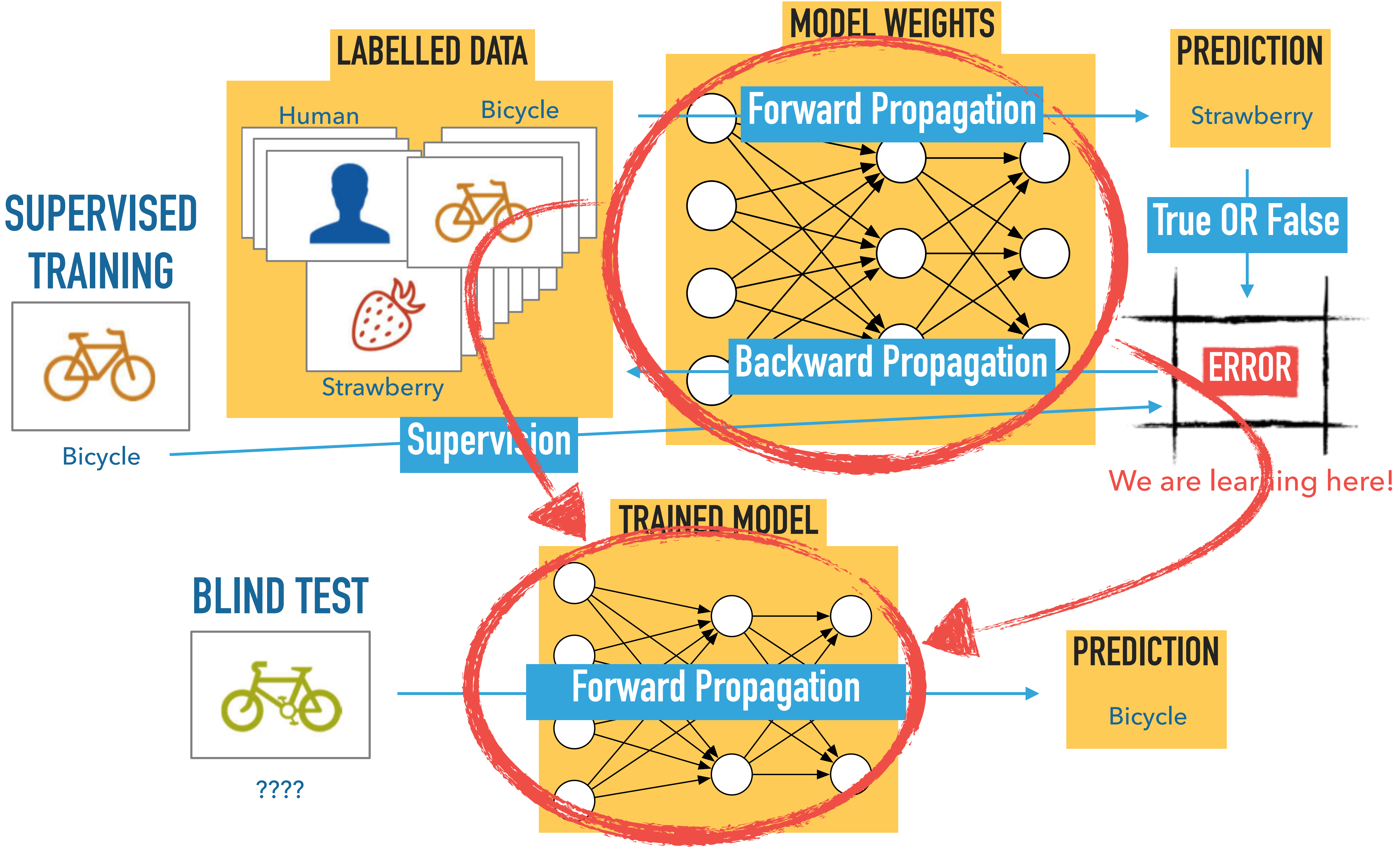
TRAINED MODEL

Forward Propagation

PREDICTION

Bicycle

FEED FORWARD NEURAL NETWORKS



SUPERVISED TRAINING

Bicycle

LABELLED DATA

Human

Bicycle

Strawberry

Supervision

MODEL WEIGHTS

Forward Propagation

Backward Propagation

PREDICTION

Strawberry

True OR False

ERROR

We are learning here!

BLIND TEST

????

TRAINED MODEL

Forward Propagation

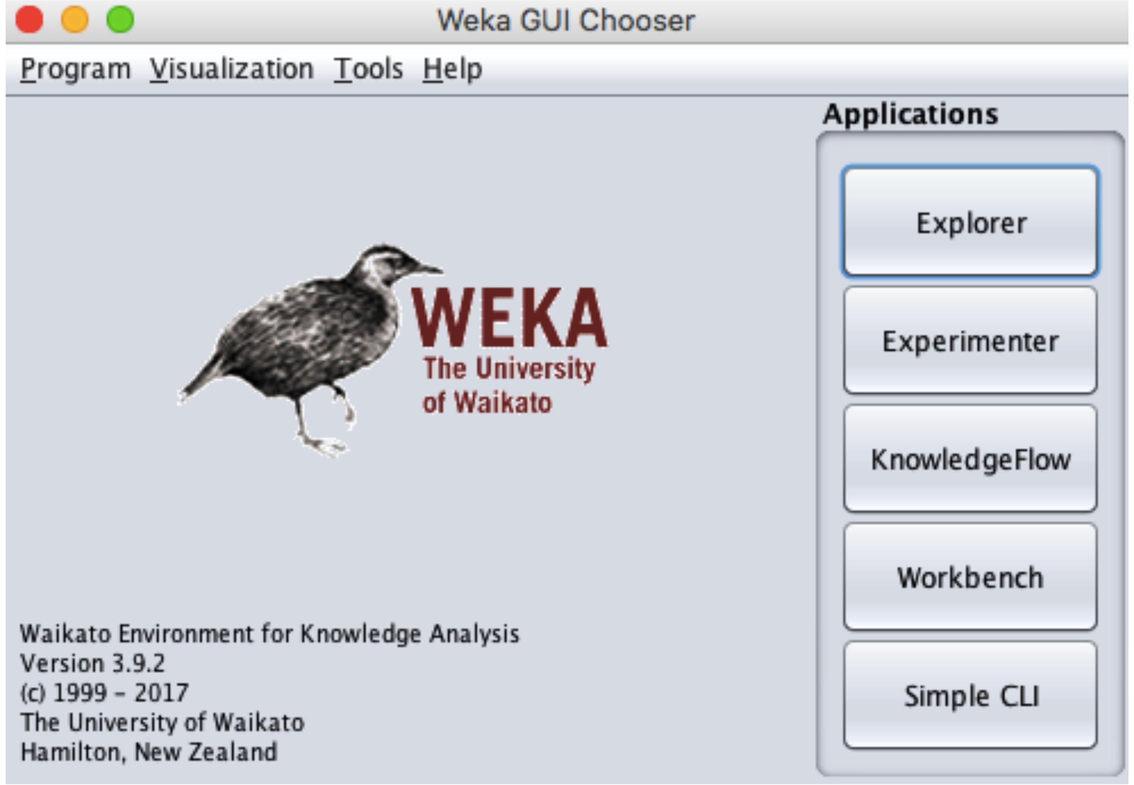
PREDICTION

Bicycle

WEKA — A DATA MINING TOOL

WEKA is developed by the University of Waikato (New Zealand) under the GNU General Public License (GPL).

It is written in the Java™ object-oriented programming language and provides a GUI for interacting with data files and producing visual results.

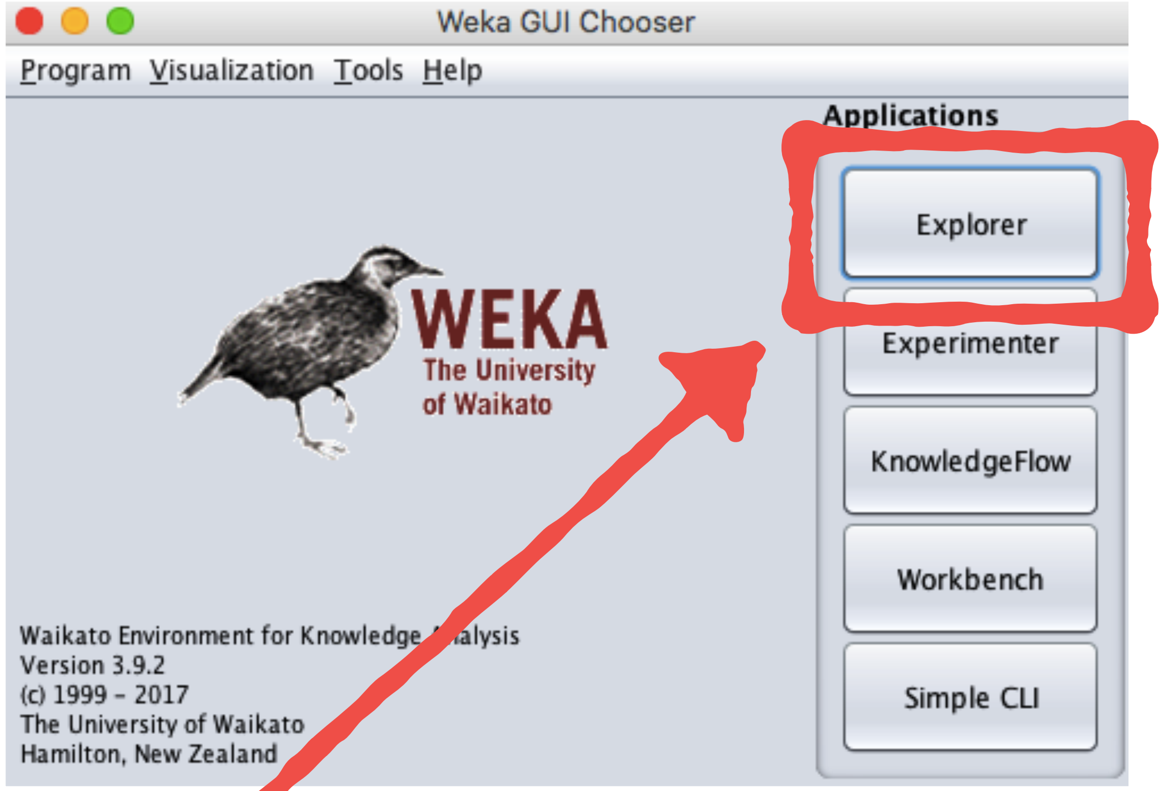


Waikato Environment for Knowledge Analysis
Version 3.9.2
(c) 1999 - 2017
The University of Waikato
Hamilton, New Zealand

WEKA — A DATA MINING TOOL

WEKA is developed by the University of Waikato (New Zealand) under the GNU General Public License (GPL).

It is written in the Java™ object-oriented programming language and provides a GUI for interacting with data files and producing visual results.



Waikato Environment for Knowledge Analysis
Version 3.9.2
(c) 1999 - 2017
The University of Waikato
Hamilton, New Zealand

FOR THE AIM OF THE TUTORIAL WE WILL USE JUST THE "Explorer" TOOL

AGENDA

	Model	Dataset
Classification	**Multivariate Linear Regression**	*House Pricing*
	Decision Tree	*Campaign*
Clustering	***k*-means**	*Behaviour Analysis*
Classification	**Neural Network VS. KNN**	*Breast Cancer*

A SIMPLE EXAMPLE OF PREDICTION

The price of the house (the dependent variable) is the result of many independent variables:

- the square footage of the house;
- the size of the lot;
- whether granite is in the kitchen;
- bathrooms are upgraded;
- etc.

We **create a model** based **on other comparable houses** in the neighbourhood and what they sold for, then put the values of our own house into this model **to produce an expected price**.

COLLECTING THE DATA

House size (square feet)	Lot size	Bedrooms	Granite	Upgraded bathroom?	Selling price
3529	9191	6	0	0	205000
3247	10061	5	1	1	224900
4032	10150	5	0	1	197900
2397	14156	4	1	0	189900
2200	9600	4	0	1	195000
3536	19994	6	1	1	325000
2983	9365	5	0	1	230000
3198	9669	5	1	1	????

COLLECTING THE DATA

House size (square feet)	Lot size	Bedrooms	Granite	Upgraded bathroom?	Selling price
3529	9191	6	0	0	205000
3247	10061	5	1	1	224900
4032	10150	5	0	1	197900
2397	14156	4	1	0	189900
2200	9600	4	0	1	195000
3536	19994	6	1	1	325000
2983	9365	5	0	1	230000
	9669	5	1	1	????

TRAINING DATA

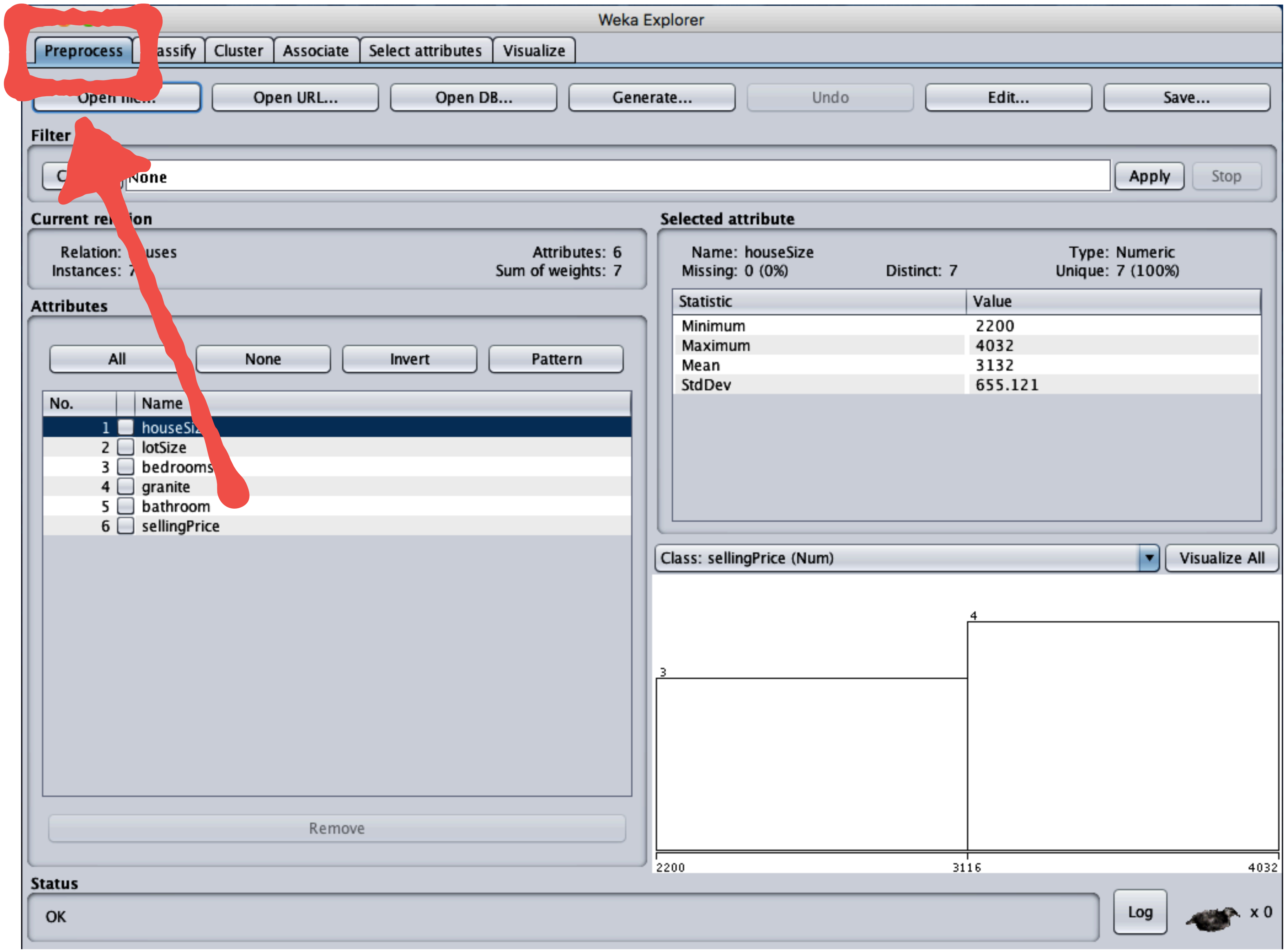
COLLECTING THE DATA

House size (square feet)	Lot size	Bedrooms	Granite	Upgraded bathroom?	Selling price
3529	9191	6	0	0	205000
3247	10061	5	1	1	224900
4032	10150	5	0	1	197900
2397	14156	4	1	0	189900
2200	9600	4	0	1	195000
3536	19994	6	1	1	325000
2983	9365	5	0	1	230000
3198	9669	5	1	1	????

TESTING DATA

PREPROCESS THE DATA WITH WEKA
LOAD THE DATASET

- Preprocess Tab



Filter

None

Current relation

Relation: uses
Instances: 7
Attributes: 6
Sum of weights: 7

Attributes

No.		Name
1	□	houseSiz
2	□	lotSize
3	□	bedrooms
4	□	granite
5	□	bathroom
6	□	sellingPrice

Selected attribute

Name: houseSize
Type: Numeric
Missing: 0 (0%)
Distinct: 7
Unique: 7 (100%)

Statistic	Value
Minimum	2200
Maximum	4032
Mean	3132
StdDev	655.121

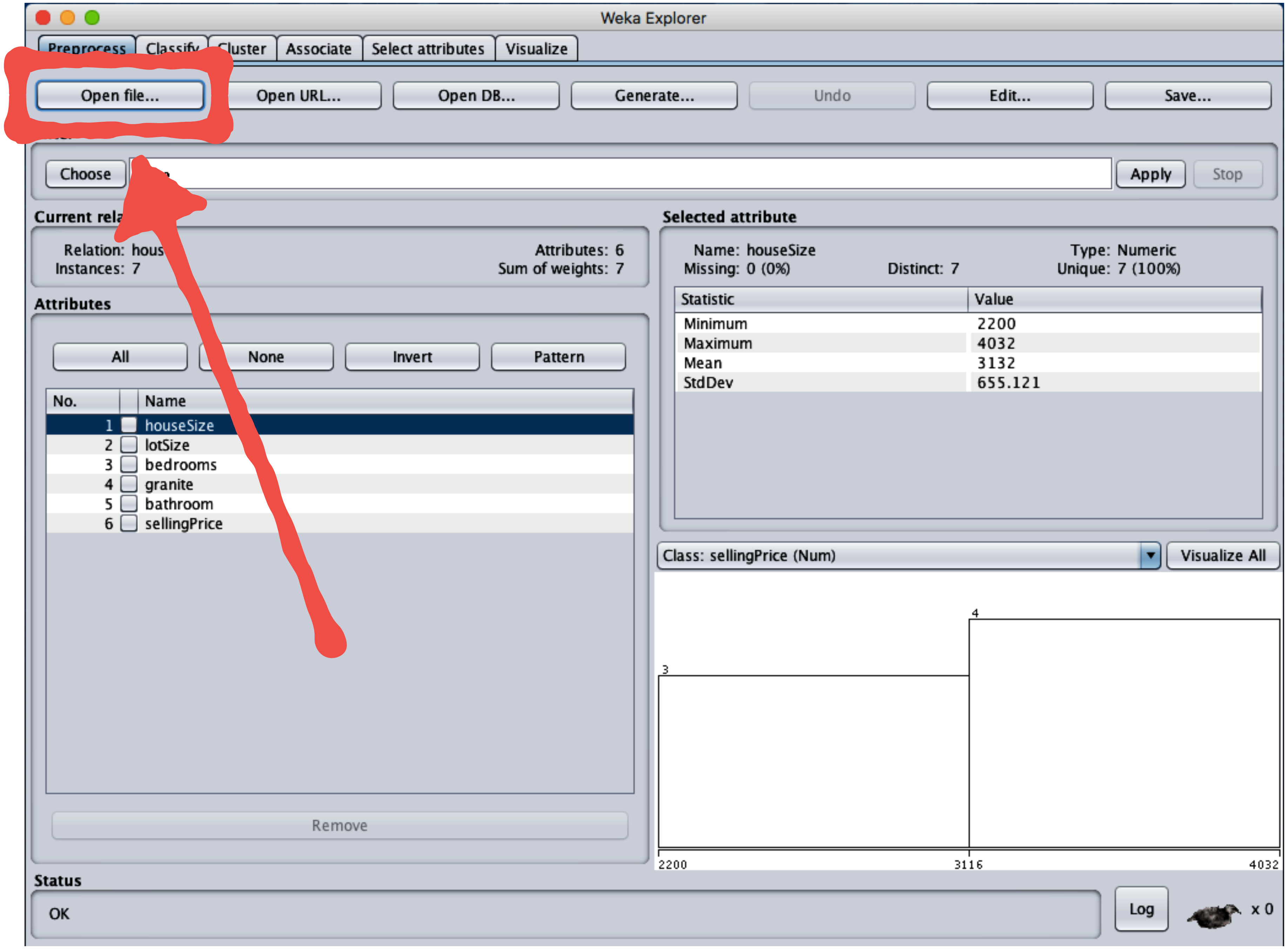
Class: sellingPrice (Num)

Status

OK

PREPROCESS THE DATA WITH WEKA
LOAD THE DATASET

- Preprocess Tab
- Open File:
  - CSV format
  - XLS format
  - ARFF format



Current rela

Relation: hous
Instances: 7
Attributes: 6
Sum of weights: 7

Attributes

No.	Name
1	houseSize
2	lotSize
3	bedrooms
4	granite
5	bathroom
6	sellingPrice

Selected attribute

Name: houseSize
Type: Numeric
Missing: 0 (0%)
Distinct: 7
Unique: 7 (100%)

Statistic	Value
Minimum	2200
Maximum	4032
Mean	3132
StdDev	655.121

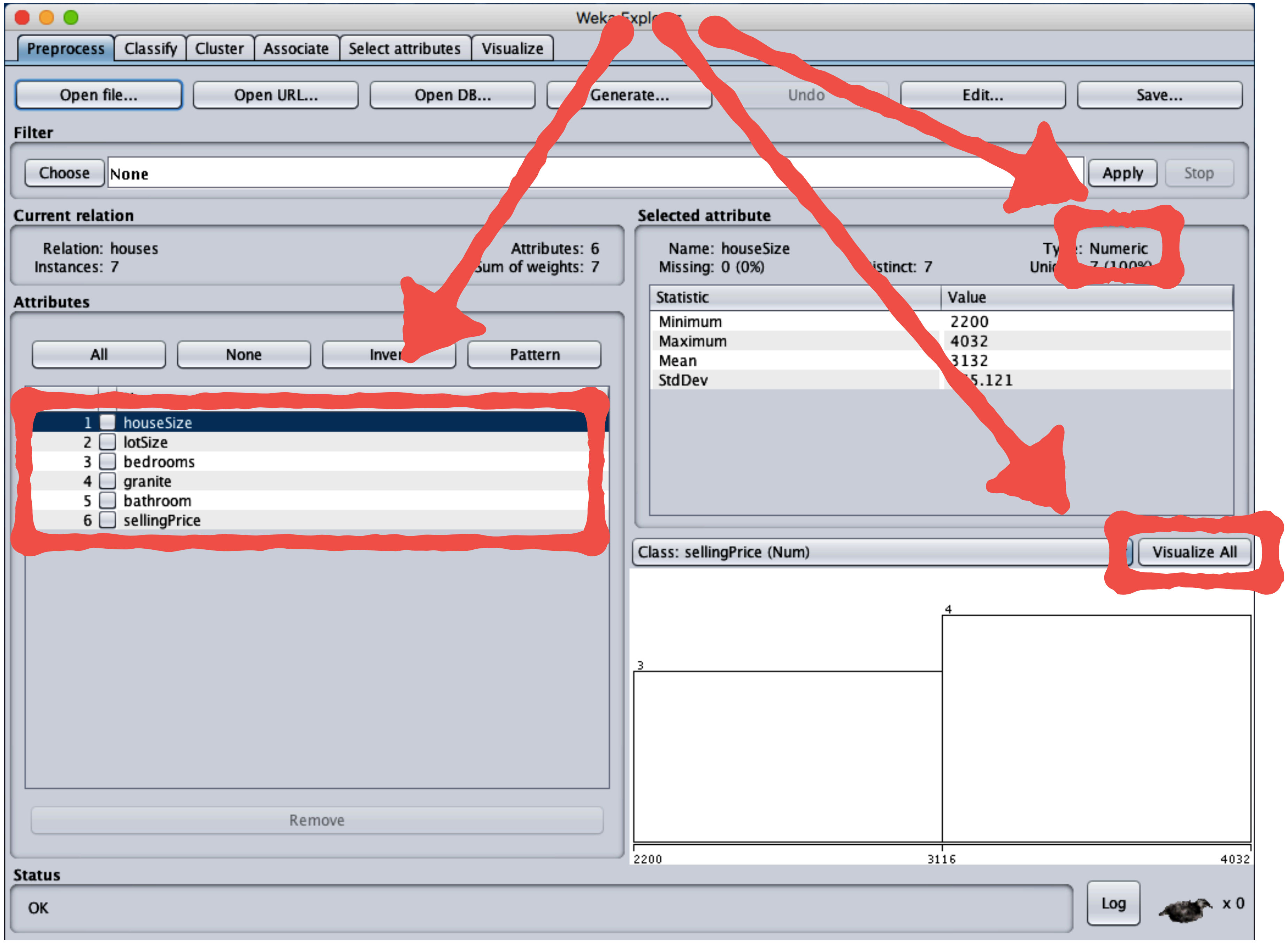
Class: sellingPrice (Num)

Status

OK

PREPROCESS THE DATA WITH WEKA
LOAD THE DATASET

- **Numerical *variables*** data with value representable with numbers
- **Visualize All** shows all graphics at once



Filter: None

Relation: houses
Instances: 7
Attributes: 6
Sum of weights: 7

1	□	houseSize
2	□	lotSize
3	□	bedrooms
4	□	granite
5	□	bathroom
6	□	sellingPrice

Name: houseSize
Missing: 0 (0%)
Distinct: 7
Type: Numeric
Unique: 7 (100%)

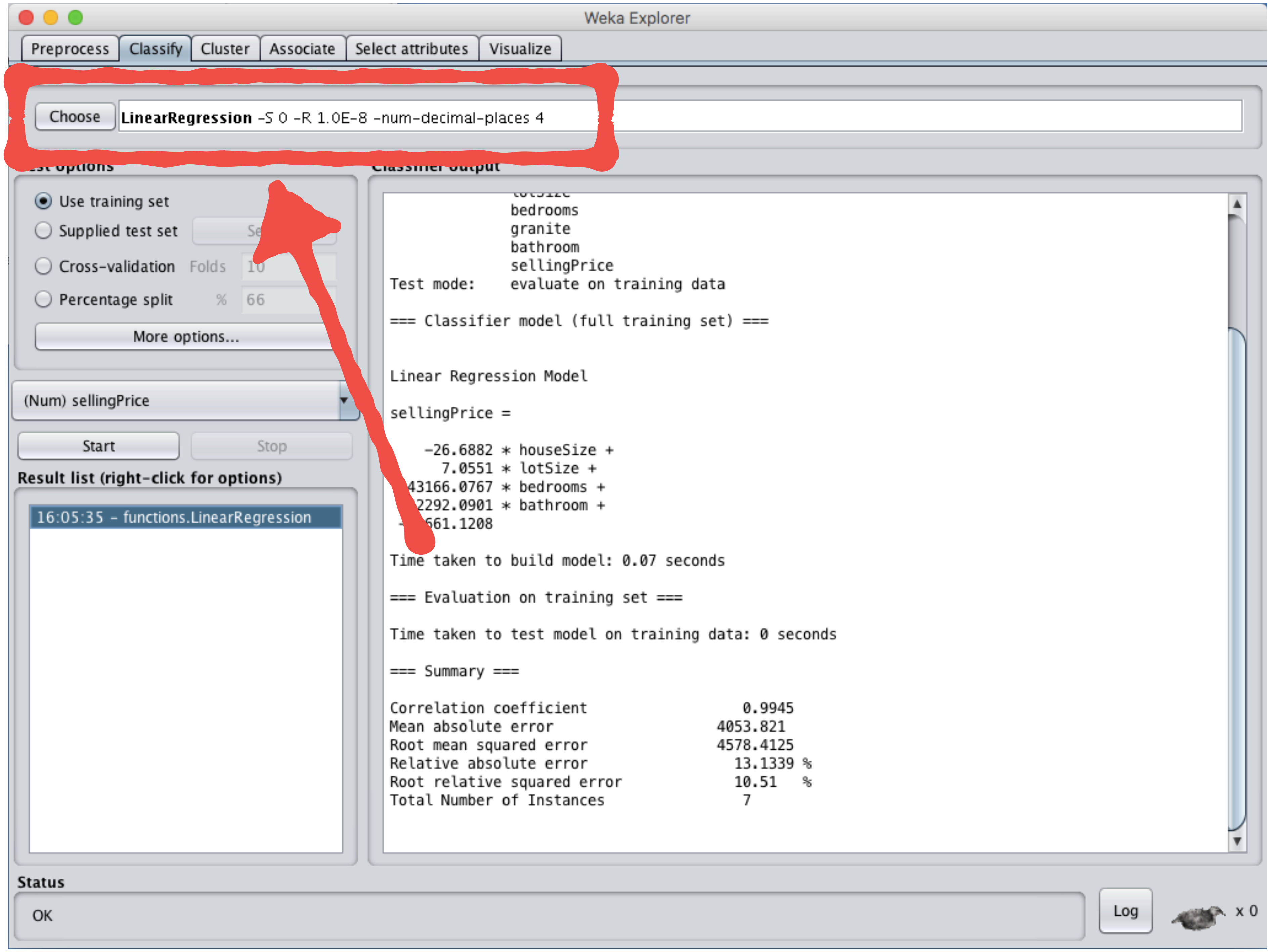
Statistic	Value
Minimum	2200
Maximum	4032
Mean	3132
StdDev	[illegible]5.121

Class: sellingPrice (Num)

OK

CLASSIFY THE DATA WITH WEKA
LINEAR REGRESSION MODEL

- Classify Tab



LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Num) sellingPrice

Result list (right-click for options)

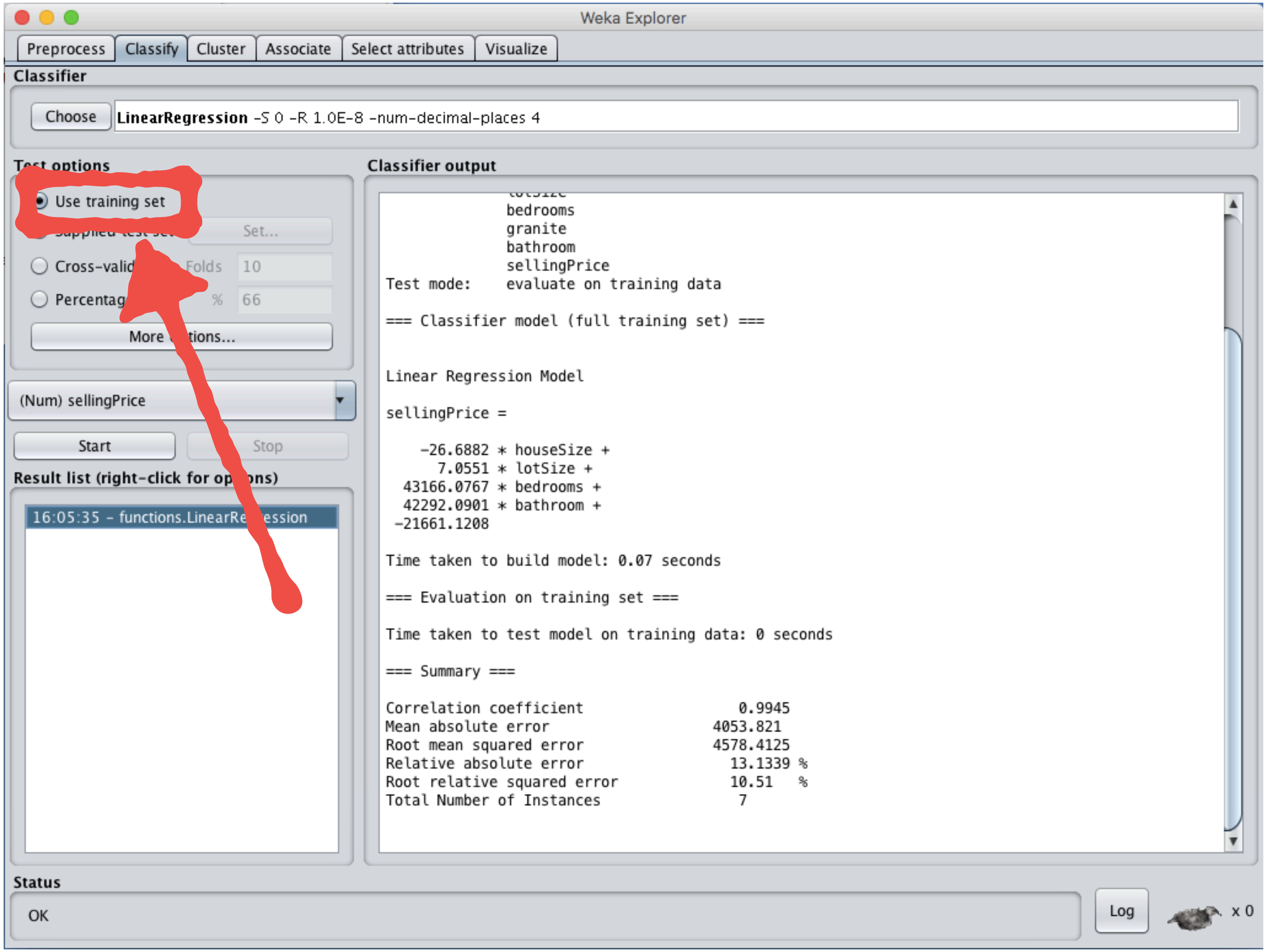
16:05:35 - functions.LinearRegression

```
              lotSize
              bedrooms
              granite
              bathroom
              sellingPrice
Test mode:    evaluate on training data

=== Classifier model (full training set) ===


Linear Regression Model

sellingPrice =

    -26.6882 * houseSize +
      7.0551 * lotSize +
  43166.0767 * bedrooms +
   2292.0901 * bathroom +
 -   661.1208

Time taken to build model: 0.07 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correlation coefficient                  0.9945
Mean absolute error                   4053.821
Root mean squared error               4578.4125
Relative absolute error                 13.1339 %
Root relative squared error             10.51   %
Total Number of Instances                7
```

Status
OK

CLASSIFY THE DATA WITH WEKA
LINEAR REGRESSION MODEL

- Classify Tab
- Use training set



Classifier: LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

Folds: 10

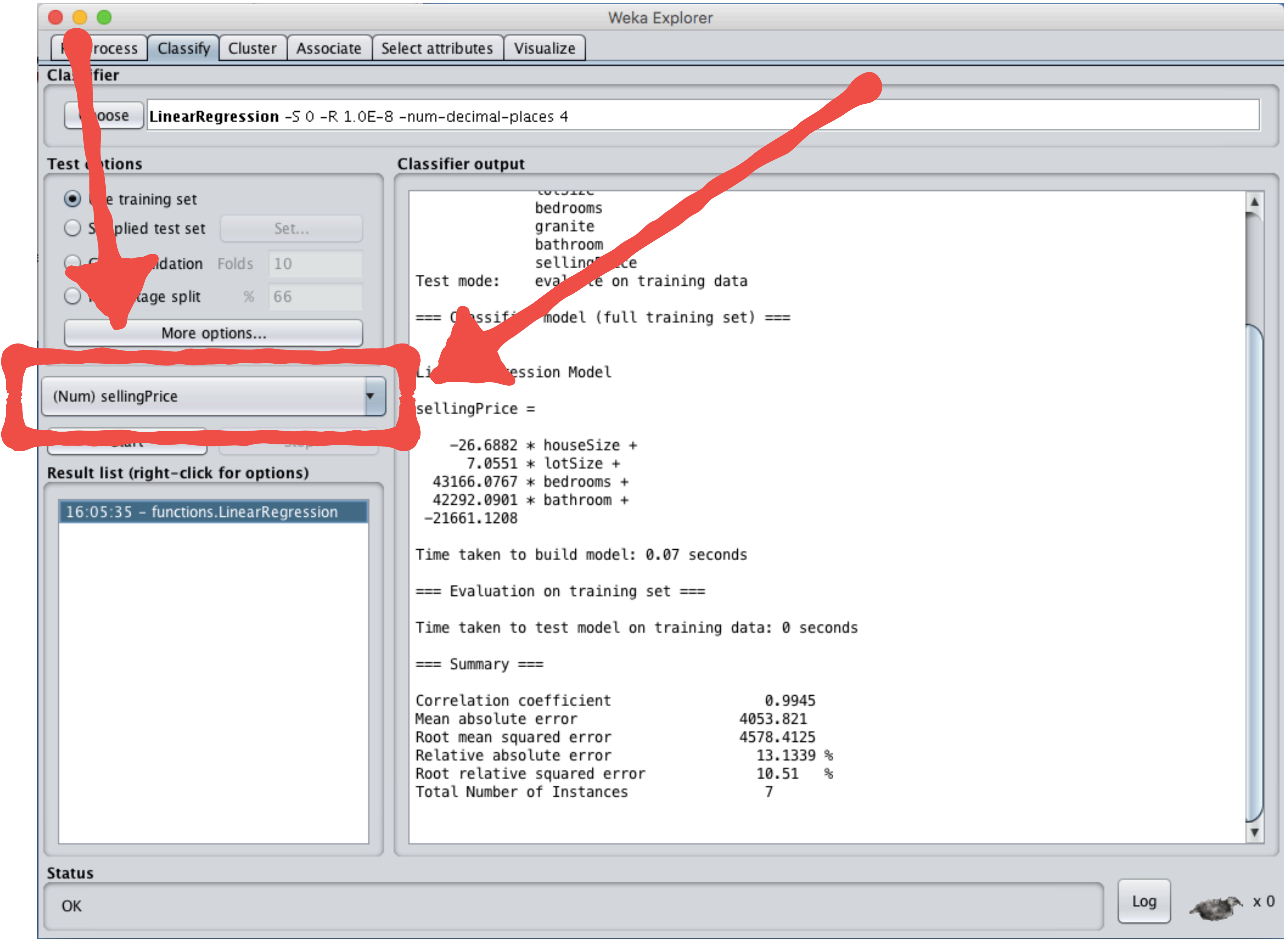
%: 66

(Num) sellingPrice

```
              lotSize
              bedrooms
              granite
              bathroom
              sellingPrice
Test mode:    evaluate on training data

=== Classifier model (full training set) ===


Linear Regression Model

sellingPrice =

    -26.6882 * houseSize +
      7.0551 * lotSize +
  43166.0767 * bedrooms +
  42292.0901 * bathroom +
 -21661.1208

Time taken to build model: 0.07 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correlation coefficient                  0.9945
Mean absolute error                   4053.821
Root mean squared error               4578.4125
Relative absolute error                 13.1339 %
Root relative squared error             10.51   %
Total Number of Instances                7
```

CLASSIFY THE DATA WITH WEKA
LINEAR REGRESSION MODEL

- Classify Tab;
- Use training set;
- Class = sellingPrice;



Cla ifier

oose LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

Test tions

● e training set

○ S plied test set Set...

○ C idation Folds 10

○ tage split % 66

(Num) sellingPrice

Result list (right-click for options)

16:05:35 - functions.LinearRegression

Classifier output

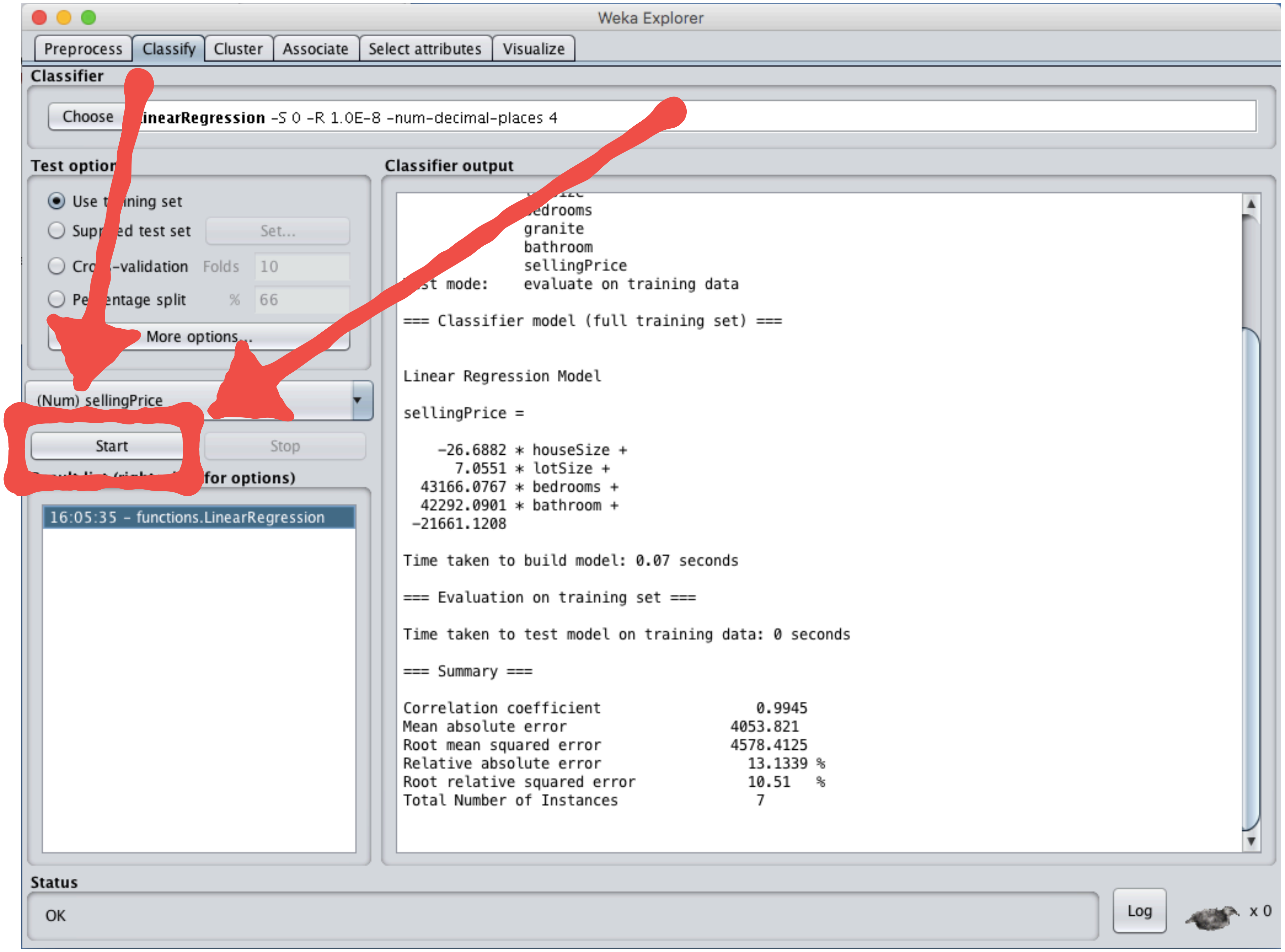
```
                 bedrooms
                 granite
                 bathroom
                 selling  ce
Test mode:       eval   e on training data

=== C ssifi    model (full training set) ===


Li         ession Model

sellingPrice =

    -26.6882 * houseSize +
      7.0551 * lotSize +
  43166.0767 * bedrooms +
  42292.0901 * bathroom +
 -21661.1208

Time taken to build model: 0.07 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correlation coefficient                  0.9945
Mean absolute error                   4053.821
Root mean squared error               4578.4125
Relative absolute error                 13.1339 %
Root relative squared error             10.51   %
Total Number of Instances                7
```

Status

OK

CLASSIFY THE DATA WITH WEKA
LINEAR REGRESSION MODEL

- Classify Tab;
- Use training set;
- Class = sellingPrice;
- Start building the model



Classifier: inearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

○ Use t ining set (●)
Folds 10
% 66

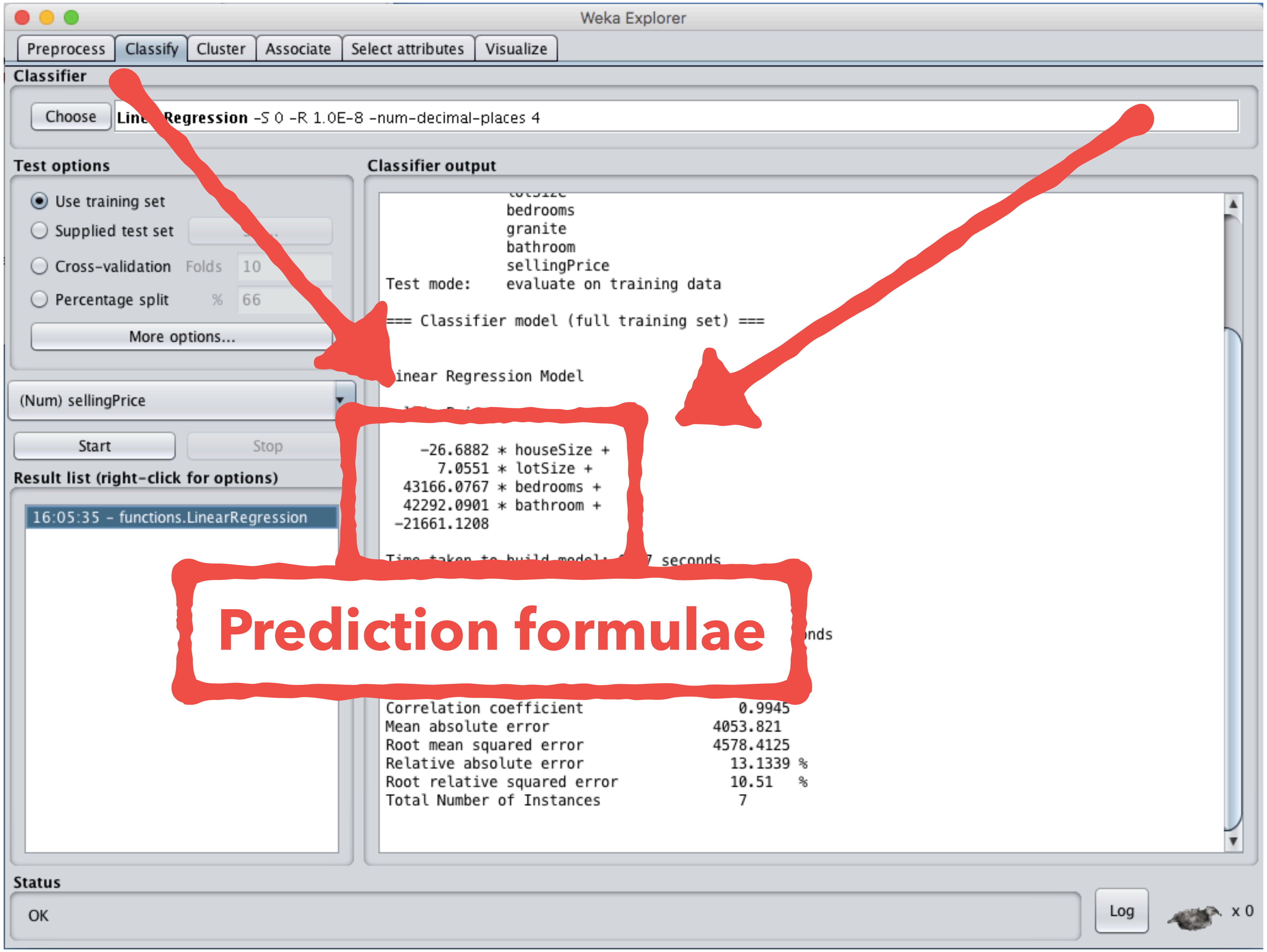
(Num) sellingPrice

Status: OK

```
            edrooms
                 granite
                 bathroom
                 sellingPrice
 st mode:        evaluate on training data

=== Classifier model (full training set) ===


Linear Regression Model

sellingPrice =

    -26.6882 * houseSize +
      7.0551 * lotSize +
  43166.0767 * bedrooms +
  42292.0901 * bathroom +
 -21661.1208

Time taken to build model: 0.07 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correlation coefficient                  0.9945
Mean absolute error                   4053.821
Root mean squared error               4578.4125
Relative absolute error                 13.1339 %
Root relative squared error             10.51   %
Total Number of Instances                7
```

CLASSIFY THE DATA WITH WEKA
LINEAR REGRESSION MODEL

- Classify Tab;
- Use training set;
- Class = sellingPrice;
- Start building the model



LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Num) sellingPrice

16:05:35 - functions.LinearRegression

```
              lotSize
              bedrooms
              granite
              bathroom
              sellingPrice
Test mode:    evaluate on training data

=== Classifier model (full training set) ===


Linear Regression Model


    -26.6882 * houseSize +
      7.0551 * lotSize +
  43166.0767 * bedrooms +
  42292.0901 * bathroom +
 -21661.1208

Time taken to build model: 0.[illegible]7 seconds

                              nds

Correlation coefficient                  0.9945
Mean absolute error                   4053.821
Root mean squared error               4578.4125
Relative absolute error                 13.1339 %
Root relative squared error             10.51   %
Total Number of Instances                7
```

OK

Prediction formulae

FINAL PREDICTION

sellingPrice =

- 26,68 * **[houseSize = 3198]**

+ 7,05 * **[lotSize = 9669]**

+ 43.166,07 * **[bedrooms = 5]**

+ 42.292,09 * **[bathroom = 1]**

- 21.661,12 = **219.328,25**

ANOTHER EXAMPLE OF CLASSIFICATION
CAR DEALERSHIP

The dealership is **starting a promotional campaign**, whereby it is **trying to push a two-year extended warranty** to its past customers.

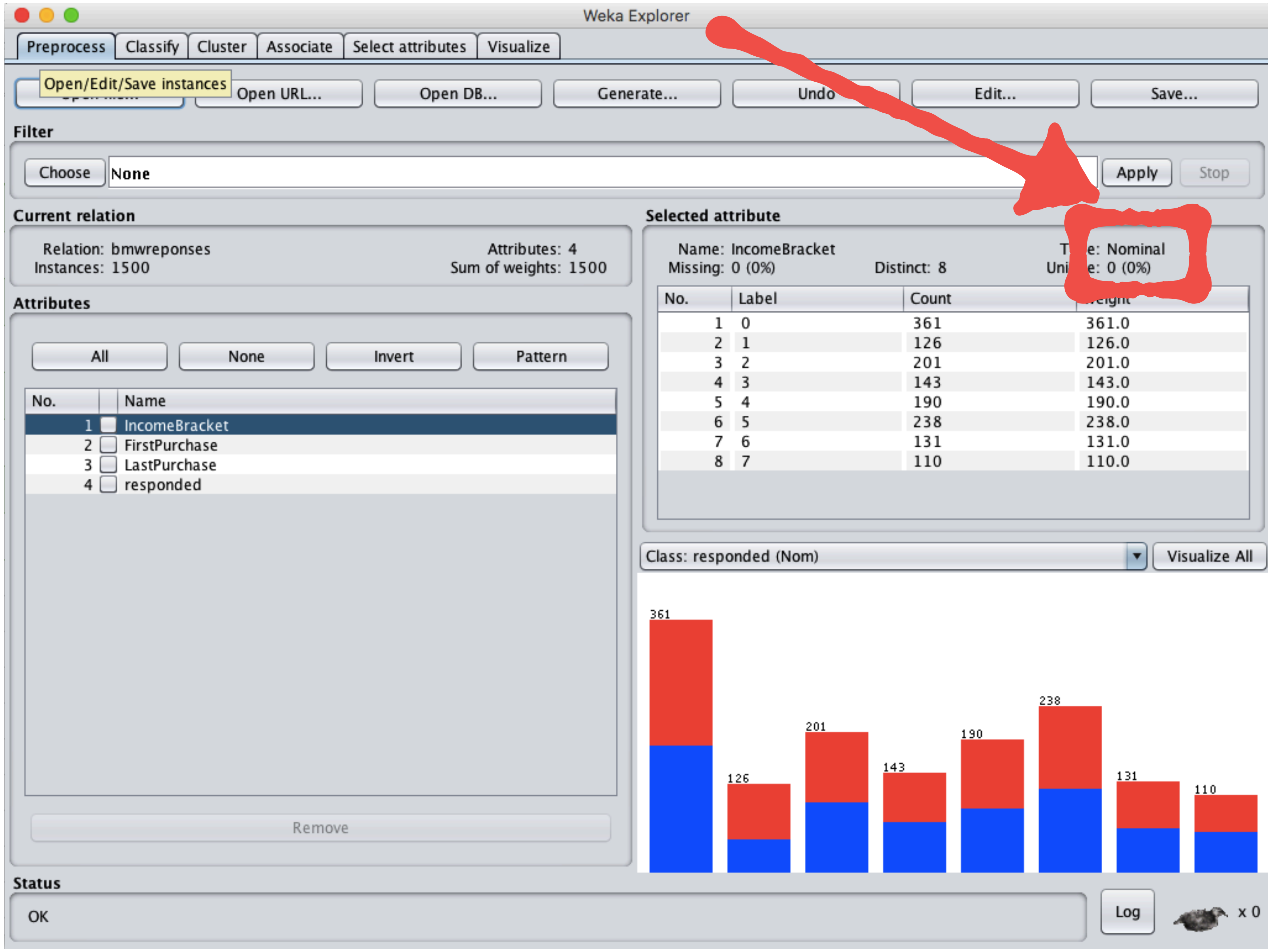
The dealership **has done this before** and has gathered **4,500 data points from past** sales of extended warranties.

The attributes in the data set are:

- Income bracket [0=$0-$30k, 1=$31k-$40k, 2=$41k-$60k, 3=$61k-$75k, 4=$76k-$100k, 5=$101k-$150k, 6=$151k-$500k, 7=$501k+]
- Year/month first car bought
- Year/month most recent car bought
- **Whether they responded or not** to the extended warranty offer in the past

PREPROCESS THE DATA WITH WEKA
LOAD THE DATASET FOR TRAINING

- **Nominal** ***variables*** – *labelled data*



Filter: None

Current relation

Relation: bmwreponses
Instances: 1500
Attributes: 4
Sum of weights: 1500

Attributes

No.		Name
1	□	IncomeBracket
2	□	FirstPurchase
3	□	LastPurchase
4	□	responded

Selected attribute

Name: IncomeBracket
Missing: 0 (0%)
Distinct: 8
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	0	361	361.0
2	1	126	126.0
3	2	201	201.0
4	3	143	143.0
5	4	190	190.0
6	5	238	238.0
7	6	131	131.0
8	7	110	110.0

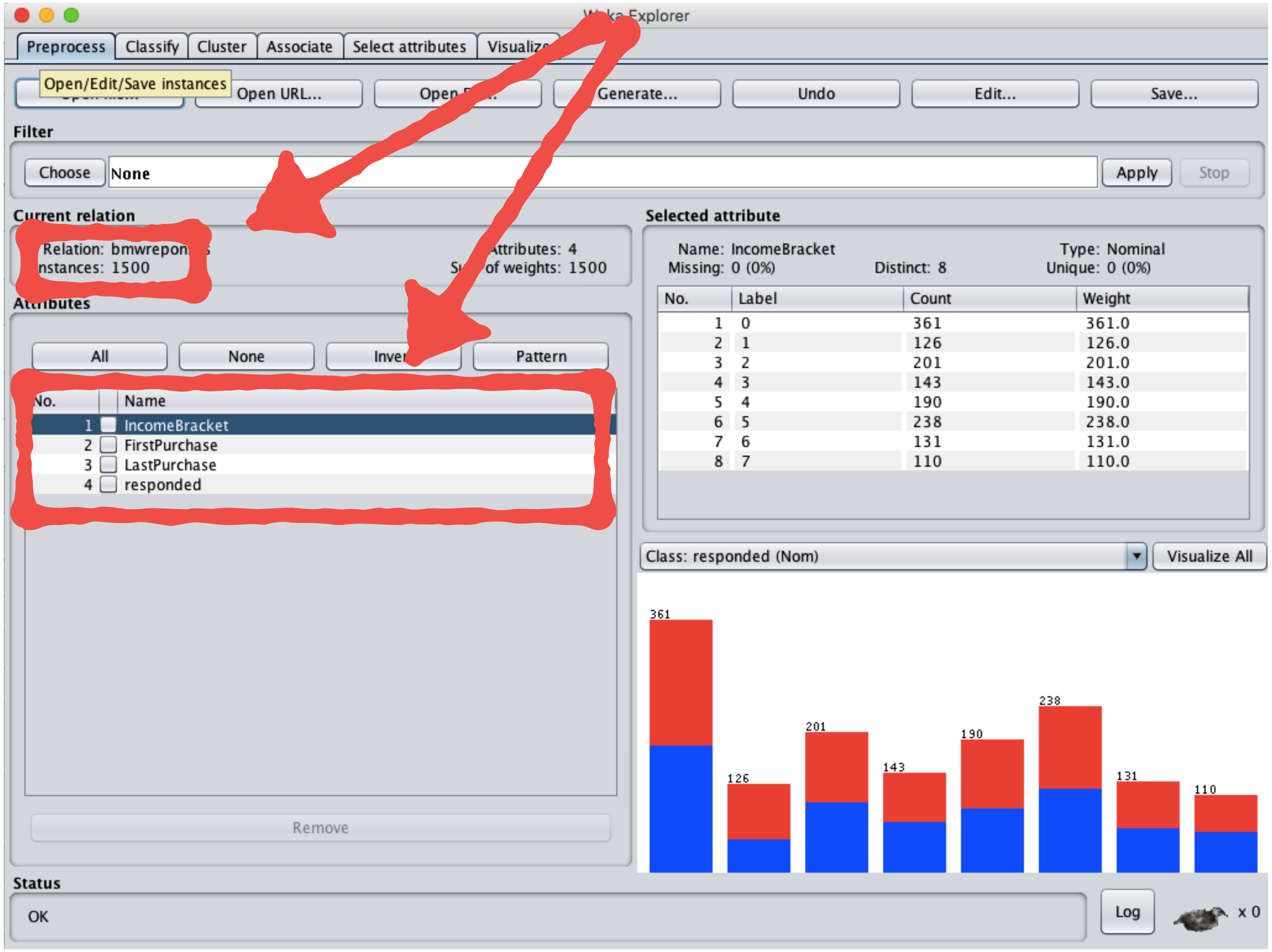
Class: responded (Nom)

Status

OK

PREPROCESS THE DATA WITH WEKA
LOAD THE DATASET FOR TRAINING

- **Nominal *variables*** – *labelled data*
- **1500 instances**



Filter: None

Relation: bmwrepon...s
...nstances: 1500
Attributes: 4
Su... of weights: 1500

No.	Name
1	IncomeBracket
2	FirstPurchase
3	LastPurchase
4	responded

Name: IncomeBracket
Missing: 0 (0%)
Distinct: 8
Type: Nominal
Unique: 0 (0%)

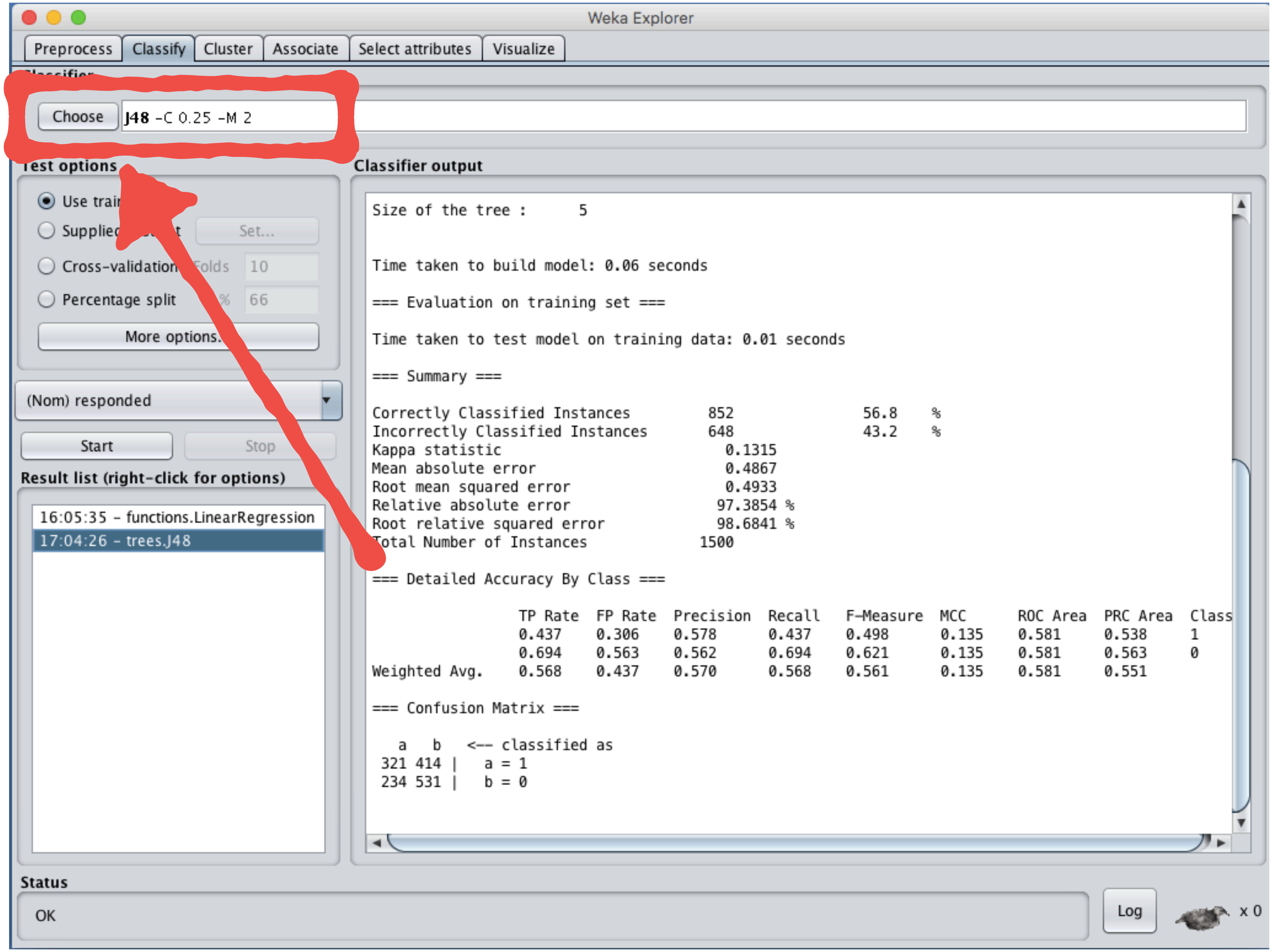
No.	Label	Count	Weight
1	0	361	361.0
2	1	126	126.0
3	2	201	201.0
4	3	143	143.0
5	4	190	190.0
6	5	238	238.0
7	6	131	131.0
8	7	110	110.0

Class: responded (Nom)

Status: OK

CLASSIFY THE DATA WITH WEKA
DECISION TREE MODEL TRAINING

- Classify Tab;
- Use training set;
- Class = responded;
- Start building the model



J48 -C 0.25 -M 2

● Use trai
○ Supplie t
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) responded

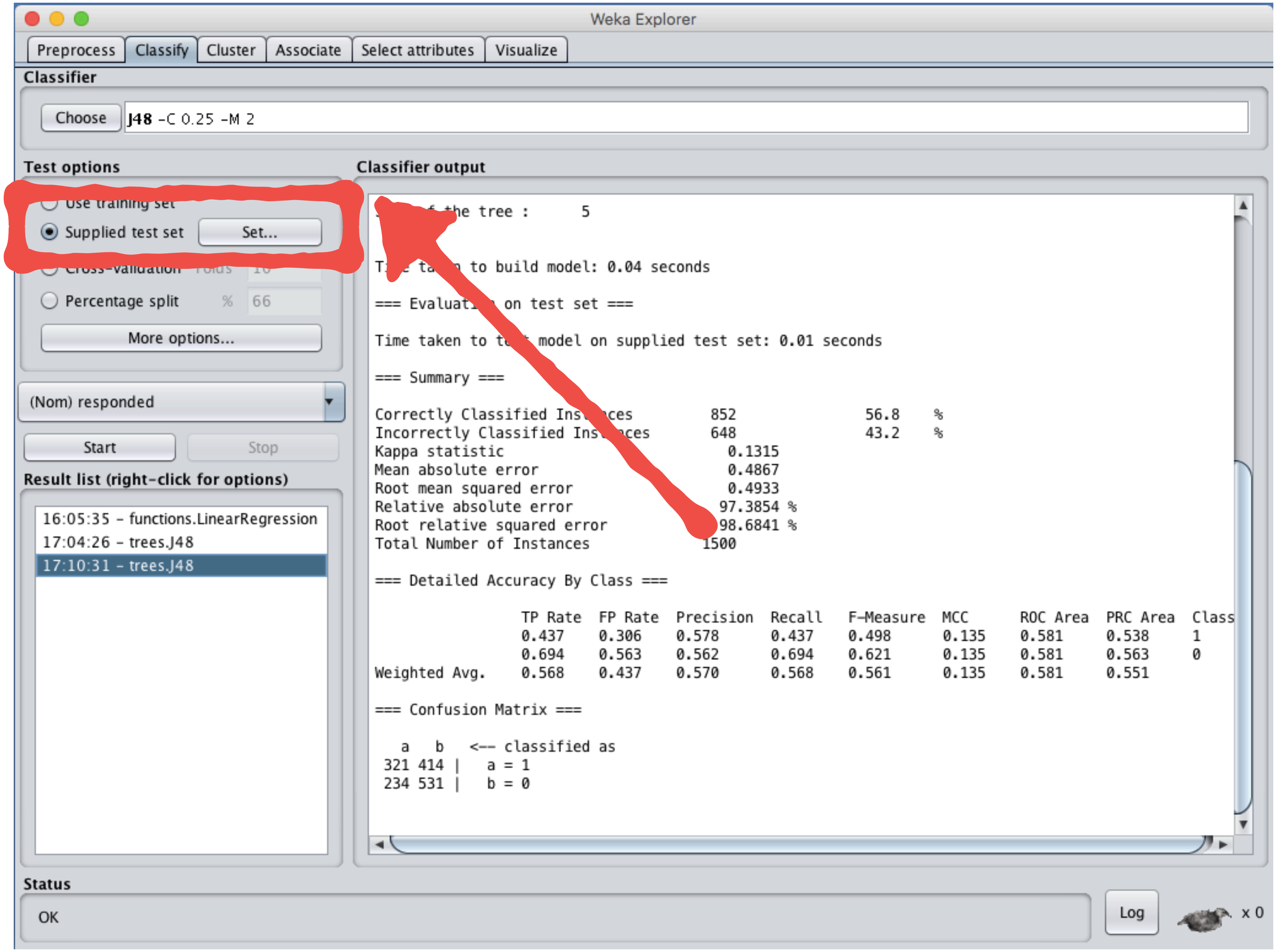
16:05:35 - functions.LinearRegression
17:04:26 - trees.J48

```
Size of the tree :      5


Time taken to build model: 0.06 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances         852               56.8    %
Incorrectly Classified Instances       648               43.2    %
Kappa statistic                          0.1315
Mean absolute error                      0.4867
Root mean squared error                  0.4933
Relative absolute error                 97.3854 %
Root relative squared error             98.6841 %
Total Number of Instances             1500

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.437    0.306    0.578      0.437    0.498      0.135    0.581     0.538     1
                 0.694    0.563    0.562      0.694    0.621      0.135    0.581     0.563     0
Weighted Avg.    0.568    0.437    0.570      0.568    0.561      0.135    0.581     0.551

=== Confusion Matrix ===

   a   b   <-- classified as
 321 414 |   a = 1
 234 531 |   b = 0
```

OK

CLASSIFY THE DATA WITH WEKA DECISION TREE MODEL TESTING

- Supplied test set;



Classifier: J48 -C 0.25 -M 2

Test options:

- ○ Use training set
- ● Supplied test set
- ○ Cross-validation Folds 10
- ○ Percentage split % 66

(Nom) responded

Result list:

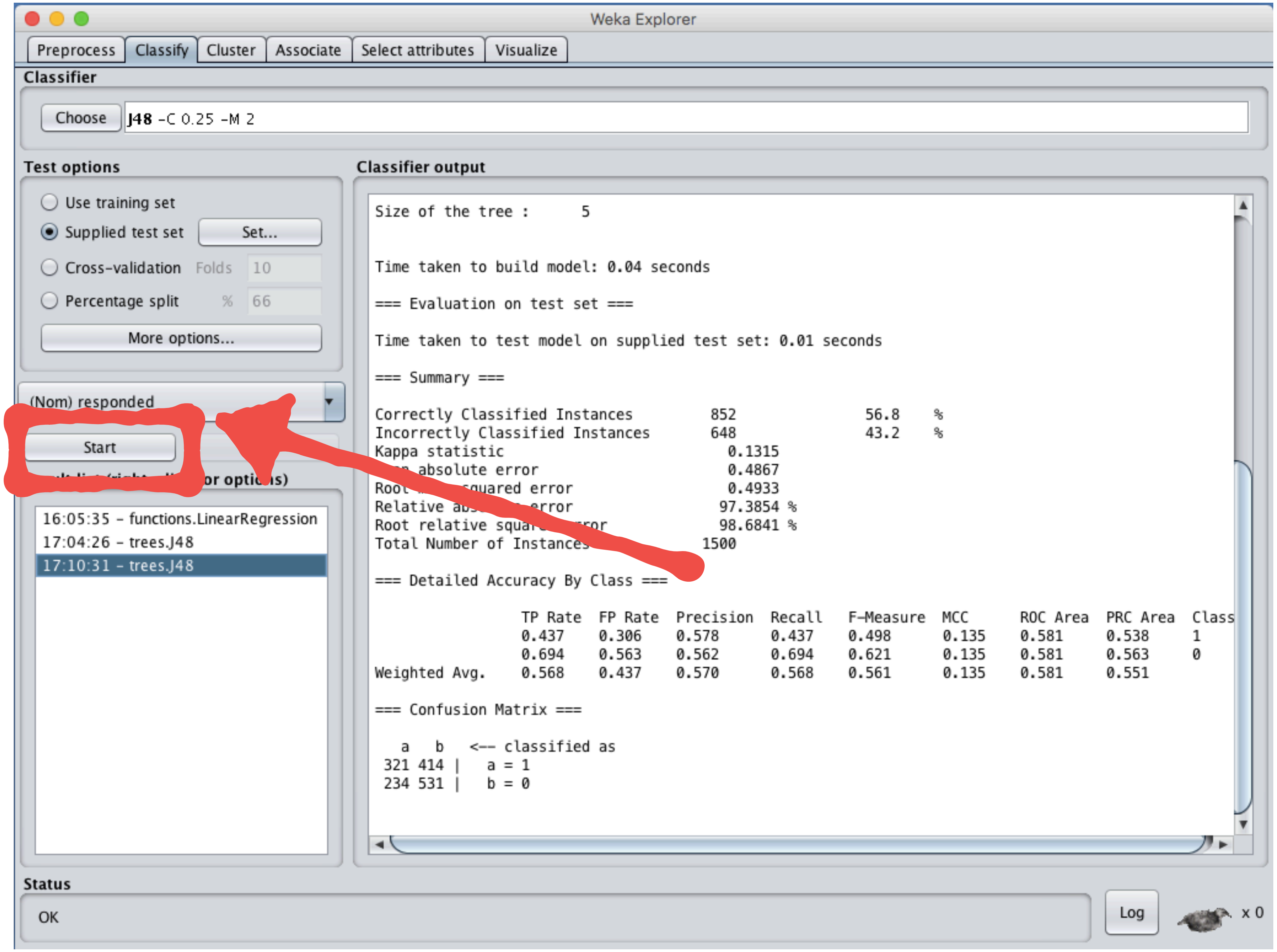
- 16:05:35 - functions.LinearRegression
- 17:04:26 - trees.J48
- 17:10:31 - trees.J48

Status: OK

```
  f the tree :      5


Ti e ta   n to build model: 0.04 seconds

=== Evaluati   on test set ===

Time taken to te   model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Ins   ces        852               56.8    %
Incorrectly Classified Ins   nces      648               43.2    %
Kappa statistic                          0.1315
Mean absolute error                      0.4867
Root mean squared error                  0.4933
Relative absolute error                 97.3854 %
Root relative squared error             98.6841 %
Total Number of Instances             1500

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.437    0.306    0.578      0.437    0.498      0.135    0.581     0.538     1
                 0.694    0.563    0.562      0.694    0.621      0.135    0.581     0.563     0
Weighted Avg.    0.568    0.437    0.570      0.568    0.561      0.135    0.581     0.551

=== Confusion Matrix ===

   a   b   <-- classified as
 321 414 |   a = 1
 234 531 |   b = 0
```

CLASSIFY THE DATA WITH WEKA
DECISION TREE MODEL TESTING

- Supplied test set;
- Start testing the model;



J48 -C 0.25 -M 2

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) responded

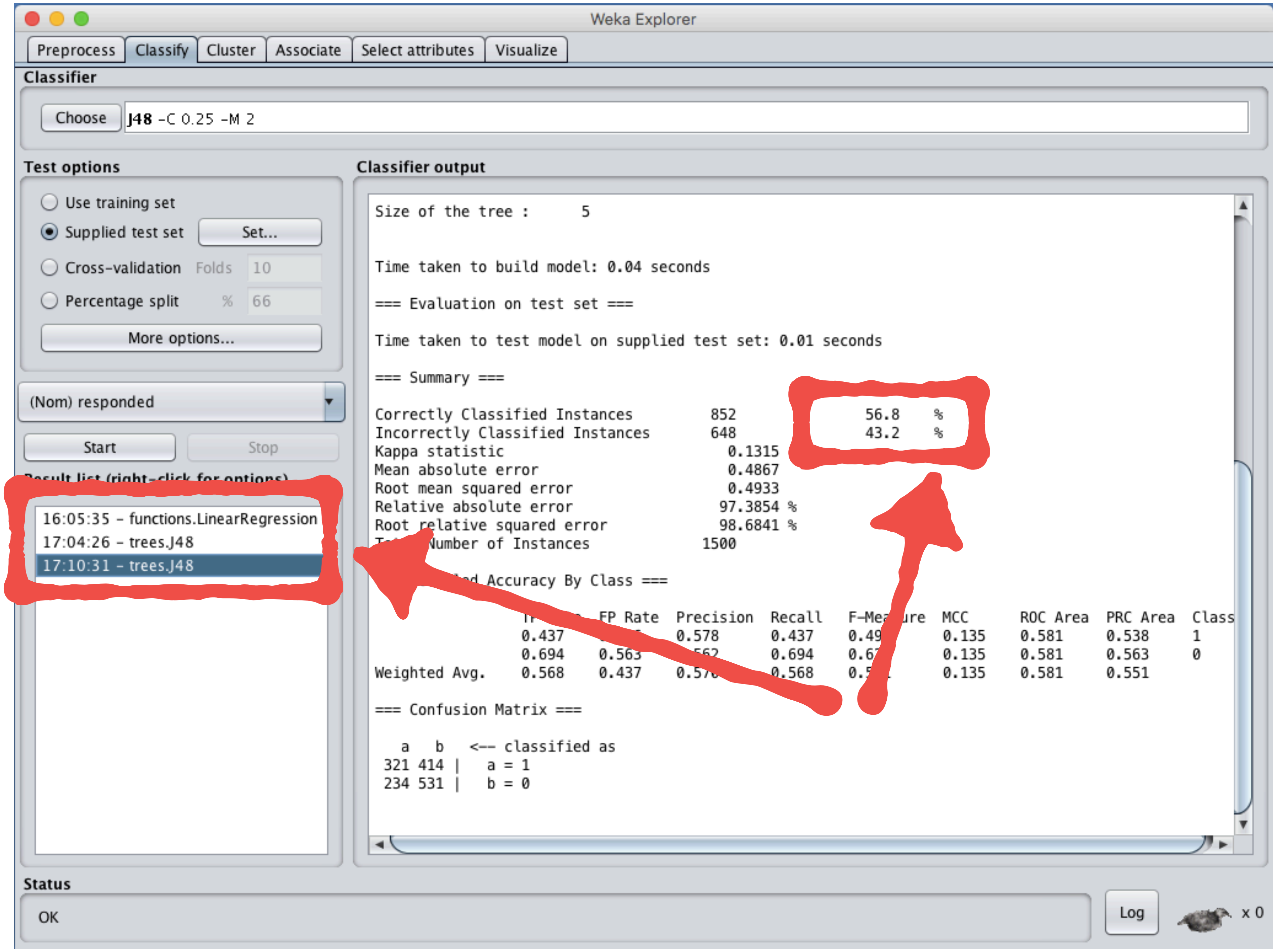
16:05:35 - functions.LinearRegression
17:04:26 - trees.J48
17:10:31 - trees.J48

```
Size of the tree :      5


Time taken to build model: 0.04 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         852               56.8    %
Incorrectly Classified Instances       648               43.2    %
Kappa statistic                          0.1315
Mean absolute error                      0.4867
Root mean squared error                  0.4933
Relative absolute error                 97.3854 %
Root relative squared error             98.6841 %
Total Number of Instances             1500

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.437    0.306    0.578      0.437    0.498      0.135    0.581     0.538     1
                 0.694    0.563    0.562      0.694    0.621      0.135    0.581     0.563     0
Weighted Avg.    0.568    0.437    0.570      0.568    0.561      0.135    0.581     0.551

=== Confusion Matrix ===

   a   b   <-- classified as
 321 414 |   a = 1
 234 531 |   b = 0
```

Status: OK

CLASSIFY THE DATA WITH WEKA
DECISION TREE MODEL TESTING

- Supplied test set;
- Start testing the model;
- Compare models accuracy between train and test.



J48 -C 0.25 -M 2

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) responded

16:05:35 - functions.LinearRegression
17:04:26 - trees.J48
17:10:31 - trees.J48

```
Size of the tree :      5


Time taken to build model: 0.04 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         852               56.8    %
Incorrectly Classified Instances       648               43.2    %
Kappa statistic                          0.1315
Mean absolute error                      0.4867
Root mean squared error                  0.4933
Relative absolute error                 97.3854 %
Root relative squared error             98.6841 %
Total Number of Instances             1500

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.437    [illegible]  0.578      0.437    0.49       0.135    0.581     0.538     1
                 0.694    0.563    0.562      0.694    0.6        0.135    0.581     0.563     0
Weighted Avg.    0.568    0.437    0.57       0.568    0.5        0.135    0.581     0.551

=== Confusion Matrix ===

   a   b   <-- classified as
 321 414 |   a = 1
 234 531 |   b = 0
```

Status: OK

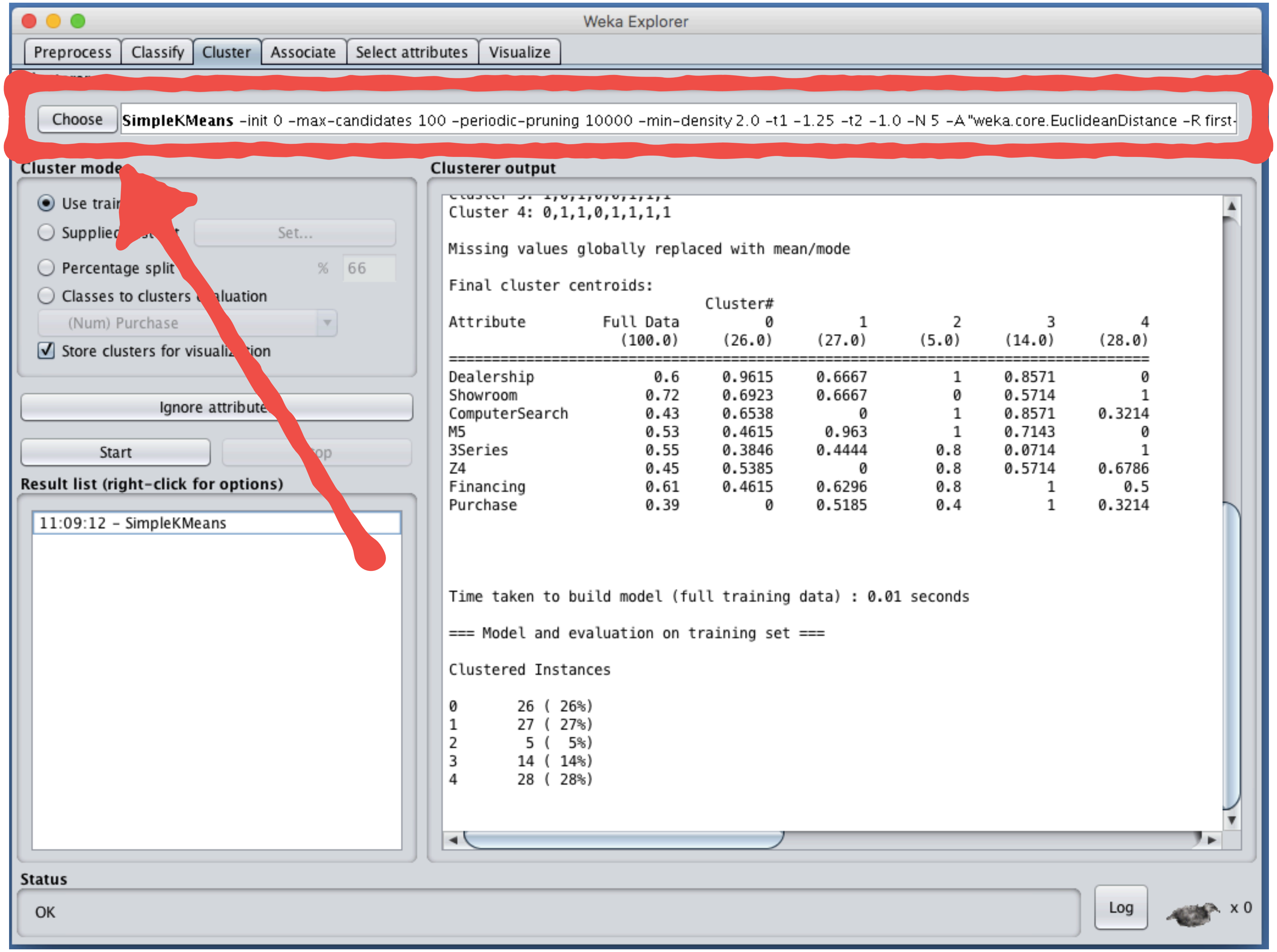
AN EXAMPLE OF CLUSTERING
CAR DEALERSHIP BEHAVIOUR ANALYSIS

The dealership has **kept track of how people walk through the dealership** and the showroom, what cars they look at, and how often they ultimately make purchases.

They are hoping to **mine this data by finding patterns** in the data and by using clusters to determine **if certain behaviours in their customers emerge**.

CLUSTERING THE DATA WITH WEKA
K-MEANS BEHAVIOUR ANALYSIS

- Cluster Tab;
- Use training set;
- No Class;
- Start



SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 5 -A "weka.core.EuclideanDistance -R first-

● Use trai
○ Supplie
○ Percentage split % 66
○ Classes to clusters luation
(Num) Purchase
☑ Store clusters for visuali ion

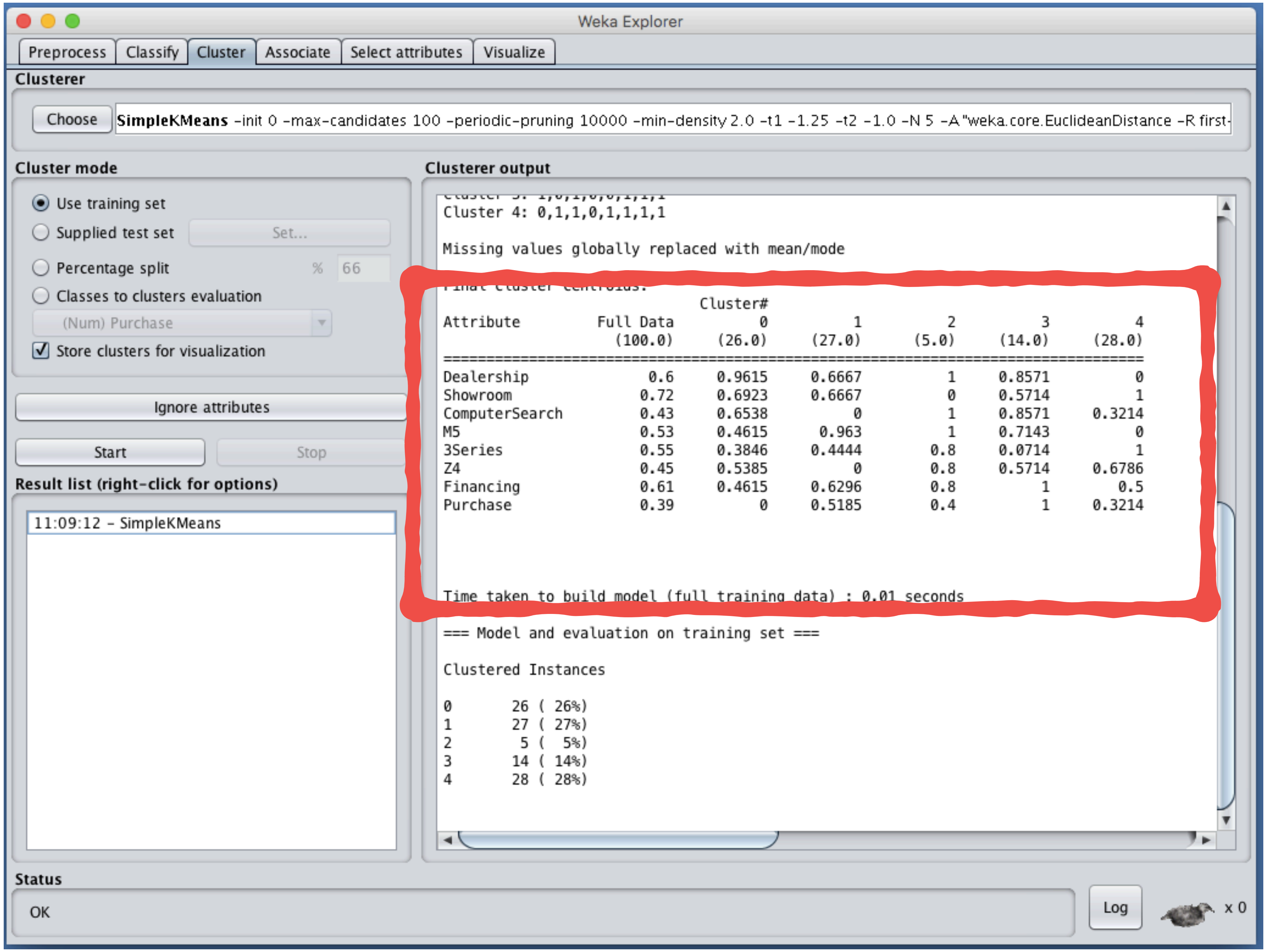
11:09:12 - SimpleKMeans

```
Cluster 3: 1,0,1,0,0,1,1,1
Cluster 4: 0,1,1,0,1,1,1,1

Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1          2          3          4
               (100.0)     (26.0)     (27.0)      (5.0)     (14.0)     (28.0)
================================================================================
Dealership         0.6     0.9615     0.6667          1     0.8571          0
Showroom          0.72     0.6923     0.6667          0     0.5714          1
ComputerSearch    0.43     0.6538          0          1     0.8571     0.3214
M5                0.53     0.4615      0.963          1     0.7143          0
3Series           0.55     0.3846     0.4444        0.8     0.0714          1
Z4                0.45     0.5385          0        0.8     0.5714     0.6786
Financing         0.61     0.4615     0.6296        0.8          1        0.5
Purchase          0.39          0     0.5185        0.4          1     0.3214




Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       26 ( 26%)
1       27 ( 27%)
2        5 (  5%)
3       14 ( 14%)
4       28 ( 28%)
```

OK

CLUSTERING THE DATA WITH WEKA
K-MEANS BEHAVIOUR ANALYSIS

- Cluster Tab;
- Use training set;
- No Class;
- Start;
- Evaluate patterns.



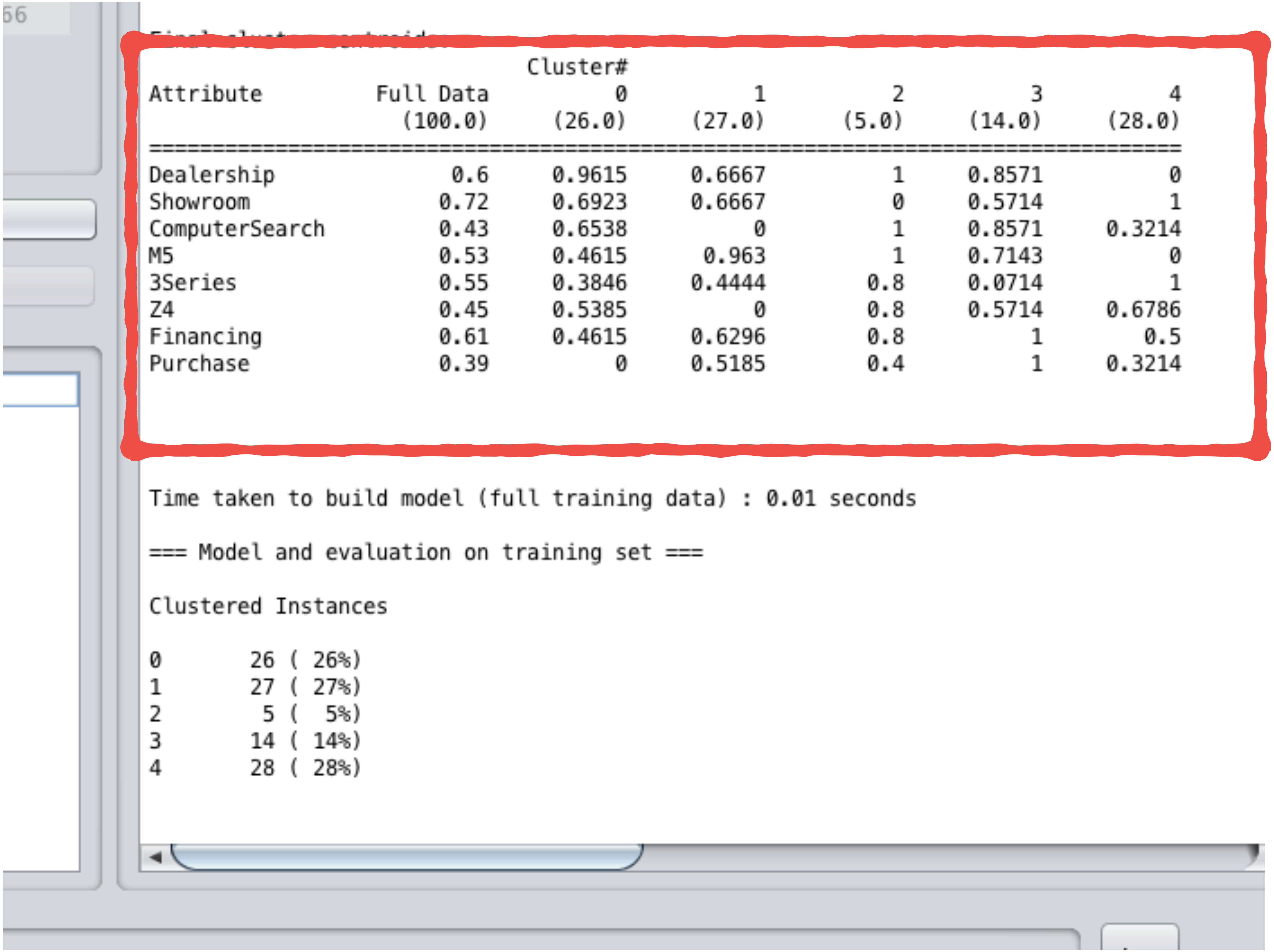
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 5 -A "weka.core.EuclideanDistance -R first-

- ● Use training set
- ○ Supplied test set Set...
- ○ Percentage split % 66
- ○ Classes to clusters evaluation
  (Num) Purchase
- ☑ Store clusters for visualization

11:09:12 - SimpleKMeans

```
Cluster 3: 1,0,1,0,0,1,1,1
Cluster 4: 0,1,1,0,1,1,1,1

Missing values globally replaced with mean/mode

Final cluster centroids:
                          Cluster#
Attribute     Full Data          0          1          2          3          4
                (100.0)     (26.0)     (27.0)      (5.0)     (14.0)     (28.0)
================================================================================
Dealership          0.6     0.9615     0.6667          1     0.8571          0
Showroom           0.72     0.6923     0.6667          0     0.5714          1
ComputerSearch     0.43     0.6538          0          1     0.8571     0.3214
M5                 0.53     0.4615      0.963          1     0.7143          0
3Series            0.55     0.3846     0.4444        0.8     0.0714          1
Z4                 0.45     0.5385          0        0.8     0.5714     0.6786
Financing          0.61     0.4615     0.6296        0.8          1        0.5
Purchase           0.39          0     0.5185        0.4          1     0.3214




Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       26 ( 26%)
1       27 ( 27%)
2        5 (  5%)
3       14 ( 14%)
4       28 ( 28%)
```

Status: OK



```
Final cluster centroids:
                             Cluster#
Attribute         Full Data         0         1         2         3         4
                    (100.0)    (26.0)    (27.0)     (5.0)    (14.0)    (28.0)
===============================================================================
Dealership              0.6    0.9615    0.6667         1    0.8571         0
Showroom               0.72    0.6923    0.6667         0    0.5714         1
ComputerSearch         0.43    0.6538         0         1    0.8571    0.3214
M5                     0.53    0.4615     0.963         1    0.7143         0
3Series                0.55    0.3846    0.4444       0.8    0.0714         1
Z4                     0.45    0.5385         0       0.8    0.5714    0.6786
Financing              0.61    0.4615    0.6296       0.8         1       0.5
Purchase               0.39         0    0.5185       0.4         1    0.3214



Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       26 ( 26%)
1       27 ( 27%)
2        5 (  5%)
3       14 ( 14%)
4       28 ( 28%)
```

EXAMPLE CONCLUSION
K-MEANS BEHAVIOUR ANALYSIS

- **Cluster 0**– This group we can call the "Dreamers," as they appear to wander around the dealership, looking at cars parked outside on the lots, but trail off when it comes to coming into the dealership, and worst of all, they don't purchase anything.
- **Cluster 1**– We'll call this group the "M5 Lovers" because they tend to walk straight to the M5s, ignoring the 3-series cars and the Z4. However, they don't have a high purchase rate – only 52 percent. This is a potential problem and could be a focus for improvement for the dealership, perhaps by sending more salespeople to the M5 section.
- **Cluster 2**– This group is so small we can call them the "Throw-Aways" because they aren't statistically relevent, and we can't draw any good conclusions from their behaviour. (This happens sometimes with clusters and may indicate that you should reduce the number of clusters you've created).
- **Cluster 3**– This group we'll call the "BMW Babies" because they always end up purchasing a car and always end up financing it. Here's where the data shows us some interesting things: It appears they walk around the lot looking at cars, then turn to the computer search available at the dealership. Ultimately, they tend to buy M5s or Z4s (but never 3-series). This cluster tells the dealership that it should consider making its search computers more prominent around the lots (outdoor search computers?), and perhaps making the M5 or Z4 much more prominent in the search results. Once the customer has made up his mind to purchase the vehicle, he always qualifies for financing and completes the purchase.
- **Cluster 4**– This group we'll call the "Starting Out With BMW" because they always look at the 3-series and never look at the much more expensive M5. They walk right into the showroom, choosing not to walk around the lot and tend to ignore the computer search terminals. While 50 percent get to the financing stage, only 32 percent ultimately finish the transaction. The dealership could draw the conclusion that these customers looking to buy their first BMWs know exactly what kind of car they want (the 3-series entry-level model) and are hoping to qualify for financing to be able to afford it. The dealership could possibly increase sales to this group by relaxing their financing standards or by reducing the 3-series prices.

NEURAL NETWORKS VS. K-NEAREST NEIGHBOUR BREAST CANCER CLASSIFICATION

Lets **look at the differences** between adopting two classification techniques, one will be the **neural network multilayer perceptron** classification, compared with **k-nearest neighbour model**.

- Dataset: breast-cancer.arff (286 instances)
- Nominal Class : no-recurrence-events | recurrence-events
- Classification with: Multilayer Perceptron | KNN (IBk in Weka)
- Cross Validation Test Options: takes 10% of the dataset for testing in 10 folds, computing the mean.
- Pay attention to the False Positive rate, which are the differences?

EXERCISE

Apply K-nearest neighbour Model (IBk in the choose button) to the problem of car dealership:

- **Start parameter KNN = 5;**
- **Try decreasing KNN.**

REFERENCES

- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. [PDF]
- The WEKA software. [LINK]
- Wikipedia – Multivariate Linear Regression Model. [LINK]
- Wikipedia – Decision Tree. [LINK]
- Wikipedia – K-means. [LINK]
- Gary Marcus, "*Deep Learning: A Critical Appraisal*", arXiv:1801.00631.
- Wikipedia – K-nearest neighbour. [LINK]
- Web version of the tutorial by Michael Abernethy, IBM. – [LINK]