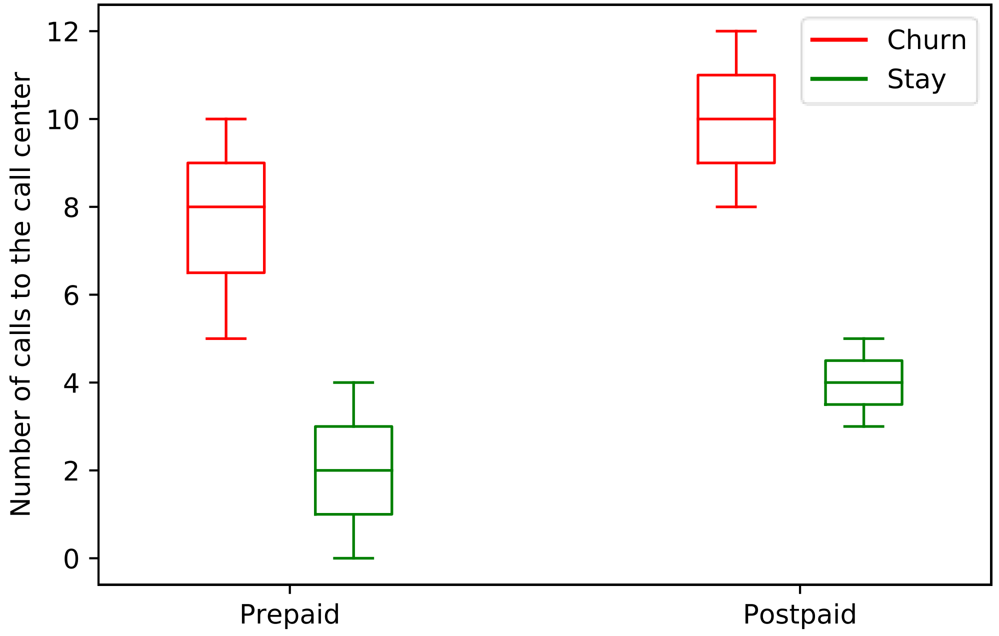
DECISION TREE AND RANDOM FOREST

Ratchainant Thammasudjarit, Ph.D.

- Visualization



Decision Tree

Given data

Group	Call	Churn
Prepaid	5	Yes
Prepaid	8	Yes
Prepaid	10	Yes
Prepaid	0	No
Prepaid	2	No
Prepaid	4	No
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes
Postpaid	3	No
Postpaid	4	No
Postpaid	5	No

- Concepts

Group	Call	Churn
Prepaid	5	Yes
Prepaid	8	Yes
Prepaid	10	Yes
Prepaid	0	No
Prepaid	2	No
Prepaid	4	No
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes
Postpaid	3	No
Postpaid	4	No
Postpaid	5	No

Call

≥ 5

Group	Call	Churn
Prepaid	5	Yes
Prepaid	8	Yes
Prepaid	10	Yes
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes
Postpaid	5	No

< 5

Group	Call	Churn
Prepaid	0	No
Prepaid	2	No
Prepaid	4	No
Postpaid	3	No
Postpaid	4	No

Note: In this example, **call** refers to the number of calls to the call center

Decision Tree

The decision tree makes data partitioning until no more data to be partitioned

- Concepts

Call

≥ 5

< 5

Group	Call	Churn
Prepaid	5	Yes
Prepaid	8	Yes
Prepaid	10	Yes
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes
Postpaid	5	No

Which criteria to use for the next splitting

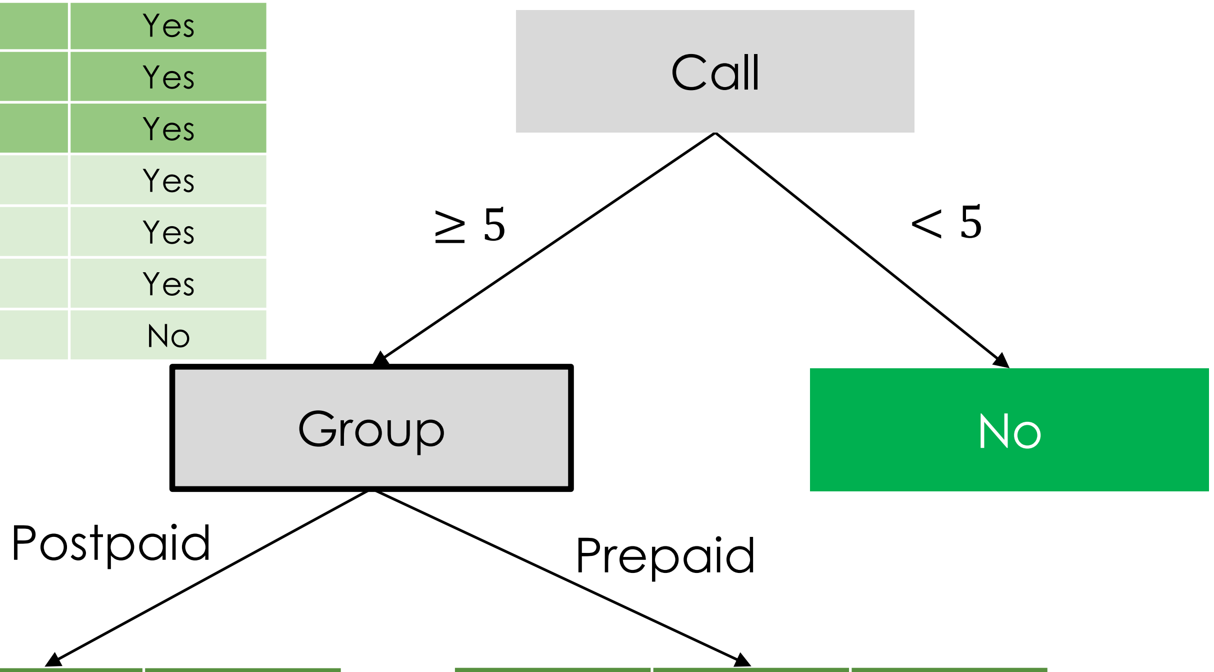
No

Group	Call	Churn
Prepaid	0	No
Prepaid	2	No
Prepaid	4	No
Postpaid	3	No
Postpaid	4	No

Decision Tree

The decision tree makes data partitioning until no more data to be partitioned

Group	Call	Churn
Prepaid	5	Yes
Prepaid	8	Yes
Prepaid	10	Yes
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes
Postpaid	5	No



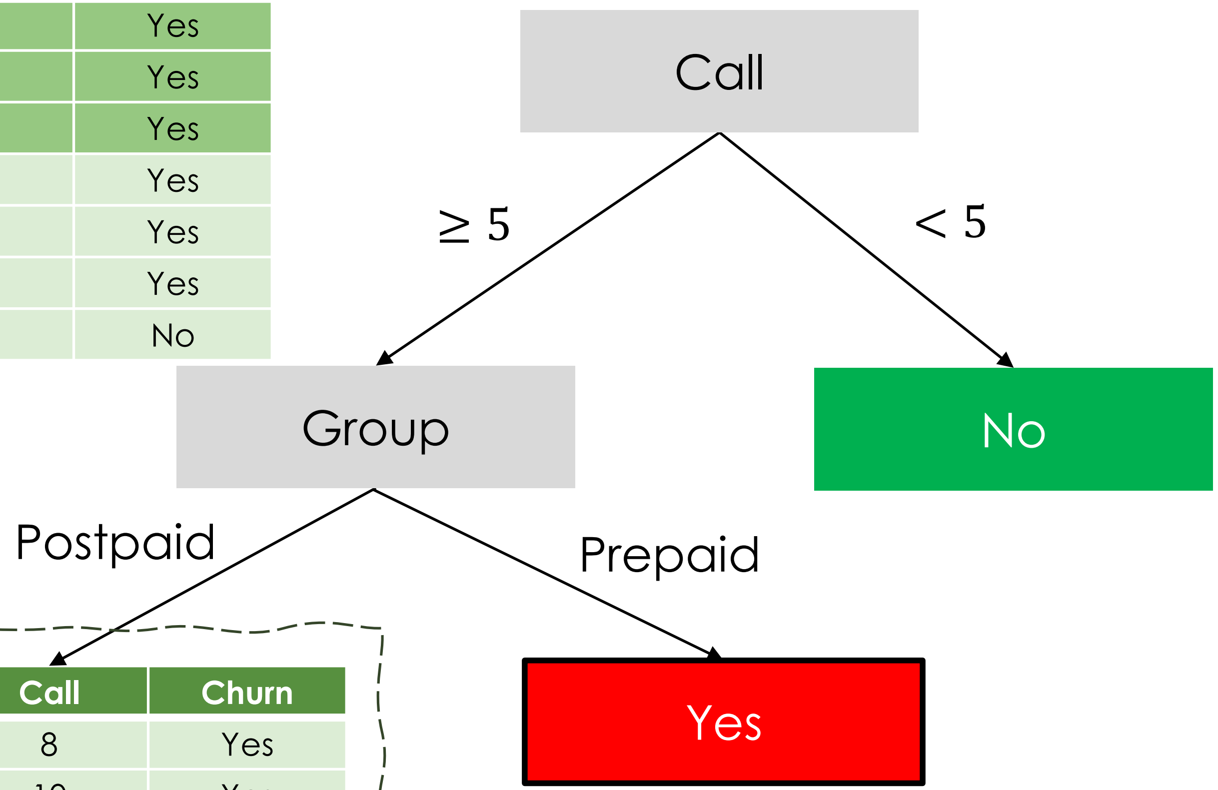
Group	Call	Churn
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes
Postpaid	5	No

Group	Call	Churn
Prepaid	5	Yes
Prepaid	8	Yes
Prepaid	10	Yes

Decision Tree

The decision tree makes data partitioning until no more data to be partitioned

Group	Call	Churn
Prepaid	5	Yes
Prepaid	8	Yes
Prepaid	10	Yes
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes
Postpaid	5	No



Group	Call	Churn
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes
Postpaid	5	No

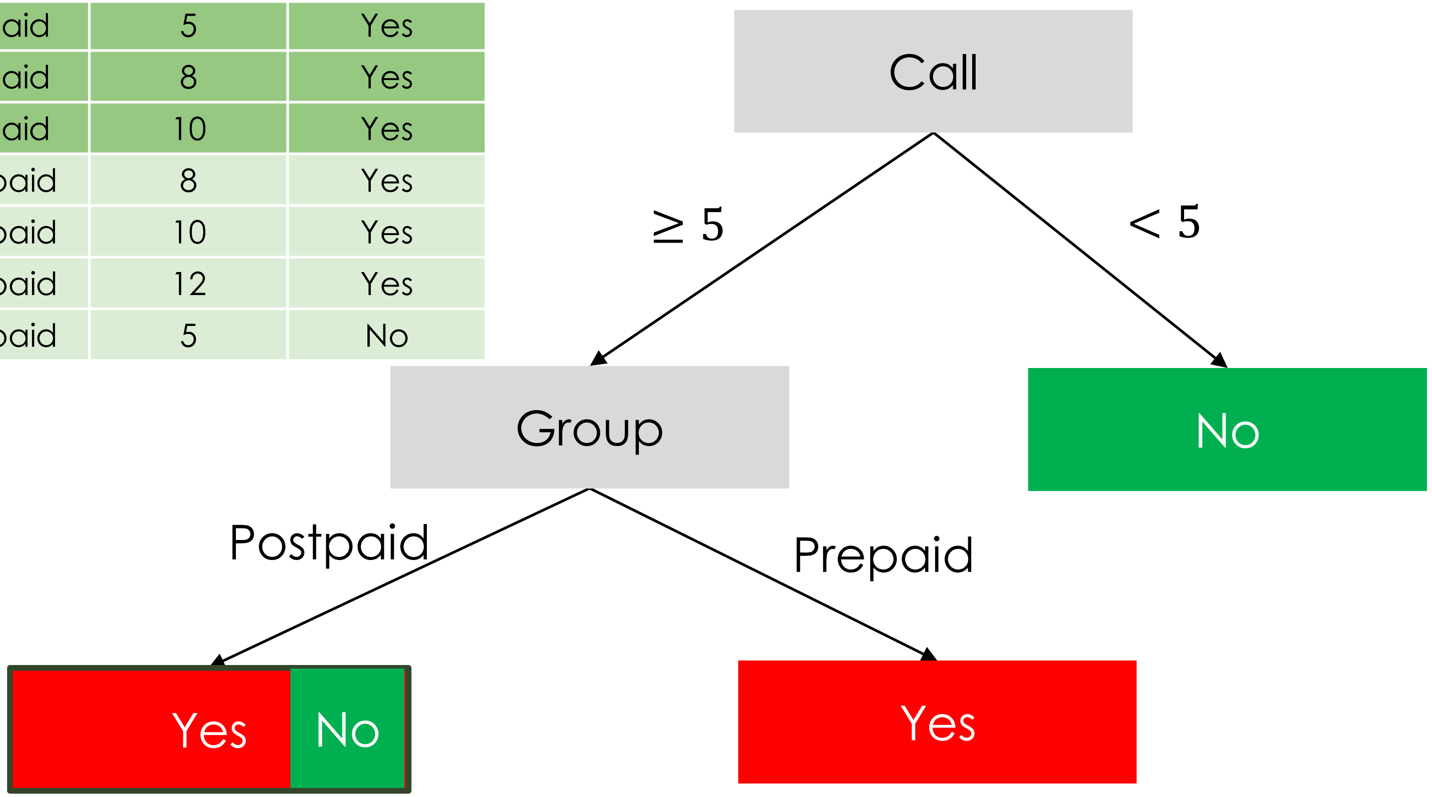
Group	Call	Churn
Prepaid	5	Yes
Prepaid	8	Yes
Prepaid	10	Yes

Which criteria to use for the next splitting

Decision Tree

The decision tree makes data partitioning until no more data to be partitioned

Group	Call	Churn
Prepaid	5	Yes
Prepaid	8	Yes
Prepaid	10	Yes
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes
Postpaid	5	No



Group	Call	Churn
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes
Postpaid	5	No

Group	Call	Churn
Prepaid	5	Yes
Prepaid	8	Yes
Prepaid	10	Yes

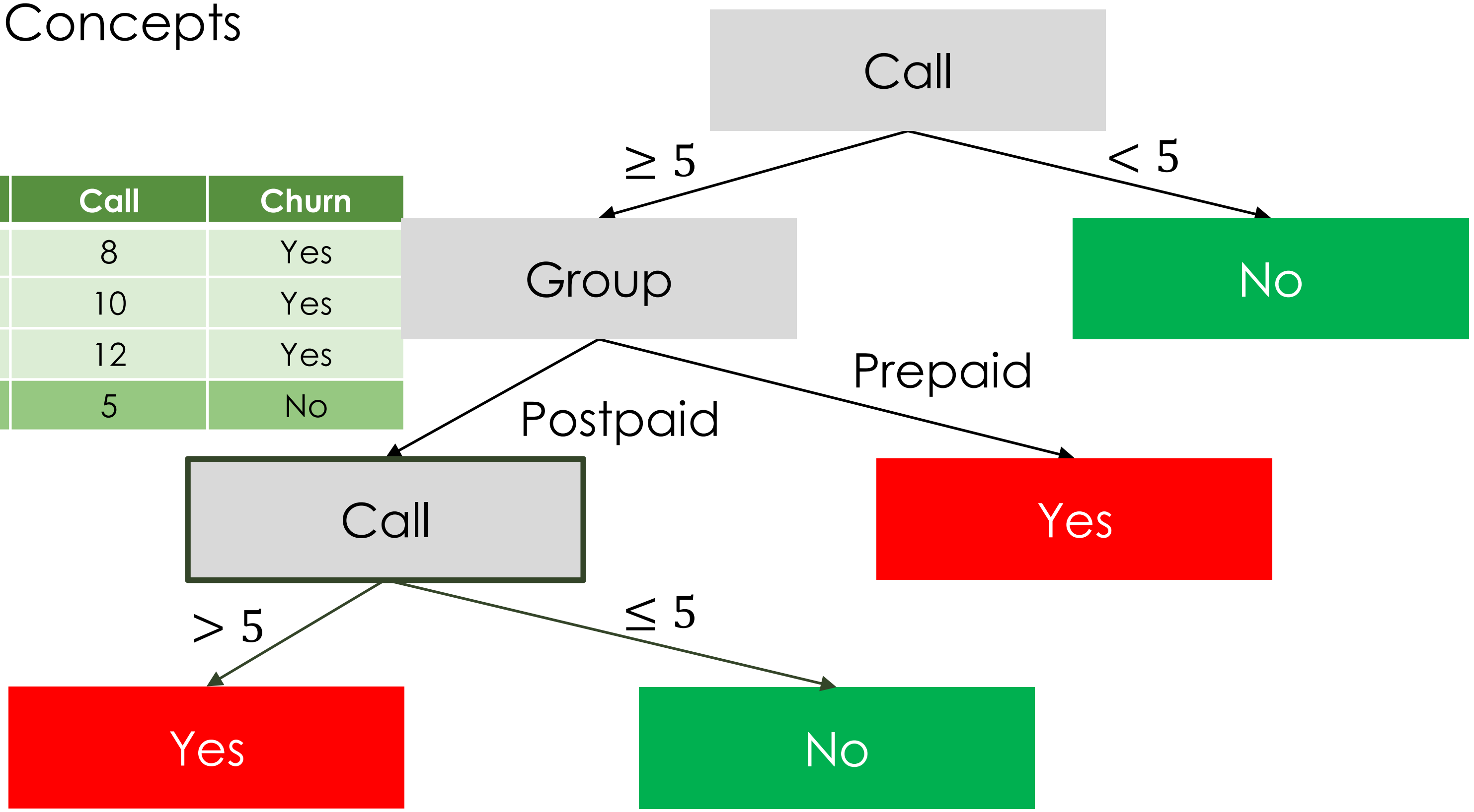
If the maximum depth is defined as 2, the decision tree will stop learning at this stage

Decision Tree

The decision tree makes data partitioning until no more data to be partitioned

- Concepts

Group	Call	Churn
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes
Postpaid	5	No



Group	Call	Churn
Postpaid	8	Yes
Postpaid	10	Yes
Postpaid	12	Yes

Group	Call	Churn
Postpaid	5	No

Otherwise, the decision tree will keep partitioning data until the end

Decision Tree

The decision tree makes data partitioning until no more data to be partitioned

Impurity Measure

Impurity measure is applied to make decision which criteria to use for splitting

- Impurity measures for classification task
  - Entropy
  - Gini
  - Classification error
- Impurity measure for regression task
  - Variance

Impurity Measure

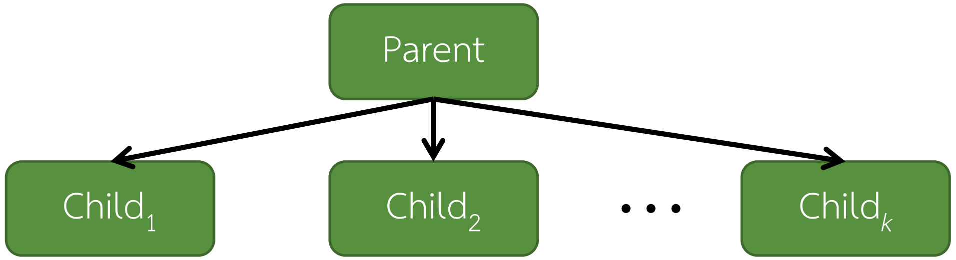
Impurity measure is applied to make decision which criteria to use for splitting

- Entropy

$$H(x) = \sum_j -P(j)log_2 P(j)$$

where j is any possible value in a given feature
$P(j)$ is the probability of j

- Splitting



$$IG(Parent, Children) = H(Parent) - \sum_i \frac{N_{Child_k}}{N_{parent}} \cdot H(Child_i)$$

Where $H(\cdot)$ is entropy of a particular node
$N_{parrent}$ is the number of datapoint of the parent node
N_{child_k} is the number of datapoint of the k^{th} child node

- Entropy (Example): Measure Entropy of Call

Call	Churn
5	Yes
8	Yes
10	Yes
0	No
2	No
4	No
8	Yes
10	Yes
12	Yes
3	No
4	No
5	No

Call

$$H(parent) = -\underbrace{\frac{6}{12}}_{P(churn=Yes)}\log_2 \underbrace{\frac{6}{12}}_{P(churn=Yes)} - \underbrace{\frac{6}{12}}_{P(churn=No)}\log_2 \underbrace{\frac{6}{12}}_{P(churn=No)} = 1$$

Impurity Measure

Impurity measure is applied to make decision which criteria to use for splitting

- Entropy (Example): Measure Entropy of Call

Call

$H(parent) = 1$

≥ 5

Call	Churn
5	Yes
8	Yes
10	Yes
8	Yes
10	Yes
12	Yes
5	No

< 5

Call	Churn
0	No
2	No
4	No
3	No
4	No

$$H(call < 5) = -\frac{5}{5}\log_2\frac{5}{5} - \frac{5}{5}\log_2\frac{5}{5} = 0$$

$$H(call \geq 5) = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7} = 0.59$$

$$IG(parent, children) = H(parent) - \frac{N_{call\geq 5}}{N_{parent}} \cdot H(call \geq 5) - \frac{N_{call<5}}{N_{parent}} \cdot H(call < 5)$$

$$= 1 - \frac{7}{12}(0.59) - \frac{5}{12}(0)$$

$$= 0.65$$

Impurity Measure

Impurity measure is applied to make decision which criteria to use for splitting

Impurity Measure

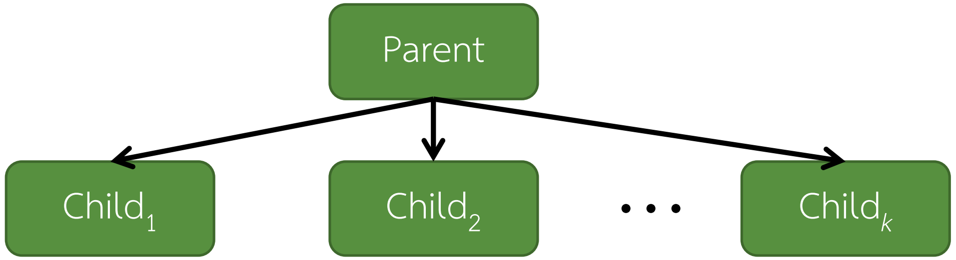
Impurity measure is applied to make decision which criteria to use for splitting

- Gini

$$G(x) = 1 - \sum_j P(j)^2$$

where j is any possible value in a given feature
$P(j)$ is the probability of j

- Splitting



$$Gini(Parent, Children) = G(Parent) - \sum_i \frac{N_{Child_k}}{N_{parent}} \cdot G(Child_i)$$

Where $G(\cdot)$ is Gini of a particular node
$N_{parrent}$ is the number of datapoint of the parent node
N_{child_k} is the number of datapoint of the k^{th} child node

- Gini (Example): Measure Gini of Call

Call	Churn
5	Yes
8	Yes
10	Yes
0	No
2	No
4	No
8	Yes
10	Yes
12	Yes
3	No
4	No
5	No

Call

$$G(parent) = 1 - \left(\underbrace{\frac{6}{12}}_{P(churn = Yes)}\right)^2 - \left(\underbrace{\frac{6}{12}}_{P(churn = No)}\right)^2 = 0.5$$

Impurity Measure

Impurity measure is applied to make decision which criteria to use for splitting

- Gini (Example): Measure Gini of Call

Call $\quad G(parent) = 0.5$

≥ 5 | < 5

Call	Churn
5	Yes
8	Yes
10	Yes
8	Yes
10	Yes
12	Yes
5	No

Call	Churn
0	No
2	No
4	No
3	No
4	No

$$G(call < 5) = 1 - \left(\frac{5}{5}\right)^2 = 0$$

$$G(call \geq 5) = 1 - \left(\frac{1}{7}\right)^2 - \left(\frac{6}{7}\right)^2 = 0.24$$

$$G(parent, children) = G(parent) - \frac{N_{call\geq 5}}{N_{parent}} \cdot G(call \geq 5) - \frac{N_{call<5}}{N_{parent}} \cdot G(call < 5)$$

$$= 0.5 - \frac{7}{12}(0.24) - \frac{5}{12}(0)$$

$$= 0.36$$

Impurity Measure

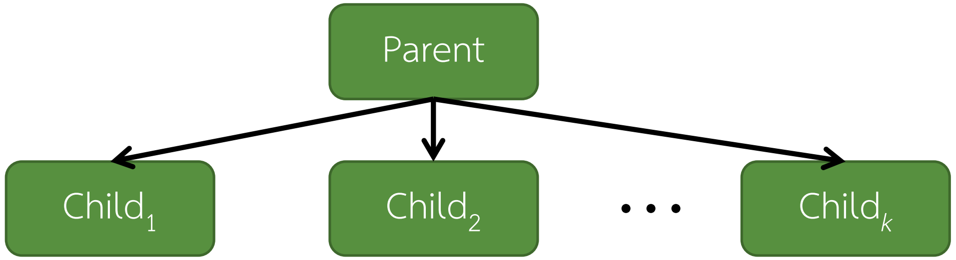
Impurity measure is applied to make decision which criteria to use for splitting

- Classification Error

$$E(x) = 1 - \max\{P(j)\}$$

where j is any possible value in a given feature
$P(j)$ is the probability of j

- Splitting



$$IG(Parent, Children) = E(Parent) - \sum_{i} \frac{N_{Child_k}}{N_{parent}} \cdot E(Child_i)$$

Where $E(\cdot)$ is classification error of a particular node
$N_{parrent}$ is the number of datapoint of the parent node
N_{child_k} is the number of datapoint of the k^{th} child node

Impurity Measure

Impurity measure is applied to make decision which criteria to use for splitting

- Classification Error (Example): Measure Classification error of Call

Call	Churn
5	Yes
8	Yes
10	Yes
0	No
2	No
4	No
8	Yes
10	Yes
12	Yes
3	No
4	No
5	No

Call $E(parent) = 1 - \frac{6}{12} = 0.5$

Impurity Measure

Impurity measure is applied to make decision which criteria to use for splitting

- Classification Error (Example): Measure Classification error of Call

Call $E(parent) = 0.5$

≥ 5

Call	Churn
5	Yes
8	Yes
10	Yes
8	Yes
10	Yes
12	Yes
5	No

$$E(call \geq 5) = 1 - \frac{6}{7} = 0.14$$

< 5

Call	Churn
0	No
2	No
4	No
3	No
4	No

$$E(call < 5) = 1 - 1 = 0$$

$$E(parent, children) = E(parent) - \frac{N_{call \geq 5}}{N_{parent}} \cdot E(call \geq 5) - \frac{N_{call<5}}{N_{parent}} \cdot E(call < 5)$$

$$= 0.5 - \frac{7}{12}(0.14) - \frac{5}{12}(0)$$

$$= 0.41$$

Impurity Measure

Impurity measure is applied to make decision which criteria to use for splitting

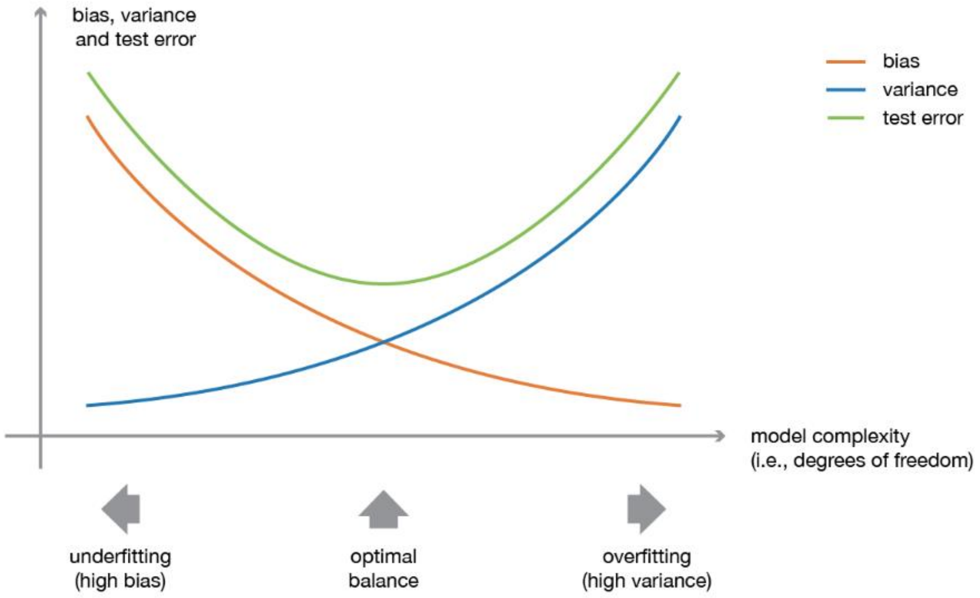
Random Forest

Ensemble Technique

Ensemble Learning

Combining multiple weak models is outperform a single strong model

- Theory
  - **Weak learners** (or **base models**) models can be used as building blocks for designing more complex models by combining several of them
  - Basic Idea:
    - Trying reducing bias and/or variance of such weak learners by combining several of them together
    - Such combination creates a **strong learner** (or **ensemble model**) that achieves better performances



Bias and Variance Trade-off

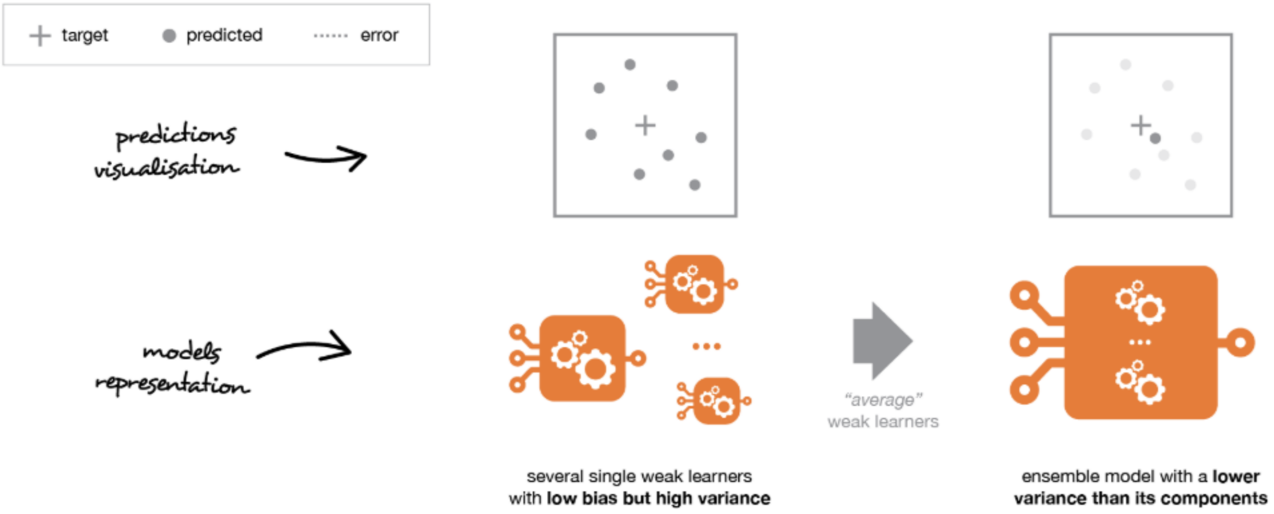
Ensemble Learning

Three ensemble concepts

- Bagging
- Boosting
- Stacking

- Bagging
  - Homogeneous weak learners
  - Learn independently in parallel then combine output using averaging method
- Boosting
  - Homogeneous weak learners
  - Learn sequentially in adaptive way then combine output using specific strategy
- Stacking
  - Heterogeneous weak learners
  - Learn in dependently in parallel then train the meta-model from the output of weak learners

- Concepts

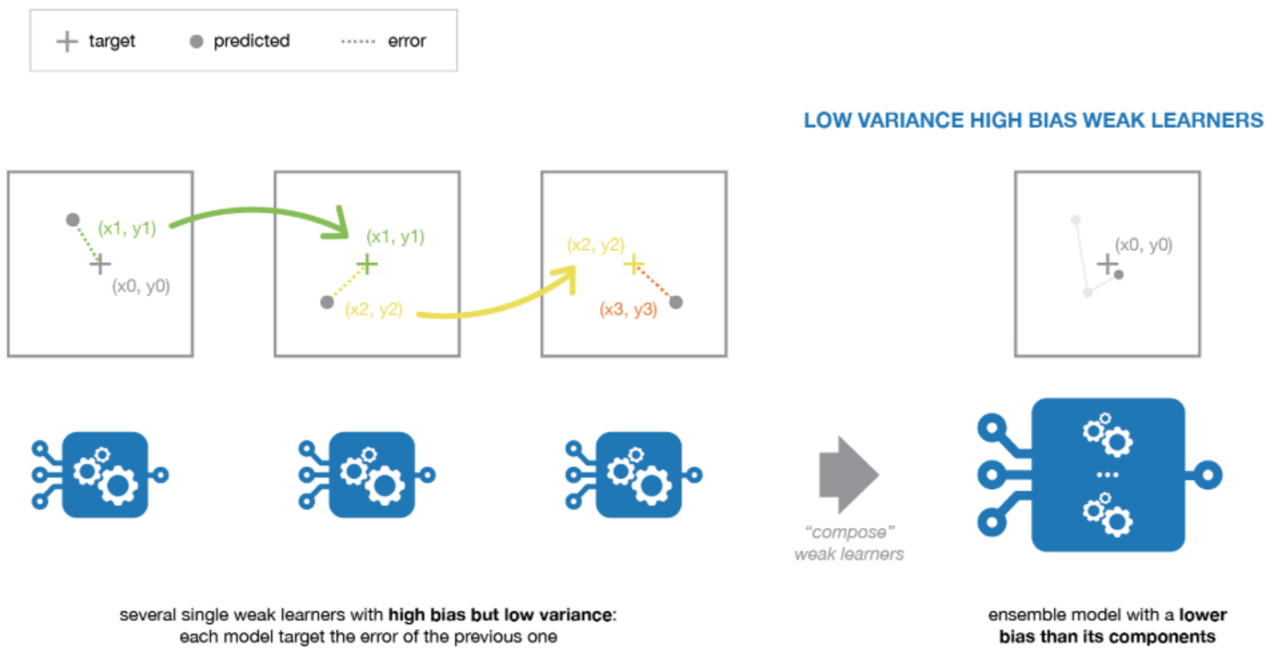


LOW BIAS HIGH VARIANCE WEAK LEARNERS

Ensemble Learning

Bagging employs **homogeneous weak learners** to **learn independently in parallel** then **combine output using averaging method**

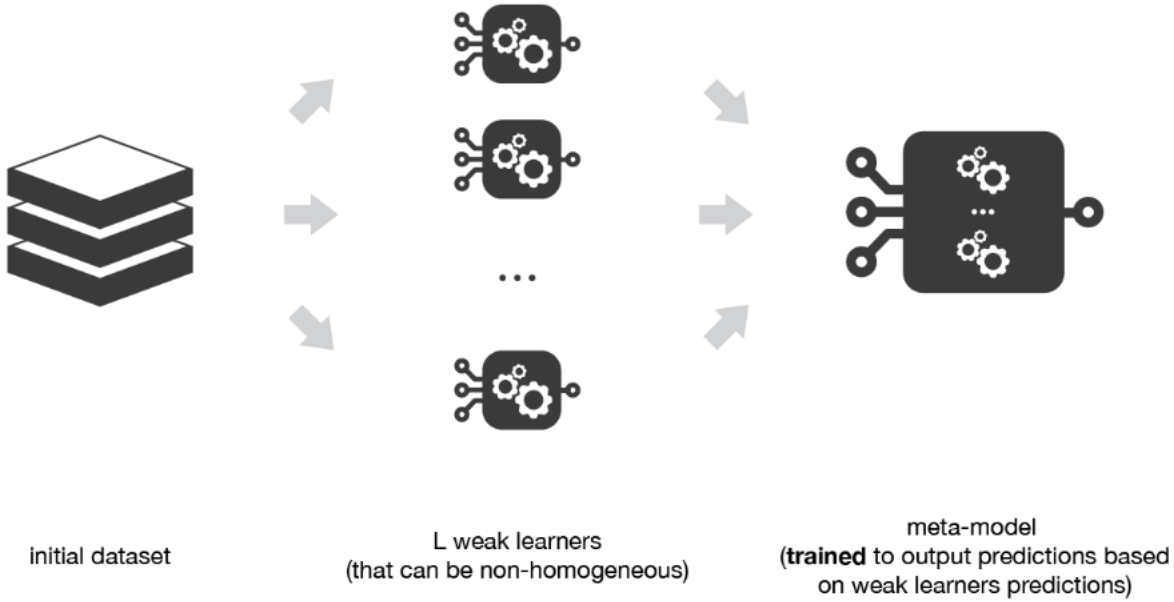
◦ Concepts



Ensemble Learning

Boosting employs **homogeneous weak learners** to **learn sequentially** then **combine output using specific strategy**

◦ Concepts



Ensemble Learning

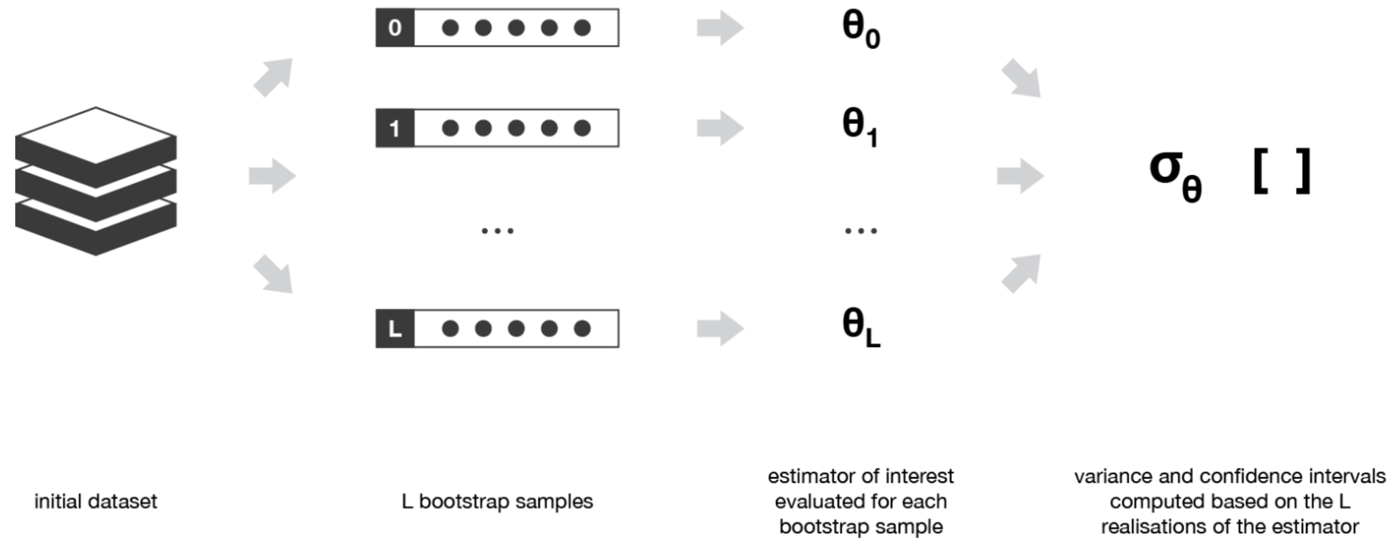
Stacking employs **heterogeneous weak learners** to **learn independently in parallel** then **train the meta-model from the output of weak learners**

Training Bagging Ensemble

Bootstrap sampling

- Concepts: Random sampling with replacement

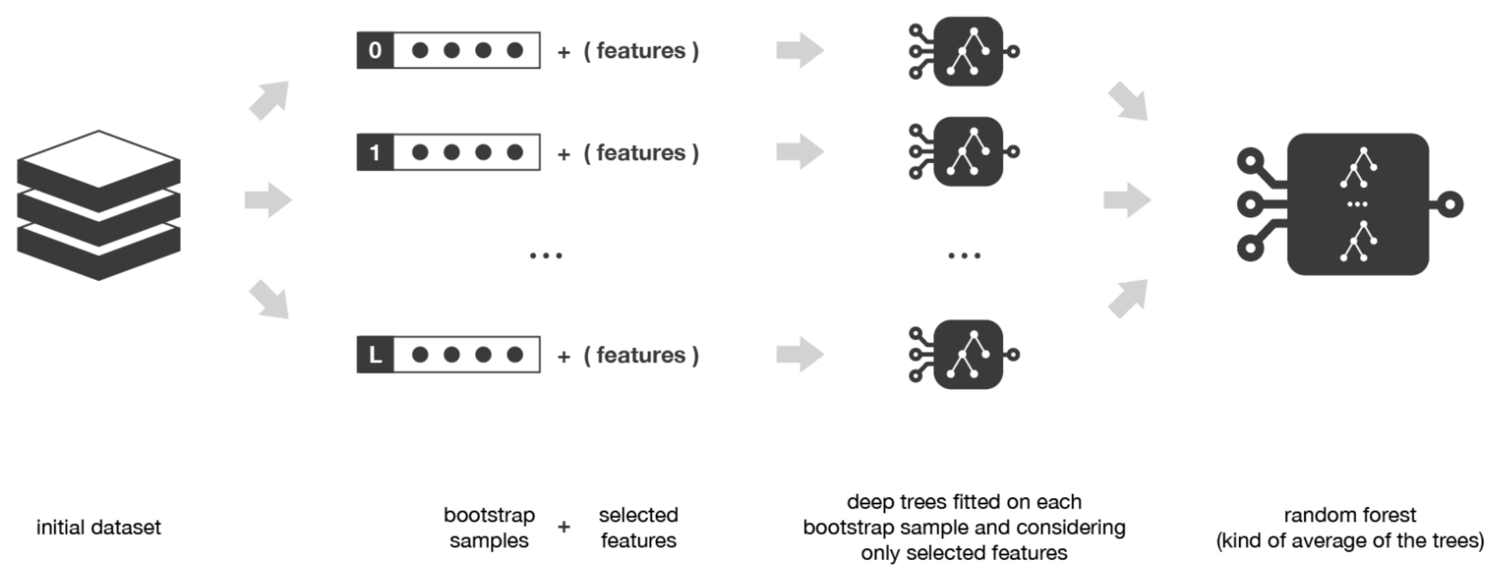
- Concepts



Training Bagging Ensemble

Multiple sets of bootstrap samples are almost equivalent to the population

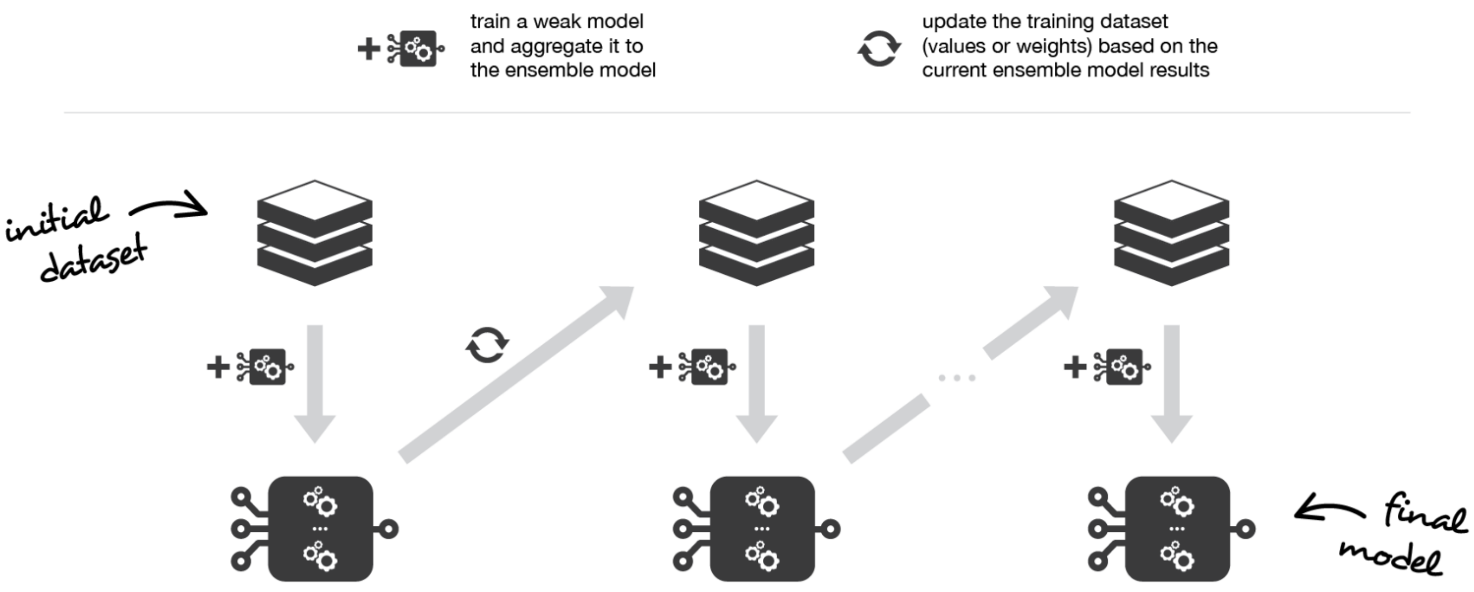
- Concepts



Training Bagging Ensemble

Bootstrap samples is often used for training bagging ensemble model

- Concepts

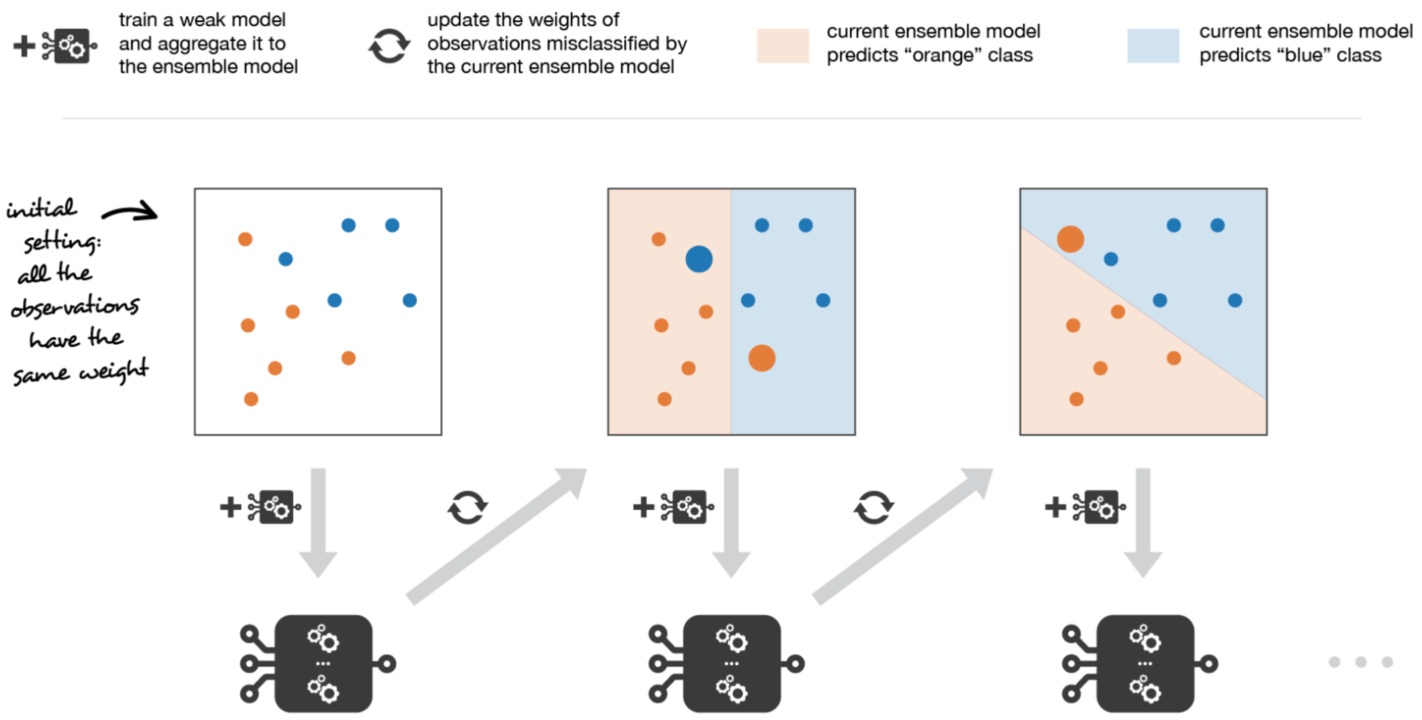


Boosting consists in, iteratively, fitting a weak learner, aggregate it to the ensemble model and "update" the training dataset to better take into account the strengths and weakness of the current ensemble model when fitting the next base model.

Training Boosting Ensemble

Focus on reducing bias

- Adaptive Boosting (AdaBoost)



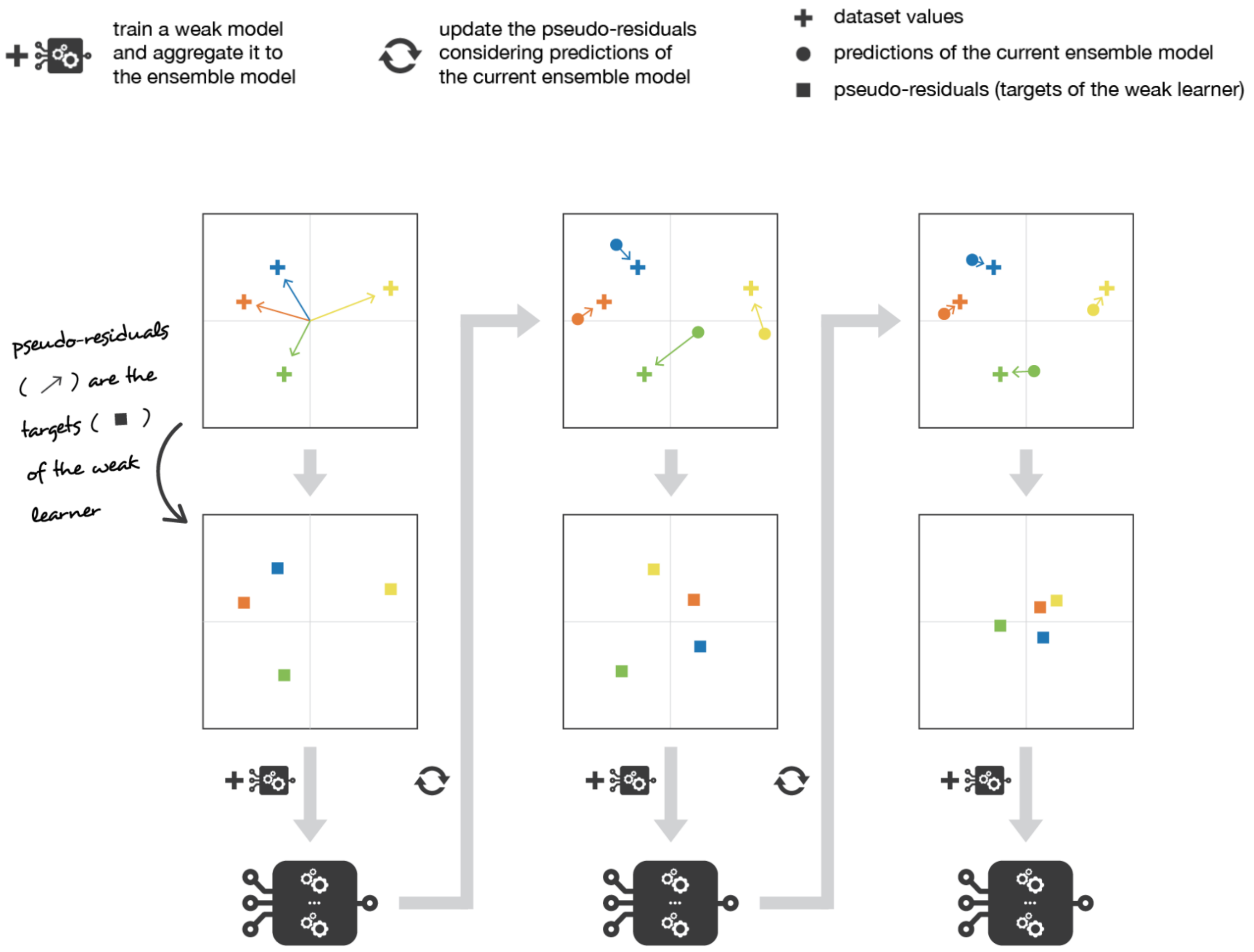
Training Boosting Ensemble

Adaboost updates weights of the observations at each iteration

Weights of well classified observations decrease relatively to weights of misclassified observations

Models that perform better have higher weights in the final ensemble model

- Gradient Boosting

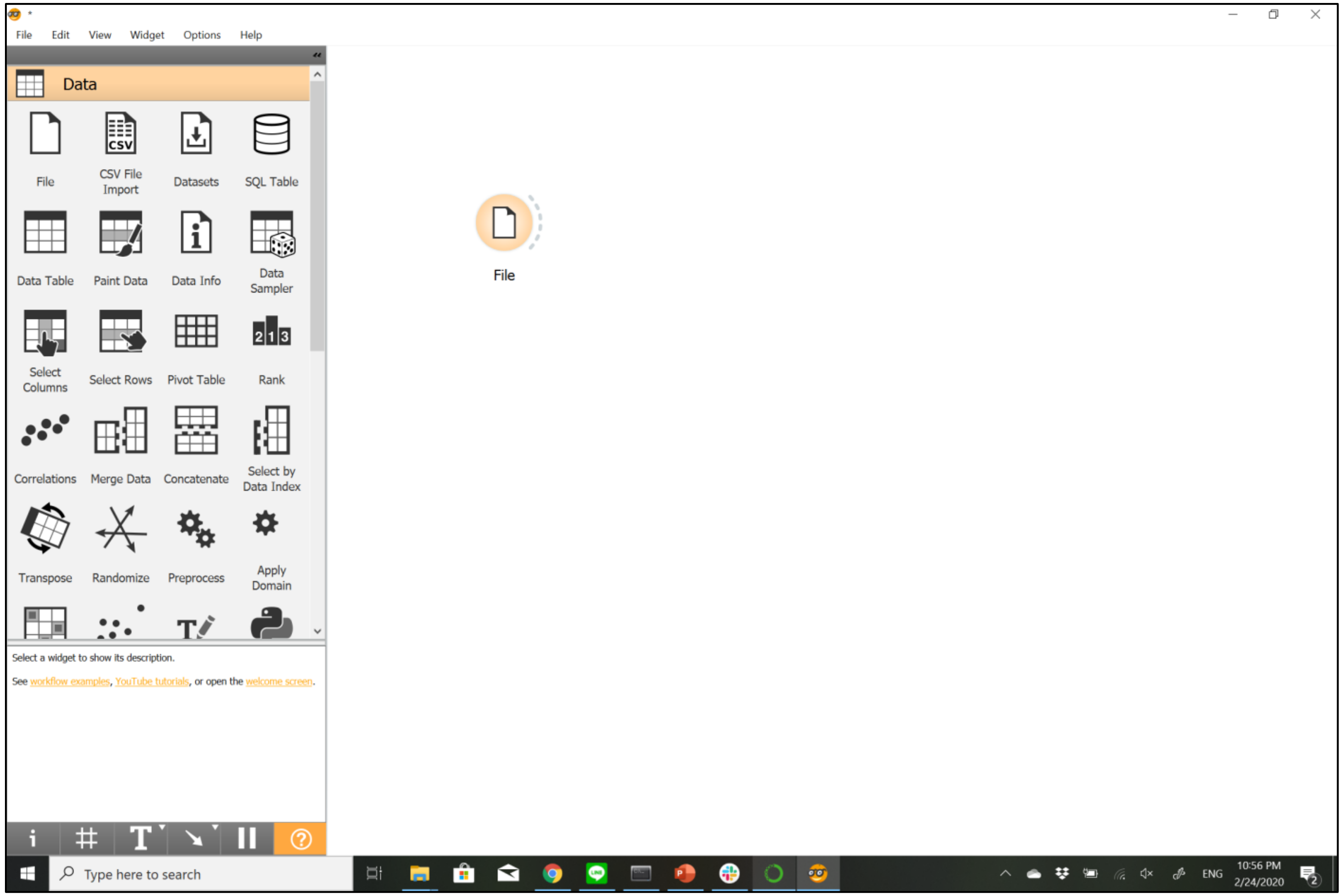


Training Boosting Ensemble

Gradient boosting updates values of the observations at each iteration

Weak learners are trained to fit the pseudo-residuals that indicate in which direction to correct the current ensemble model predictions to lower the error

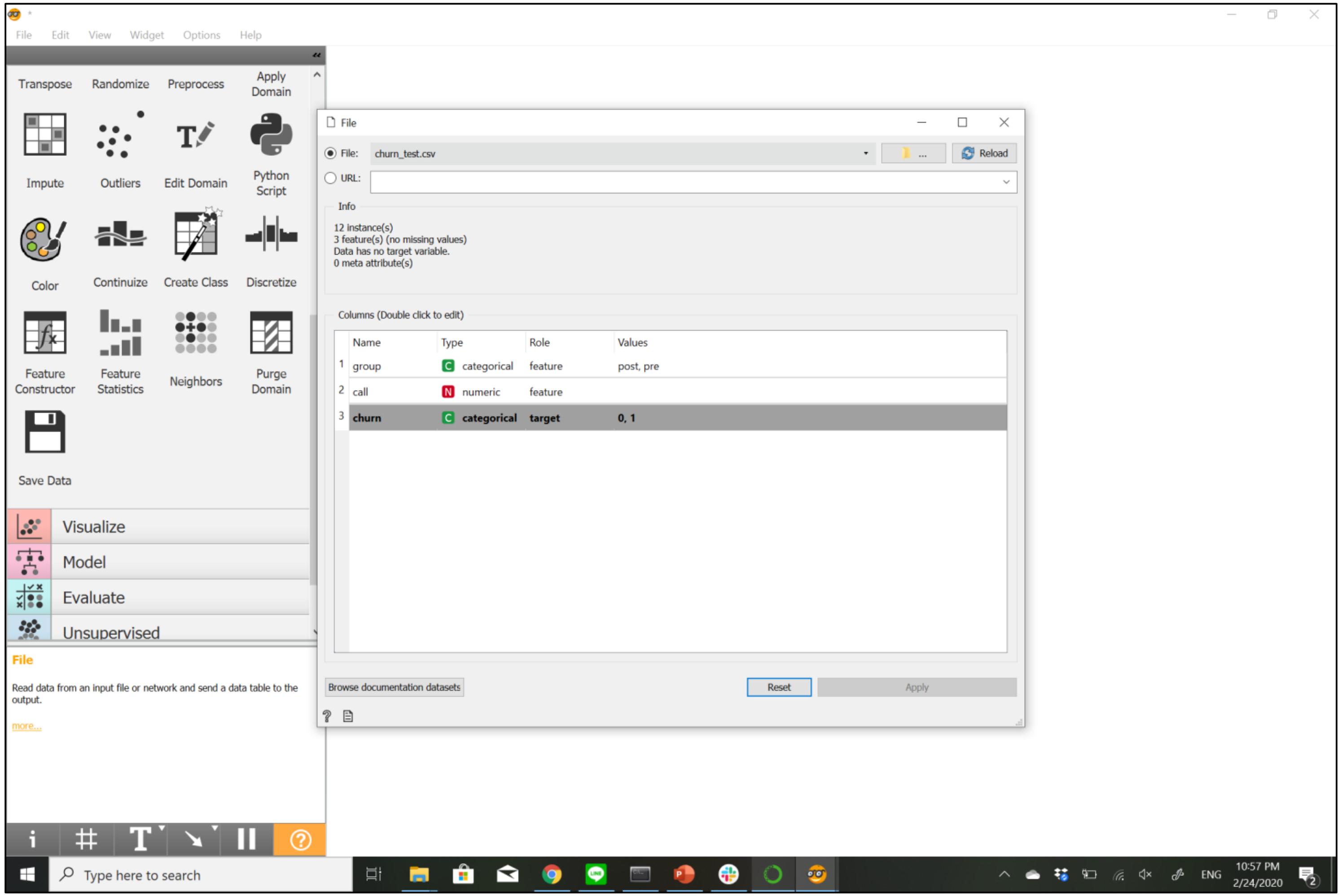
- Import data



Decision Tree in Orange

Build your model in 10 seconds

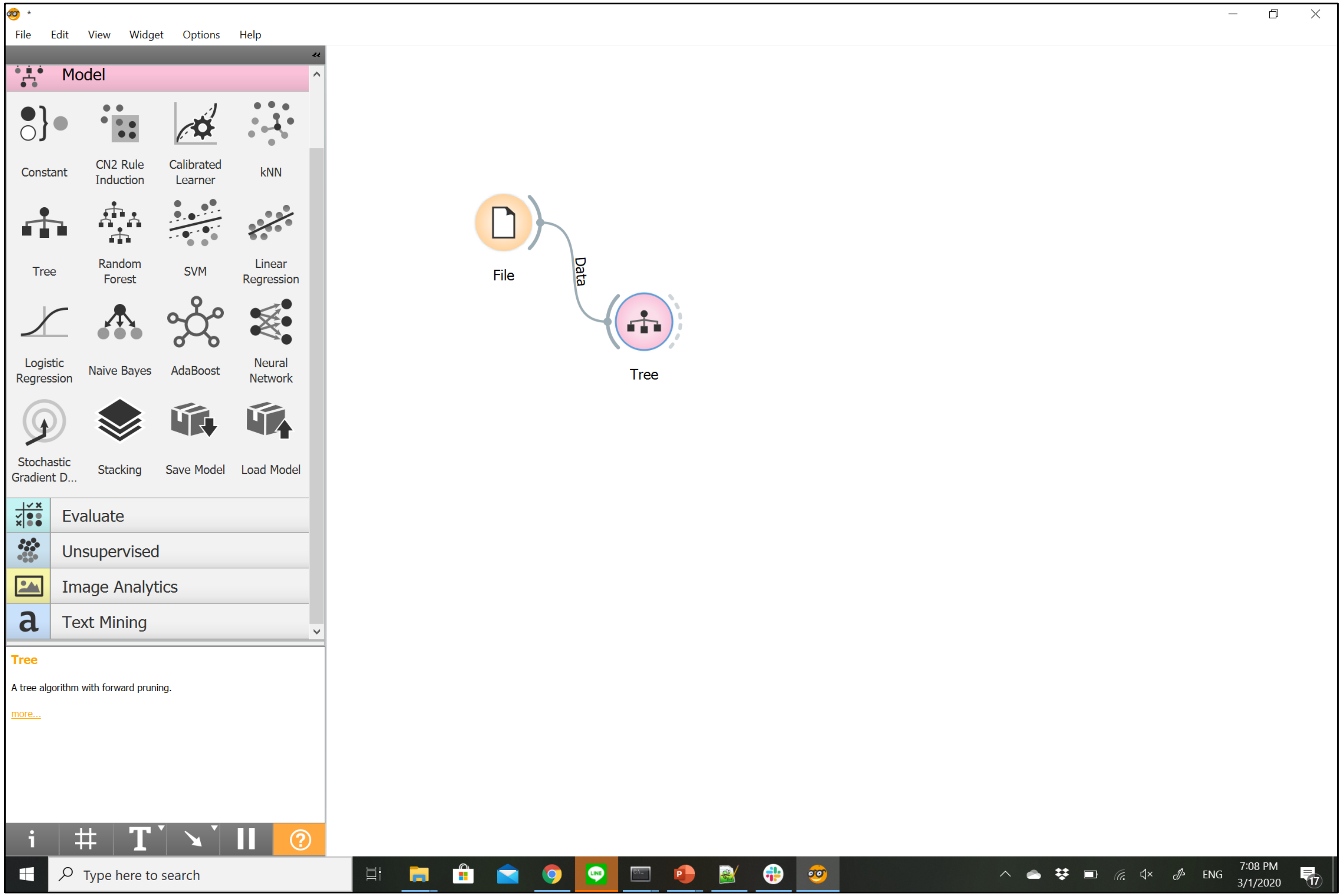
- Identify features and target



Decision Tree in Orange

Build your model in 10 seconds

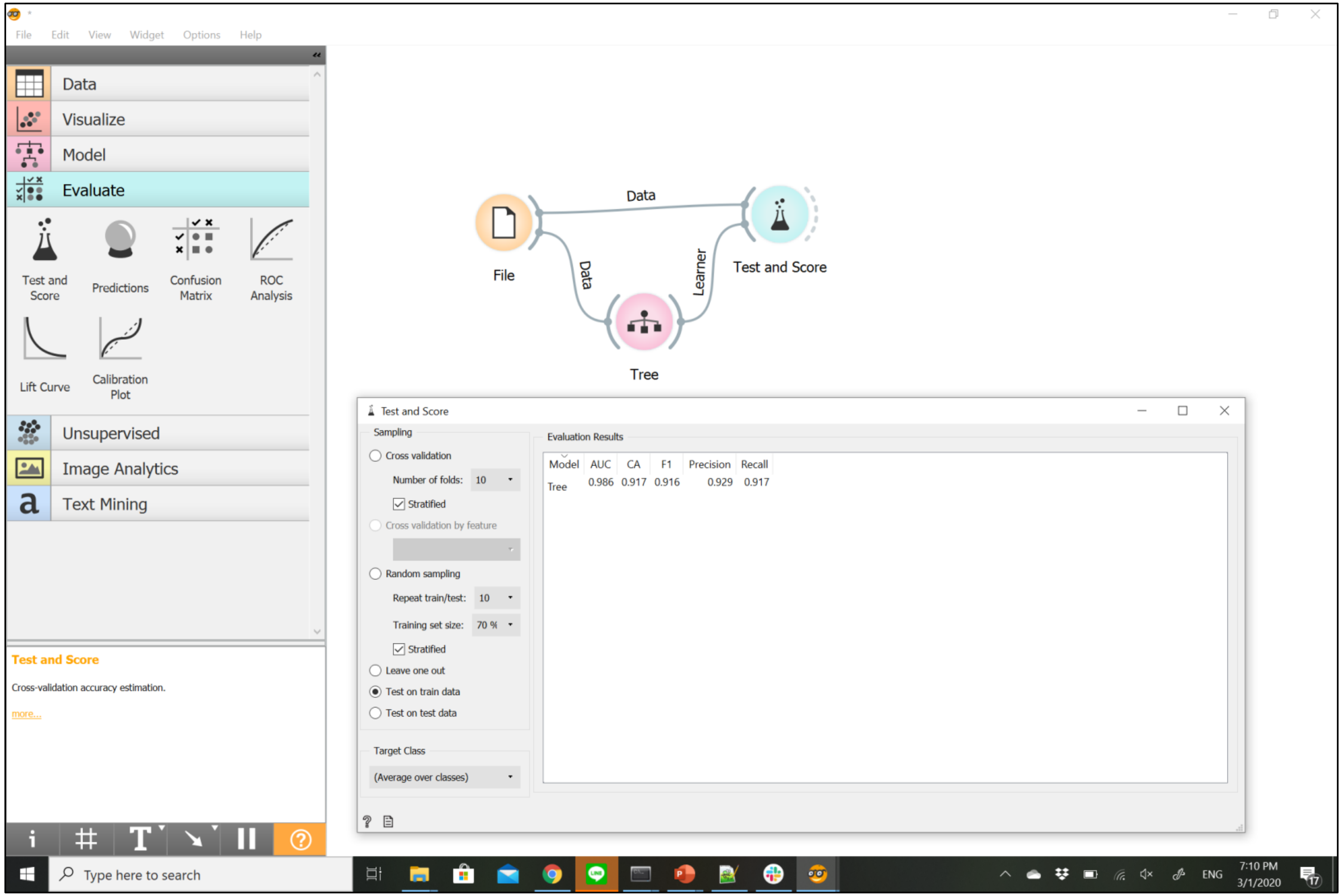
- Add model



Decision Tree in Orange

Build your model in 10 seconds

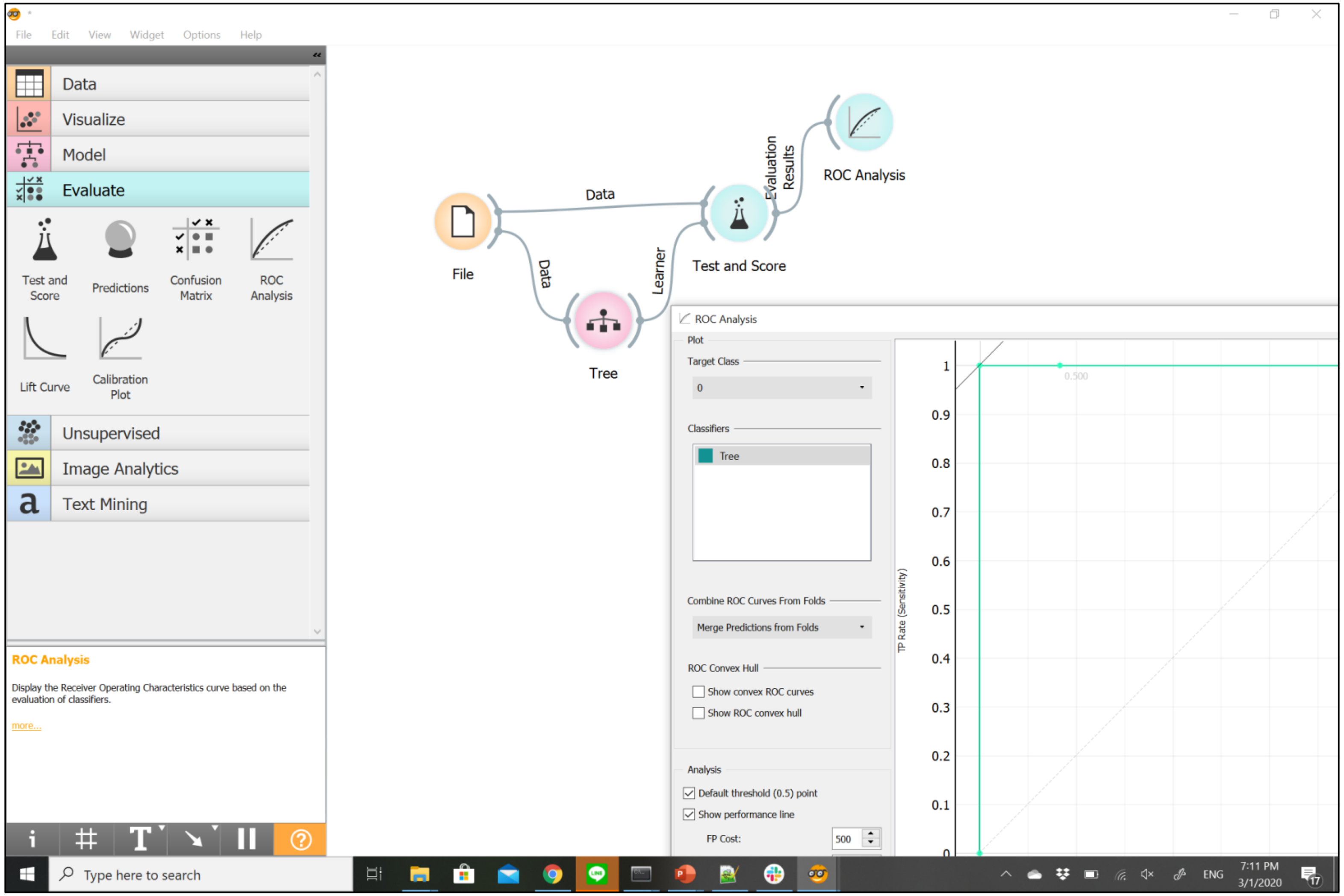
- Evaluate your model



Decision Tree in Orange

Build your model in 10 seconds

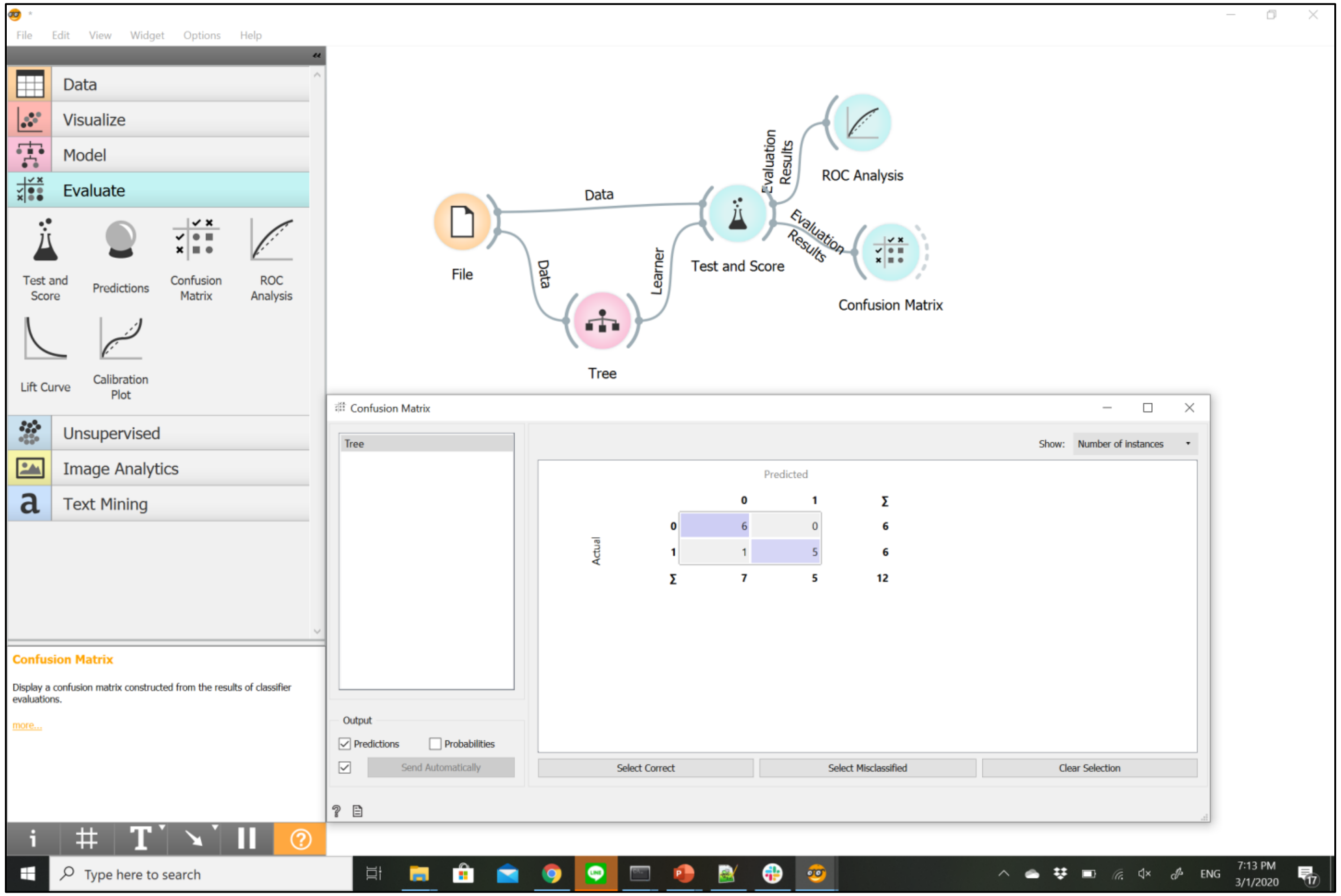
- ROC Plot



Decision Tree in Orange

Build your model in 10 seconds

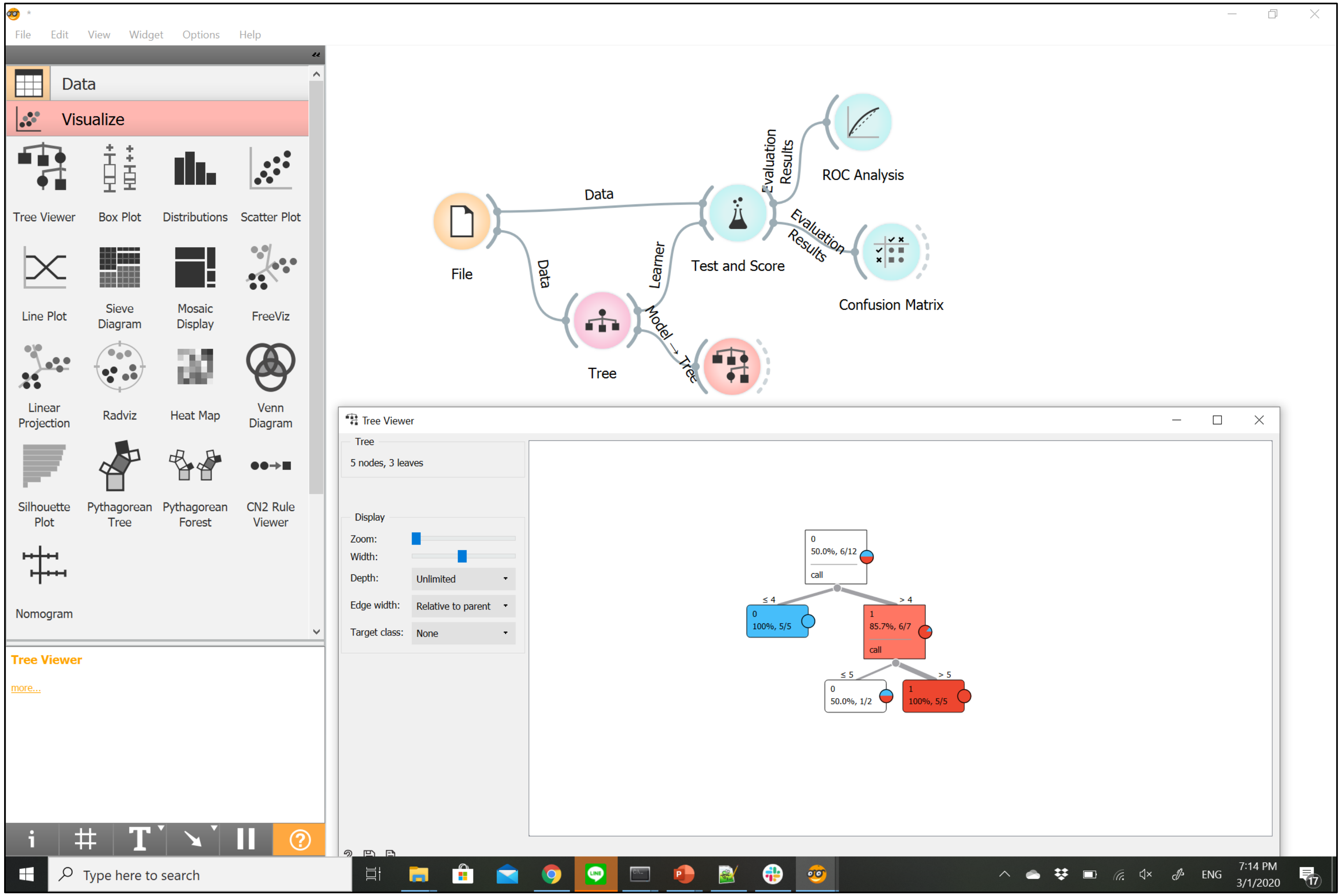
- Confusion Matrix



Decision Tree in Orange

Build your model in 10 seconds

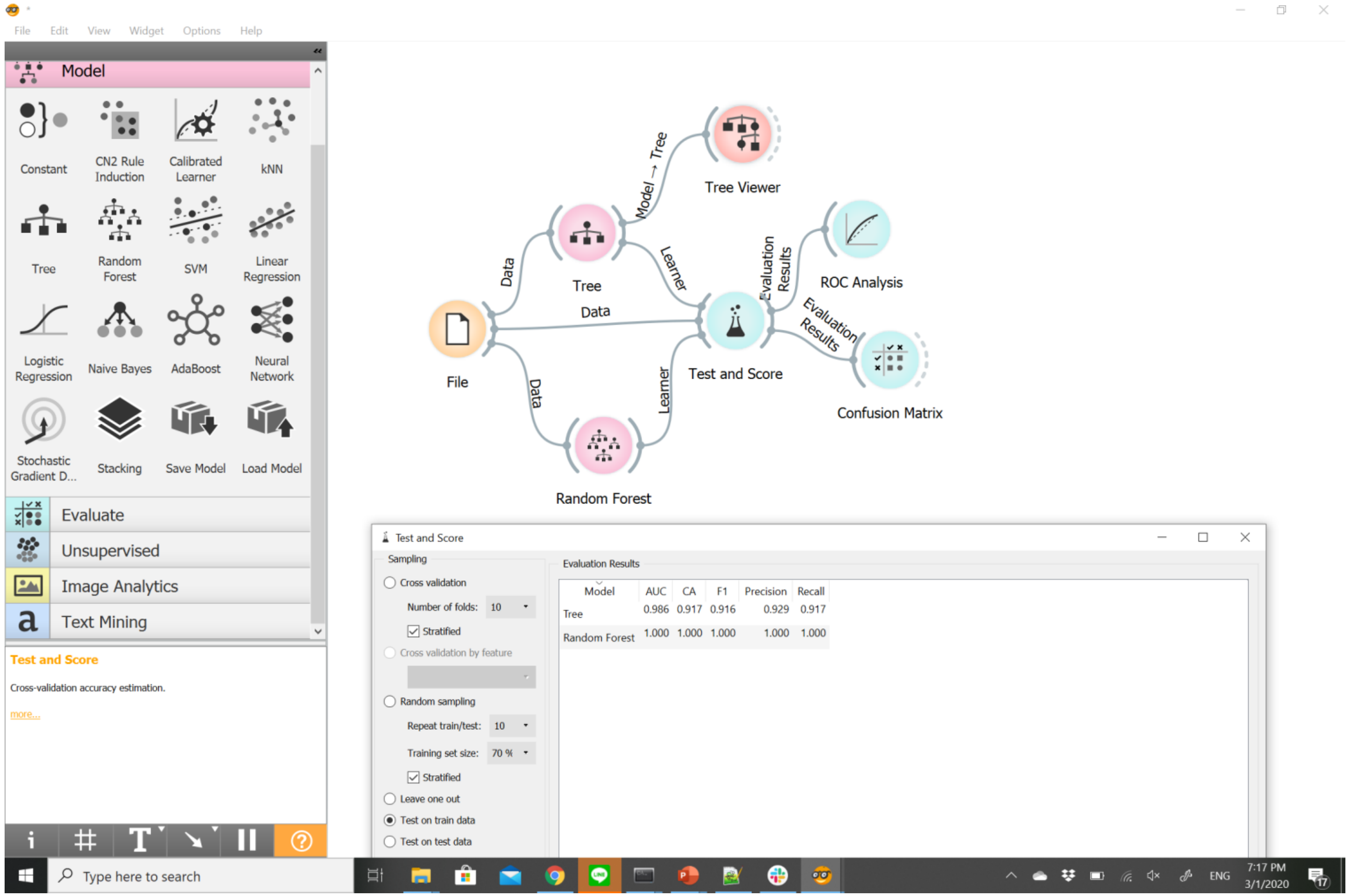
◦ View your tree



Decision Tree in Orange

Build your model in 10 seconds

◦ Add the random forest in to the canvas

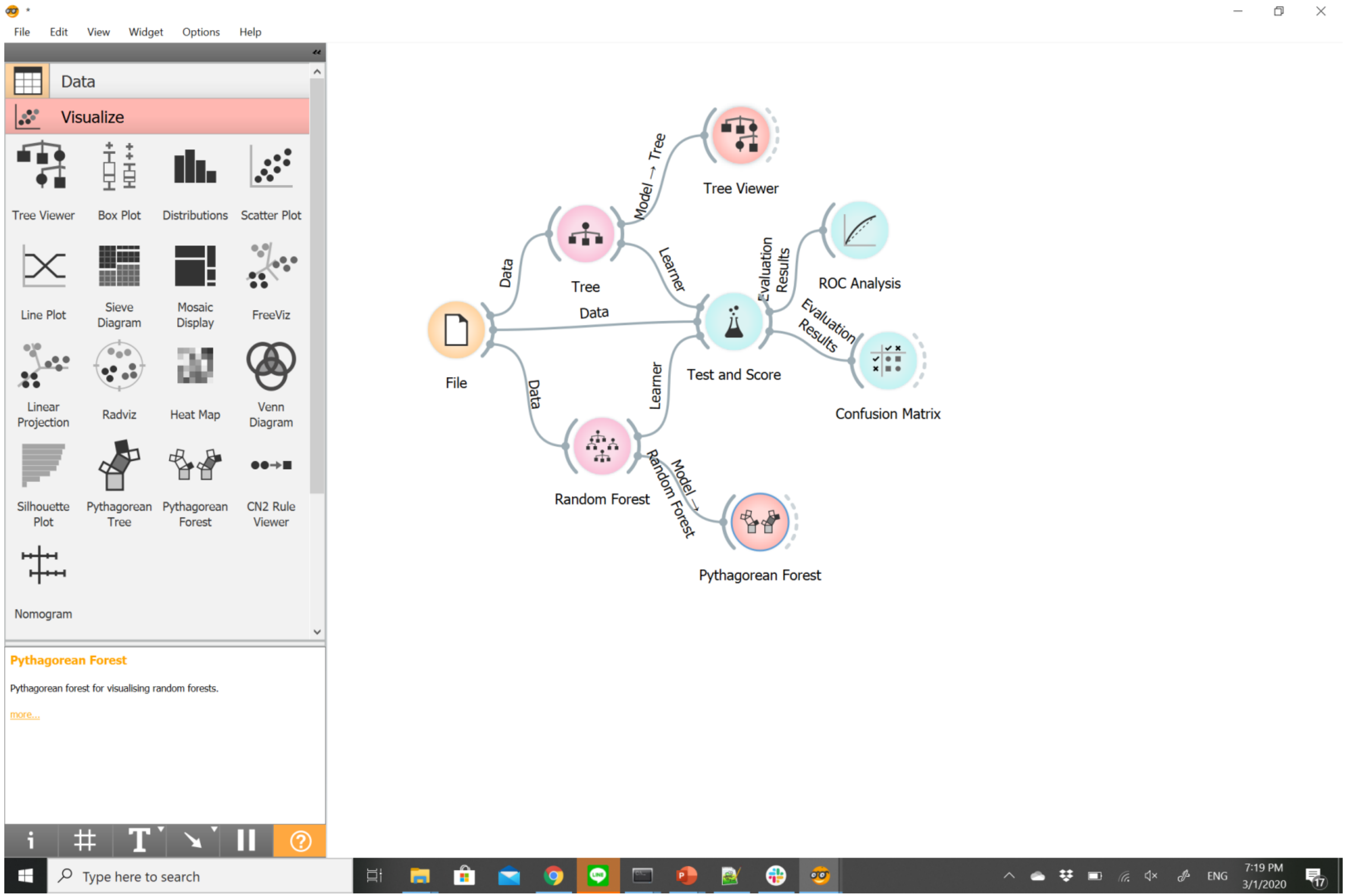


Random Forest in Orange

Build your model in 10 seconds

You can train multiple models together

- Visualize your random forest with the Pythagorean forest



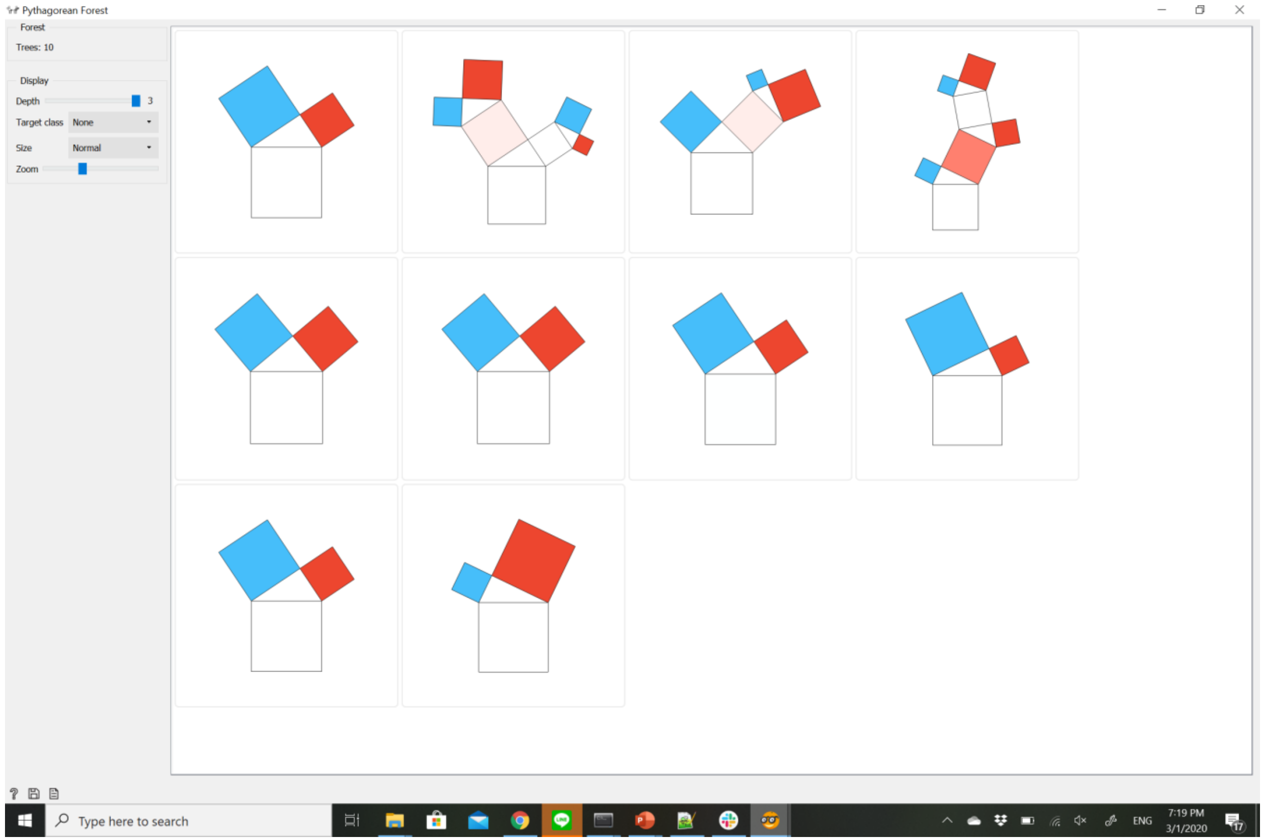
Random Forest in Orange

Build your model in 10 seconds

Random Forest in Orange

Build your model in 10 seconds

- Visualize your random forest with the Pythagorean forest



- Build your model using Kaggle dataset

Exercise

Our sample data is very small for demonstration purpose

Now it is the time to work with the dataset churn prediction of telecom in Kaggle Competition