Ensemble Methods: Exercise

Dr. José Ramón Iglesias
DSP-ASIC BUILDER GROUP
Director Semillero TRIAC
Ingenieria Electronica
Universidad Popular del Cesar

Exercise 1

Random Forests

- *Breiman first invented the theory of Bagging before he refined his idea to Random Forests. Describe the two main differences between Random Forests and Bagging using Decision Trees as Models*

- Only binary trees are built
- Only *m* out of *p* attributes are used to select the best split

Exercise 2
AdaBoost

- *Assume that we are trying to predict a binary outcome $y = y_1, y_2, \ldots, y_n$. Each C_i can take a value +1 or -1, denoted by "+" and "-", respectively. An AdaBoost method has been applied to this data, and have been trained to step t. The predicted outcomes $h_t(x_i)$, associated with input features $x_i$xi, are also listed below, as well as the corresponding weights $D_t(i)$.*

y	+	-	-	+	-	-	+
$h_t(x_i)$	+	-	+	+	-	-	-
$D_t(i)$	0.21	0.05	0.05	0.28	0.1	0.1	0.21

AdaBoost

1. *Calculate the error* $e_t = \sum_{i=1}^{n} D_t(i) \cdot \delta(y_i \neq h_t(x_i))$
2. *Calculate the learner weight* $\alpha_t = \frac{1}{2} ln \frac{1-e_t}{e_t}$
3. *Calculate* $exp\left(-\alpha_t \cdot y_t \cdot h_t(x_i)\right)$ *for each data point.*
4. *Calculate the updated weights* $D_{t+1}(i) = \frac{D_t(i) \cdot exp\left(-\alpha_t \cdot y_t \cdot h_t(x_i)\right)}{\sum_{i=1}^{n} D_t(i) \cdot exp\left(-\alpha_t \cdot y_t \cdot h_t(x_i)\right)}$ *for each data point.*

1. *Calculate the error* $e_t = \sum_{i=1}^{n} D_t(i) \cdot \delta(y_i \neq h_t(x_i))$

– There are two mis-classified data points, and those are the ones included in the error calculation

y	+	-	-	+	-	-	+
$h_t(x_i)$	+	-	+	+	-	-	-
$D_t(i)$	0.21	0.05	0.05	0.28	0.1	0.1	0.21

– Thus the error $e_t = \sum_{i=1}^{n} D_t(i) \cdot \delta(y_i \neq h_t(x_i)) = 0.05 + 0.21\ 0\ 0.26$

2. *Calculate the learner weight* $\alpha_t = \frac{1}{2} ln \frac{1-e_t}{e_t}$

– Based on the solution for the previous problem,

$$\alpha_t = \frac{1}{2} ln \frac{1-e_t}{e_t} = 0.52$$

3. *Calculate $exp(-\alpha_t \cdot y_t \cdot h_t(x_i))$ for each data point*

- With α_t from the previous step, we have $\exp(-\alpha_t) = 0.59$ for correctly classified data points, and $\exp(\alpha_t) = 1.69$ for incorrectly classified data points.

y	+	-	-	+	-	-	+
$h_t(x_i)$	+	-	+	+	-	-	-
$D_t(i)$	0.21	0.05	0.05	0.28	0.1	0.1	0.21
$exp(-\alpha_t \cdot y_t \cdot h_t(x_i))$	**0.59**	**0.59**	**1.69**	**0.59**	**0.59**	**0.59**	**1.69**

4. *Calculate the updated weights* $D_{t+1}(i) = \frac{D_t(i) \cdot exp\left(-\alpha_t \cdot y_t \cdot h_t(x_i)\right)}{\sum_{i=1}^{n} D_t(i) \cdot exp\left(-\alpha_t \cdot y_t \cdot h_t(x_i)\right)}$ *for each data point.*

– First, we calculate $D_t(i) \cdot exp\left(-\alpha_t \cdot y_t \cdot h_t(x_i)\right)$ for each data point. Then this is normalized with the sum (0.88)

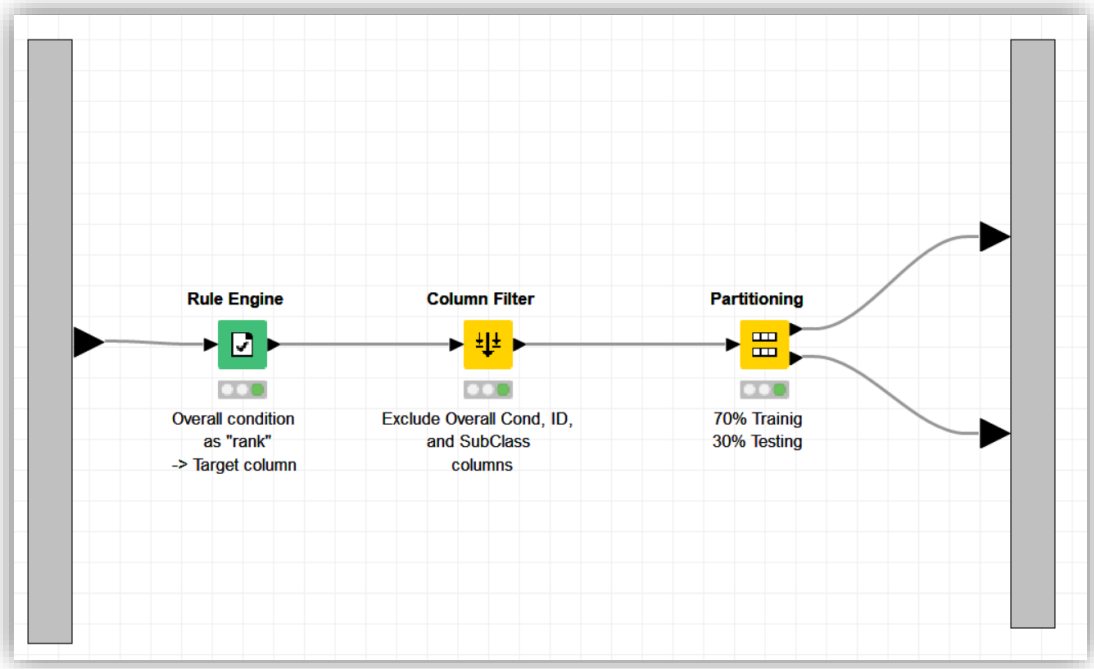
y	+	-	-	+	-	-	+
$h_t(x_i)$	+	-	+	+	-	-	-
$D_t(i)$	0.21	0.05	0.05	0.28	0.1	0.1	0.21
$exp\left(-\alpha_t \cdot y_t \cdot h_t(x_i)\right)$	0.59	0.59	1.69	0.59	0.59	0.59	1.69
$D_t(i) \cdot exp\left(-\alpha_t \cdot y_t \cdot h_t(x_i)\right)$	0.12	0.03	0.08	0.17	0.06	0.06	0.35
$D_{t+1}(i)$	**0.14**	**0.03**	**0.10**	**0.19**	**0.07**	**0.07**	**0.40**

Exercise 3
Practice with KNIME

Random Forest & Gradient Boosted trees

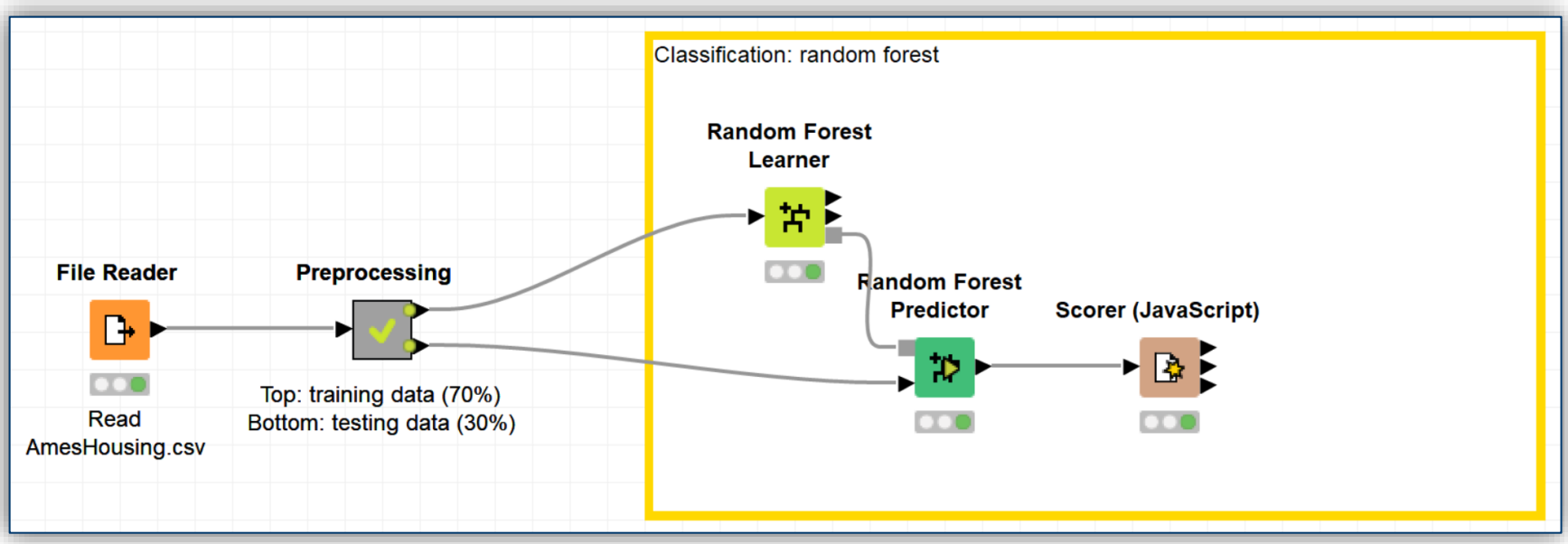
- The dataset we use in this exercise describes the sale of individual residential properties in Ames, Iowa from 2006 to 2010. One of the columns is the overall condition ranking, with values between 1 and 10.
- The goal of this exercise is to train a binary classification model, which can predict whether the overall condition is high or low. To do so, the workflow reads the data set and creates the class column based on overall condition ranking, which is called rank and has the values low if the overall condition is smaller or equal to 5, otherwise high.

- **Preprocessing:**
- Create binary target value “rank” (“High” or “Low”)
- Eliminates some features
- Partition the dataset



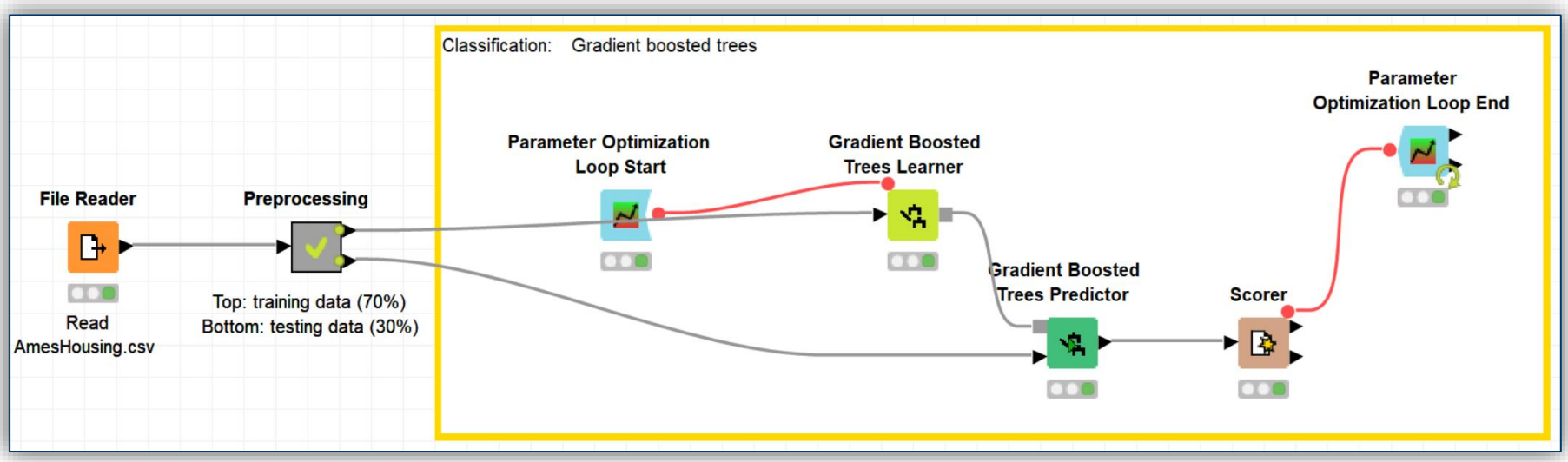
1. Train a random forest model (Random Forest Learner) to predict the overall condition of a house (high/low), with the following settings:
   - Limit tree depth to 15
   - Limit the minimum node size to 10
   - The number of trees (i.e. models) to 100
2. Examine the output table “Attribute Statistics” generated by the learner node. It lists which features were chosen for the first 3 splits in different tree models. Features chosen first are likely informative in classification. Which of the features in this dataset are very informative? Justify your answer.
3. Use the trained model to predict (Random Forest Predictor) the rank of the houses in the test set, and evaluate the accuracy of the decision tree model (Scorer JavaScript). What is the accuracy of the model?

Random Forest



4. Train a gradient boosted trees model to predict the overall condition of a house (high/low), with the following setting:
   - Limit the tree depth to 4
   - The number of models to 100
5. Use the trained model to predict the rank of the houses in the test set, and evaluate the accuracy of the decision tree model. What is the accuracy of the model?
6. (Optional) Use a parameter optimization loop to find the optimum tree depth to maximize the accuracy (use Scorer node, not Scorer JavaScript). Examine the tree depth of 1 to 10, with the step size of 1. Which tree depth produces the optimal result?

Gradient Boosted trees



Thank you