Simple Model Using a Decision Tree to Predict Breast Cancer Class

Wilmer Vidal Uruchi Ticona[1]

[1]Universitat Politécnica de Catalunya

March 29, 2019

Abstract

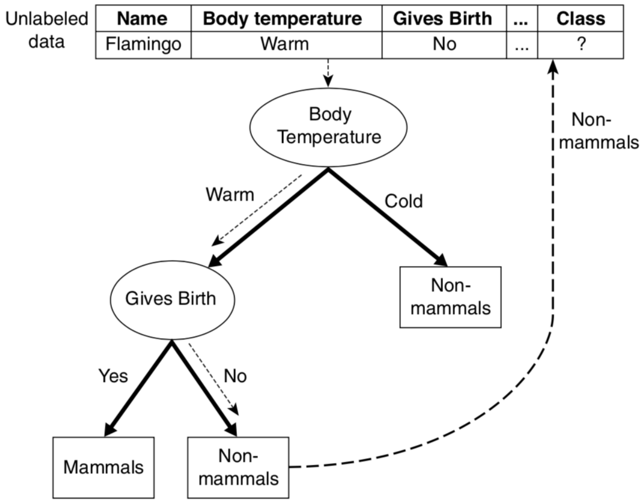
The following is a description of the process of construction of a Decision Tree Model for prediction of breast cancer class. Further experimentation has been performed and documented to test some of the configuration settings provided by the KNIME component **Decision Tree Learner**. The breast cancer database used in this project was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. The project is mostly based on information from "Introduction to Data Mining" by Tan, Seteinback, and Kumar. Since this is an introductory project into the subject, some definitions are provided; however, it is advised to follow the references for detailed information.

1 Preliminary Definitions

1.1 Decision Tree

A decision tree is a classification technique to build classification models from an input dataset. This technique employs a **learning algorithm** to build a model that best fits the relationship between the attribute set and the class label of the dataset [1]. A key objective of the learning algorithm is to build a model with good generalization capabilities, meaning that the model should (with acceptable accuracy) predict the class label from a set of attributes, from previously unknown data.

Suppose we have a set of attributes $A = a_1, a_2, a_3, ...a_k$, where a_k is the class label, i.e. the class we want to predict. The tree will build a decision model in a way that when the model tries to predict the class label a_k, it will go through a sequence of questions based on the attributes a_1 to a_{k-1}, reducing the selection space after every decision, with the objective of reaching a conclusion about a_k.



Name	Body temperature	Gives Birth	...	Class
Flamingo	Warm	No	...	?

Figure 1: Example of a decision tree[1].

There are three types of nodes in this tree:

- **Root Node**: Starting node
- **Internal Node**: Represents a question
- **Leaf or Terminal Node**: Represents an answer

To select the attributes for the *Internal Nodes*, the algorithm must have some way to measure how the attributes impact the model, for this purpose we are taking into consideration these indicators:

- **Gini Index** : Based on the degree of impurity of the child nodes, where impurity is high if after a split we get an equal number of records of each class, and it is low if it is highly skewed.
- **Gain Ratio** : In this strategy we include the number of outcomes produced by the split criterion in a way that if an attribute produces a large number of splits, its gain ratio will be lower. Higher gain ratios are favored.

A way to avoid overfitting (when the model fits too well the *training data*[1], but not the *testing data*[2]) is to apply pruning. In our final experiment we will apply **post-pruning**: In this approach, the decision tree is initially grown to its maximum size. This is followed by a tree-pruning step, which proceeds to trim the fully grown tree in a bottom-up fashion [1].

[1] Data used for the learning algorithm

[2] Data used for the prediction test

1.2 Breast Cancer

Breast cancer is cancer that forms in the cells of the breasts.

After skin cancer, breast cancer is the most common cancer diagnosed in women in the United States. Breast cancer can occur in both men and women, but it's far more common in women.

Substantial support for breast cancer awareness and research funding has helped create advances in the diagnosis and treatment of breast cancer. Breast cancer survival rates have increased, and the number of deaths associated with this disease is steadily declining, largely due to factors such as earlier detection, a new personalized approach to treatment and a better understanding of the disease [2].

1.3 KNIME

At KNIME®, we build software for fast, easy and intuitive access to advanced data science, helping organizations drive innovation.

Our KNIME Analytics Platform is the leading open solution for data-driven innovation, designed for discovering the potential hidden in data, mining for fresh insights, or predicting new futures. Organizations can take their collaboration, productivity and performance to the next level with a robust range of commercial extensions to our open source platform.

For over a decade, a thriving community of data scientists in over 60 countries has been working with our platform on every kind of data: from numbers to images, molecules to humans, signals to complex networks, and simple statistics to big data analytics.

KNIME's headquarters are based in Zurich, with additional offices in Konstanz, Berlin, and Austin. We're open for innovation®, so visit us at KNIME [3].

2 The Data

This breast cancer databases was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

1. Title: Wisconsin Breast Cancer Database (January 8, 1991)

2. Number of Instances: 699 (as of 15 July 1992)

3. Number of Attributes: 10 plus the class attribute

4. Class distribution: Benign 458 (65.5%), Malignant 241 (34.5%)

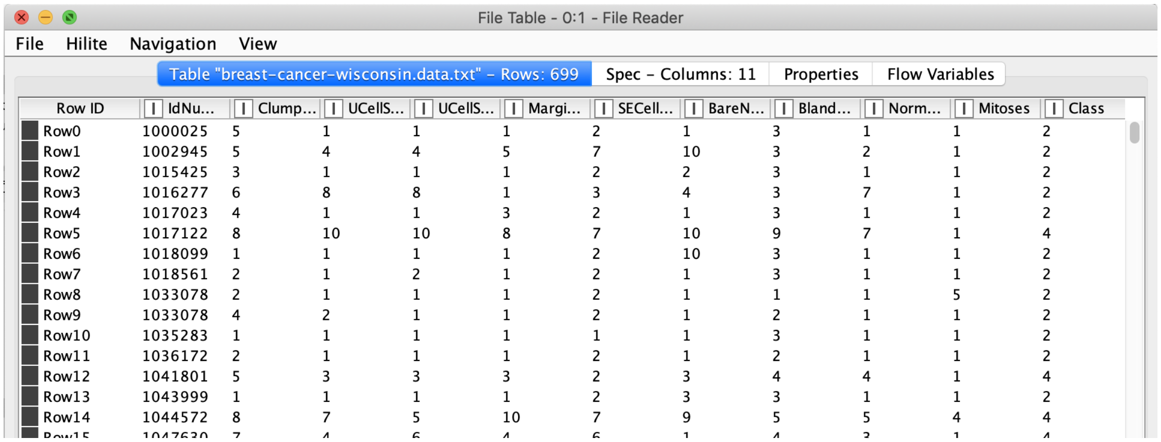
5. Attribute Information: (class attribute has been moved to last column). See **Table 1**.

Table 1: Attribute Information

Attribute	Domain	Brief Description
Sample Code Number	id number	Id of patient.
Clump Thickness	1 - 10	This is used to assess if cells are mono-layered or multi-layered. Benign cells tend to be grouped in mono-layers, while cancerous cells are often grouped in multi-layer.
Uniformity of Cell Size	1 - 10	It is used to evaluate the consistency in the size of cells in the sample. Cancer cells tend to vary in size. That is why this parameter is very valuable in determining whether the cells are cancerous or not.
Uniformity of Cell Shape	1 - 10	It is used to estimate the equality of cell shapes and identifies marginal variances, because cancer cells tend to vary in shape.
Marginal Adhesion	1 - 10	Normal cells tend to stick together. Cancer cells tend to loose this ability. So loss of adhesion is a sign of malignancy.
Single Epithelial Cell Size	1 - 10	It is related to the uniformity. Epithelial cells that are significantly enlarged may be a malignant cell.
Bare Nuclei	1 - 10	This is a term used for nuclei that is not surrounded by cytoplasm. Those are typically seen in benign tumors.
Bland Chromatin	1 - 10	Describes a uniform "texture" of the nucleus seen in benign cells. In cancer cells, the chromatin tends to be coarser.
Normal Nucleoli	1 - 10	Nucleoli are small structures seen in the nucleus. In normal cells the nucleolus is usually very small if visible at all. In cancer cells the nucleoli become much more prominent, and sometimes there are more of them.
Mitoses	1 - 10	It is an estimate of the number of mitosis that has taken place. Larger the value, greater is the chance of malignancy.
Class	2 : benign 4 : malignant	

3 The Model

We import the data from the dataset using the proper KNIME component.



File Table - 0:1 - File Reader
File
Hilite
Navigation
View
Table "breast-cancer-wisconsin.data.txt" – Rows: 699
Spec – Columns: 11
Properties
Flow Variables

Row ID	IdNu...	Clump...	UCellS...	UCellS...	Margi...	SECell...	BareN...	Bland...	Norm...	Mitoses	Class
Row0	1000025	5	1	1	1	2	1	3	1	1	2
Row1	1002945	5	4	4	5	7	10	3	2	1	2
Row2	1015425	3	1	1	1	2	2	3	1	1	2
Row3	1016277	6	8	8	1	3	4	3	7	1	2
Row4	1017023	4	1	1	3	2	1	3	1	1	2
Row5	1017122	8	10	10	8	7	10	9	7	1	4
Row6	1018099	1	1	1	1	2	10	3	1	1	2
Row7	1018561	2	1	2	1	2	1	3	1	1	2
Row8	1033078	2	1	1	1	2	1	1	1	5	2
Row9	1033078	4	2	1	1	2	1	2	1	1	2
Row10	1035283	1	1	1	1	1	1	3	1	1	2
Row11	1036172	2	1	1	1	2	1	2	1	1	2
Row12	1041801	5	3	3	3	2	3	4	4	1	4
Row13	1043999	1	1	1	1	2	3	3	1	1	2
Row14	1044572	8	7	5	10	7	9	5	5	4	4
Row15	1047630	7	4	6	4	6	1	4	3	1	4

Figure 2: Data after import.

The **IdNumber** column is not relevant for the computation of the Decision Tree, so it can be discarded. After that, the **Class** column needs to be a categorical non-numerical column, so it can work with the Decision Tree Learner component; we apply the corresponding data transformation. The sequence of transformations can be done in KNIME using the following components.

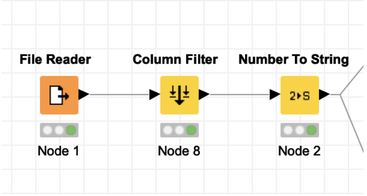


Figure 3: First we apply **Column Filter** to discard IdNumber, then we apply **Number To String** to transform Class from numerical to string.

There are 699 rows in the data set, and we have to find a way to divide them into two sets, one for the learning algorithm, and the other for the predictor. An intuitive way to do it is to divide the dataset into two almost equal subsets.

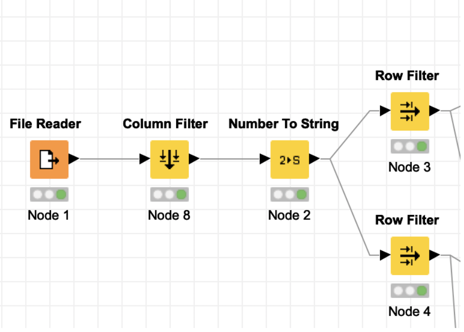


Figure 4: We apply two instances of **Row Filter** to divide the data set. The upper instance takes the first 350 rows for the Learner Algorithm (*training data*), and the bottom one takes the next 349 rows for the Predictor (*testing data*).

We add the **Decision Tree Learner** component. The *Class* column will be automatically selected as the label we want to predict, we leave everything else as default. In the next section we will play with some variables of the configuration.

According to the description of the component in KNIME, the **Decision Tree Learner** is based on the algorithm **C4.5**.

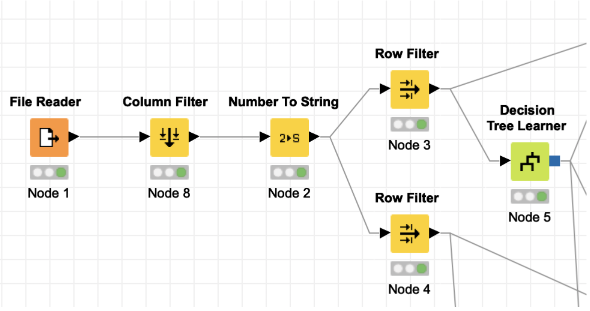


Figure 5: The **Decision Tree Learner** takes its information from the upper Row Filter.

The next step is to add the **Decision Tree Predictor** component. This component needs two inputs: the Decision Tree Learner from where it takes its model, and a data source. We are using two predictors, one for the *training data*, and the other for the *test data*, so we can calculate the corresponding accuracy percentages.

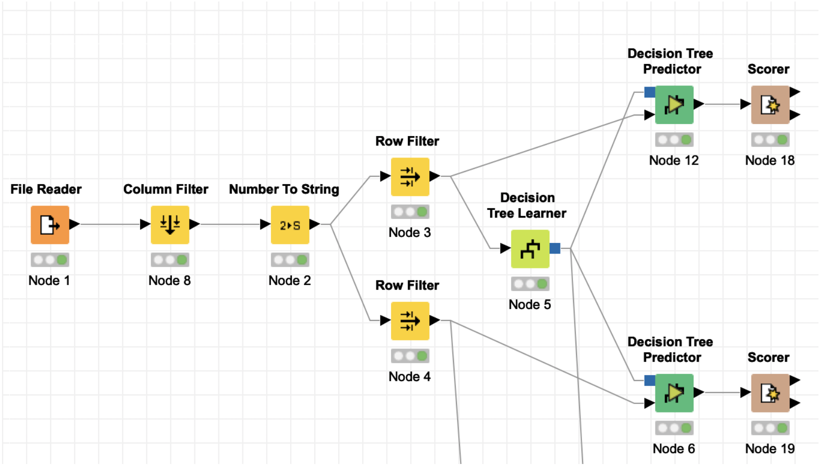


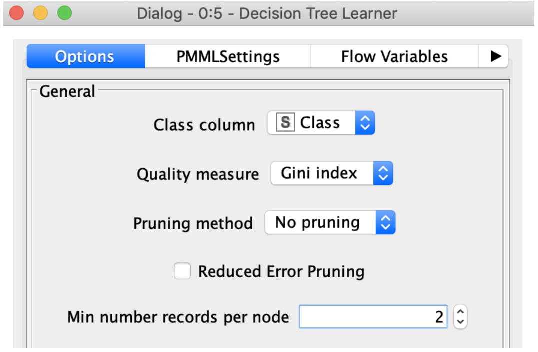
Figure 6: The **Decision Tree Predictors** report to a **Scorer**, which outputs the *Error Rate* and *Accuracy*. For our experimentation we are taking only *Accuracy* as a metric.

Finally, the model is ready for experimentation.

4 Experimentation

4.1 Definition of the Experimentation Space

We take into consideration the configuration available through the **Decision Tree Learner** component of KNIME.



Class column: S Class

Quality measure: Gini index

Pruning method: No pruning

☐ Reduced Error Pruning

Min number records per node: 2

Figure 7: The **Decision Tree Learner** configuration options that we are going to manipulate.

Where:

- **Class column**: The class the model is going to predict.

- **Quality Measure**: Defines the criteria the model is going to use to select attributes for the decision tree.
- **Pruning Method**: The MDL method is available, but we are not going to change it, except for the last experiment.
- **Reduced Error Pruning**: According to the definition given by KNIME, "It is a simple pruning method to cut the tree in a post-processing step". Left active as default.
- **Min number records per node**: According to the description given by KNIME, "To select the minimum number of records at least required in each node. If the number of records is smaller or equal to this number the tree is not grown any further. This corresponds to a stopping criteria (pre pruning)."

We are going to use the following configurations for experimentation:

Table 2: Experimental Configuration

# Experiment	Quality Measure	Min number records per node
1	Gini Index	2
2	Gini Index	4
3	Gain Ratio	2
4	Gain Ratio	4
4	Gini Index + MDL	2

4.2 Experiment 1: Gini Index + 2 Min number records per node

We obtain the following tree:

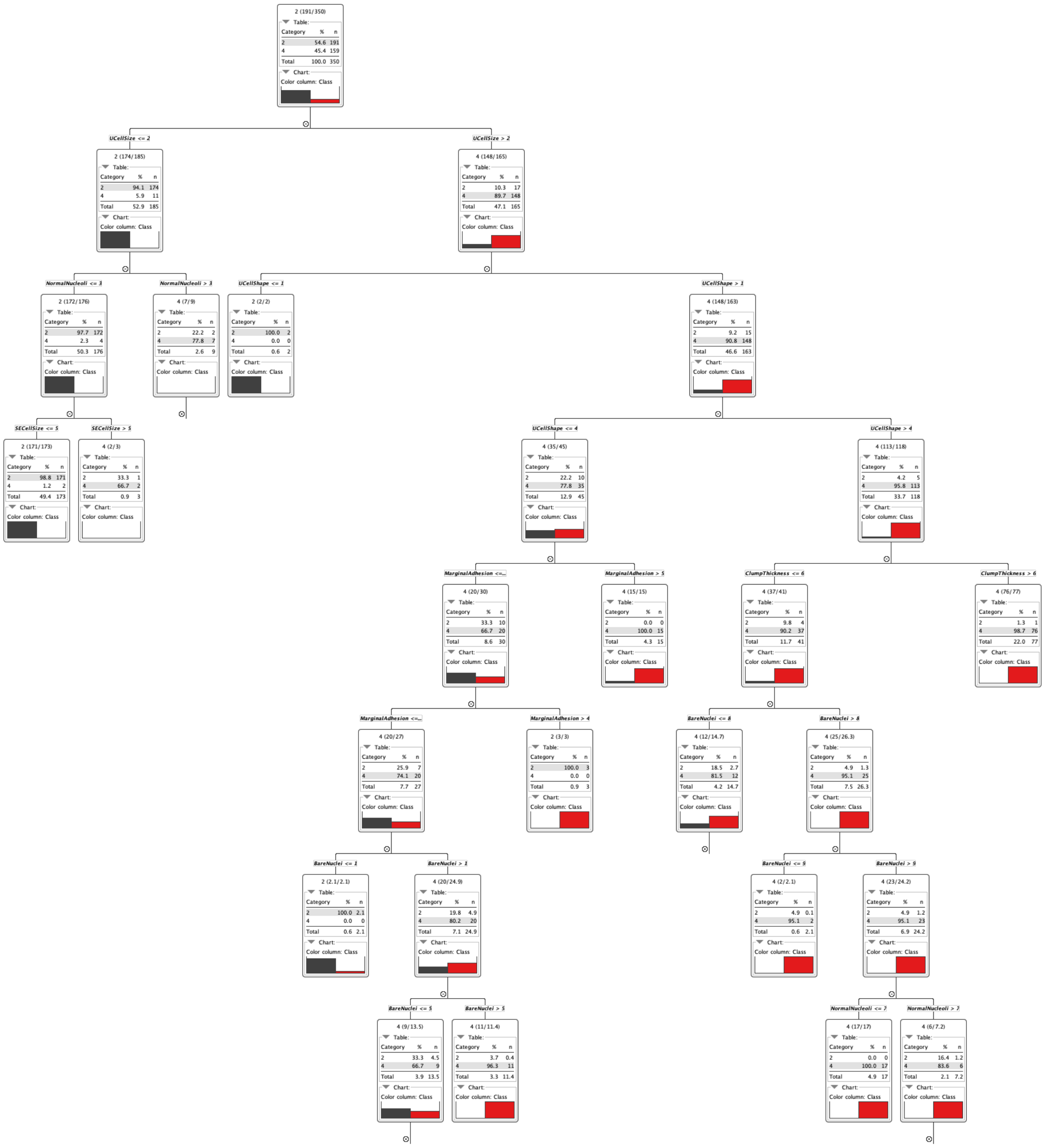


Figure 8: The tree resulting from the experiment. In gray, the rows of class 2 (benign); in red, the rows of class 4 (malignant).

After reviewing the Scorer components, we find:

- Training Accuracy: 96.857% (Correct classified: 339, Wrong classified: 11)
- Testing Accuracy: 96.562% (Correct classified: 337, Wrong classified: 12)

4.3 Experiment 2: Gini Index + 4 Min number records per node

We obtain the following tree:

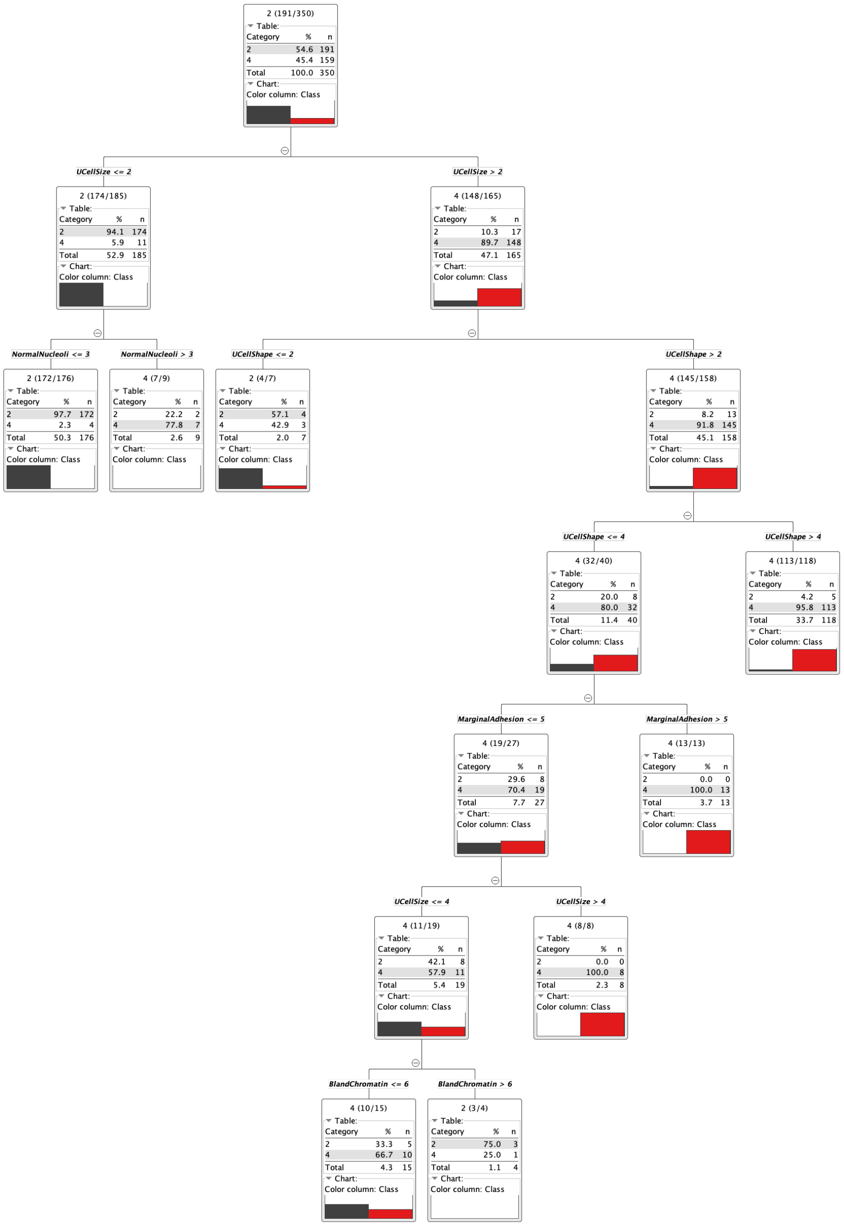


Figure 9: The tree resulting from the experiment. In gray, the rows of class 2 (benign); in red, the rows of class 4 (malignant).

After reviewing the Scorer components, we find:

- Training Accuracy: 94.286% (Correct classified: 330, Wrong classified: 20)
- Testing Accuracy: 96.275% (Correct classified: 336, Wrong classified: 13)

4.4 Experiment 3: Gain Ratio + 2 Min number records per node

We obtain the following tree:

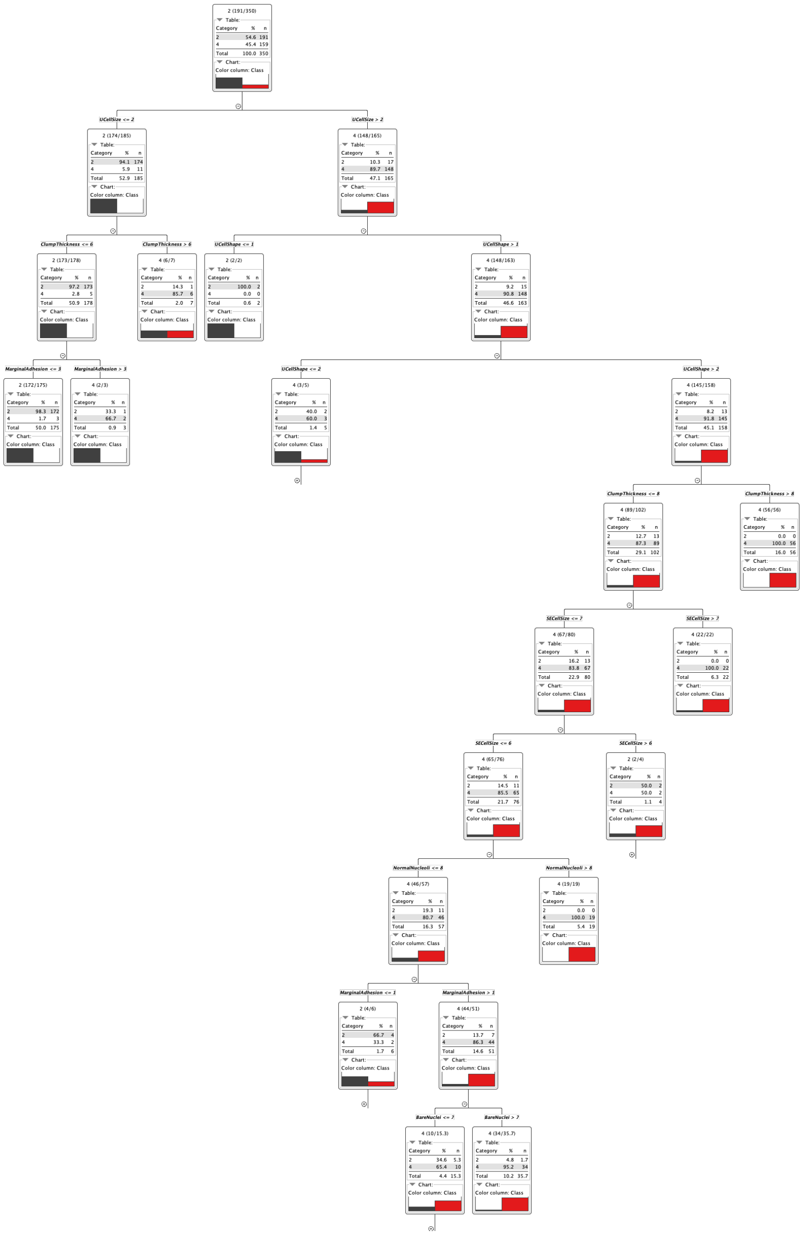


Figure 10: The tree resulting from the experiment. In gray, the rows of class 2 (benign); in red, the rows of class 4 (malignant).

After reviewing the Scorer components, we find:

- Training Accuracy: 97.429% (Correct classified: 341, Wrong classified: 9)
- Testing Accuracy: 94.842% (Correct classified: 331, Wrong classified: 18)

4.5 Experiment 4: Gain Ratio + 4 Min number records per node

We obtain the following tree:

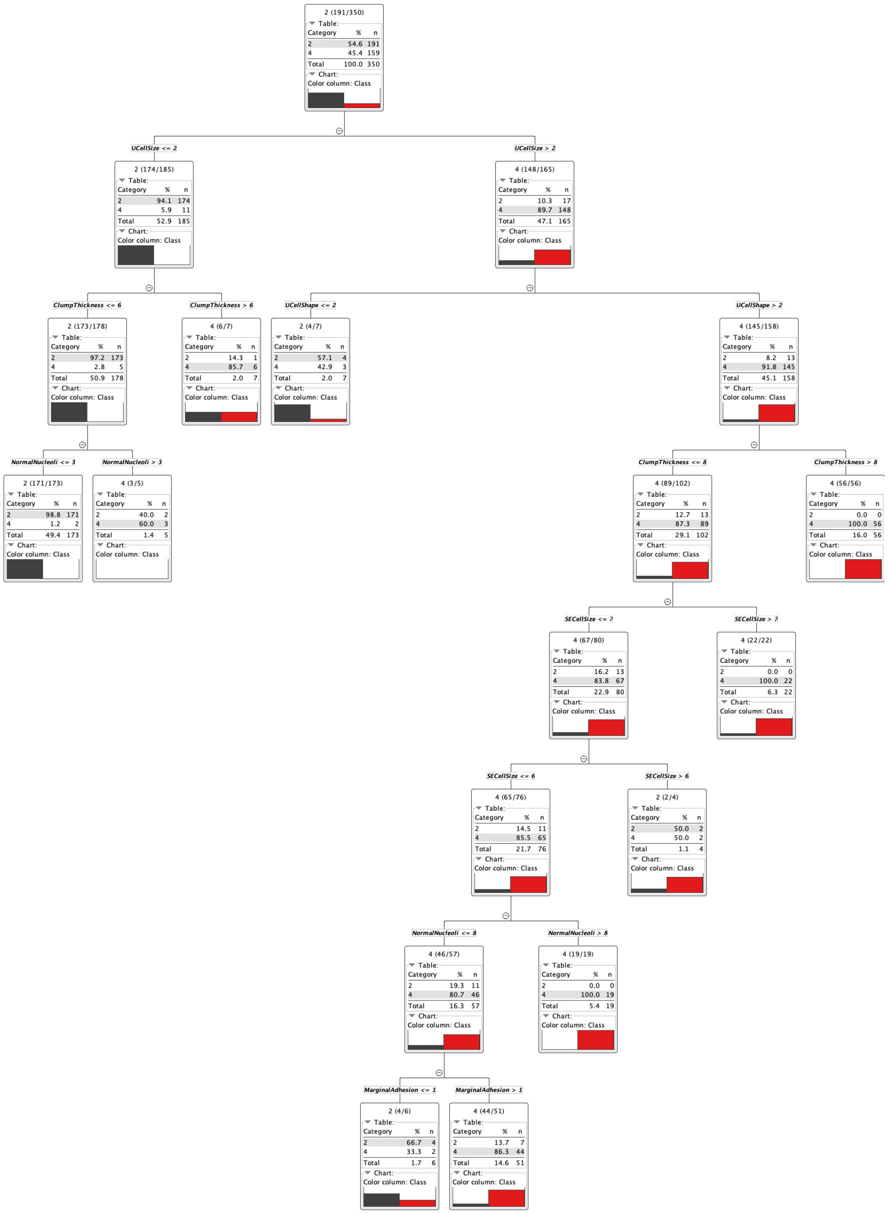


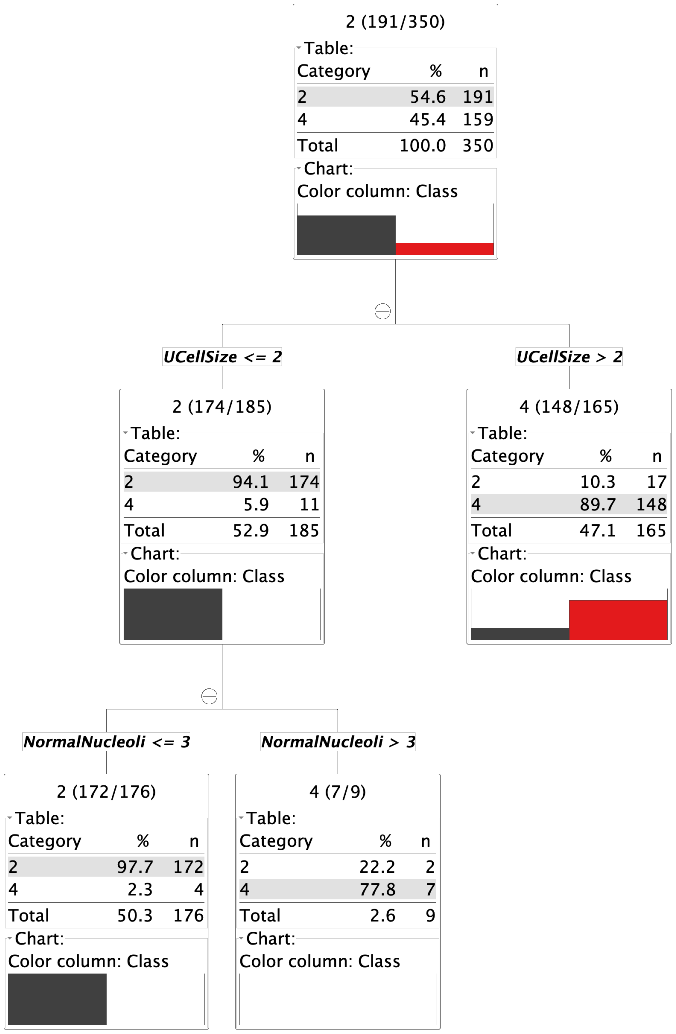
Figure 11: The tree resulting from the experiment. In gray, the rows of class 2 (benign); in red, the rows of class 4 (malignant).

After reviewing the Scorer components, we find:

- Training Accuracy: 94.571% (Correct classified: 331, Wrong classified: 19)
- Testing Accuracy: 95.989% (Correct classified: 335, Wrong classified: 14)

4.6 Experiment 5: Gini Index + 2 Min number records per node + MDL

We obtain the following tree:



2 (191/350)

Category	%	n
2	54.6	191
4	45.4	159
Total	100.0	350

UCellSize <= 2: 2 (174/185)

Category	%	n
2	94.1	174
4	5.9	11
Total	52.9	185

UCellSize > 2: 4 (148/165)

Category	%	n
2	10.3	17
4	89.7	148
Total	47.1	165

NormalNucleoli <= 3: 2 (172/176)

Category	%	n
2	97.7	172
4	2.3	4
Total	50.3	176

NormalNucleoli > 3: 4 (7/9)

Category	%	n
2	22.2	2
4	77.8	7
Total	2.6	9

Figure 12: The tree resulting from the experiment. In gray, the rows of class 2 (benign); in red, the rows of class 4 (malignant).

After reviewing the Scorer components, we find:

- Training Accuracy: 93.429% (Correct classified: 327, Wrong classified: 23)
- Testing Accuracy: 92.837% (Correct classified: 324, Wrong classified: 25)

4.7 Results

Table 3: Experimental Results

# Experiment	QM	Min Records	Training Accuracy	Testing Accuracy
1	Gini Index	2	96.857%	96.562%
2	Gini Index	4	94.286%	96.275%
3	Gain Ratio	2	97.429%	94.842%
4	Gain Ratio	4	94.571%	95.989%
5	Gini Index + MDL	2	93.429%	92.837%

5 Comments

The development of the tree using different split measurements shows that the best fit for the training model comes from the **Gain Ratio** (Experiment 3). Since Gain Ratio is usually aimed at attributes with many different values, and Gini Index performs better with binary attributes, it is not surprising that Gain Ratio performs at least a little bit better, given that all the attributes we are using in the model are not binary but range from 1 to 10. As a consequence, the tree generated by Experiment 3 is the most complex[3], with a depth greater than 10.

Using a smaller **Min number of records per node** will result in more complex trees. This happens because this configuration variable is a stopping criteria that tells the algorithm how detailed it should be. A lower value results in a model that fits the training set better, because of the level of detail it takes in consideration for the split attributes. This can result in an Overfitting problem, where the model fits the training data too well, and does not apply as well to the testing data. In our experimentation we can see this phenomenon, the experiments that use a lower value (2) perform worse on the testing data (compared to their corresponding performance on the training data) than the experiments with the greater value (4). For both, Gain Ratio and Gini Index, increasing the **Min number of records per node** also reduced the complexity of the decision tree.

In all the experiments, **Uniformity of Cell Size** has been selected as the best splitting attribute, followed closely by **Normal Nucleoli**. The former attribute has been defined as an indicator of the consistency in the size of cells in the sample. Cancer cells tend to vary in size. At least for this dataset, this seems to be the best indicator for the class of breast cancer.

[3] We define complexity in this case by the depth of the three, the greater the depth, the more complex is the tree.

Experiment 5 is an extra experiment where we took what we consider our best result: Experiment 1[4], and applied **Minimum Description Length Principle** or **MDL** which is aimed to increase the generalization performance of our model, or prediction quality. In **Figure 12** we can see that the constructed tree is of depth 3, which is way lower than the depth of the trees constructed in the other experiments. Also, by checking the metrics, we have obtained a **Training Accuracy** that is close to that from Experiment 2, and a **Testing Accuracy** that is close to the one obtained in Experiment 3. The tree also coincided with the other experiments in selecting **Uniformity of Cell Size** as the best splitting attribute. We can say that in this experiment we exchanged generalization performance for tree complexity; however, more research is necessary to tell wether the balance of this exchange is positive or negative.

KNIME provides a wide range of components for data manipulation, and easy application of complex algorithms. The documentation provided inside the application for every component makes them easy to use and understand. This documentation usually provides references that help further understand the inner workings of the corresponding components or configuration settings. The visual representation of the workflow also helps to follow the experimentation process. Further analysis of this tool and development of new components is certainly a great way of getting a better grasp of Data Mining.

[4] It has the best balance between Training Accuracy and Testing Accuracy.

References

[1] Tan Steinbach Kumar, *Introduction to Data Mining*, Pearson 1st edition, 2014.

[2] Mayo Clinic, *Diseases & Conditions: Breast Cancer Overview*, Accessed 28 March 2019, <https://www.mayoclinic.org/diseases-conditions/breast-cancer/symptoms-causes/syc-20352470>.

[3] KNIME Website, *About KNIME*, Accessed 28 March 2019, <https://www.knime.com/about>.