ELMA EDISON

A00279906

Programming for Data Analytics – (AL_KDATA_9_1)

12-11-2020

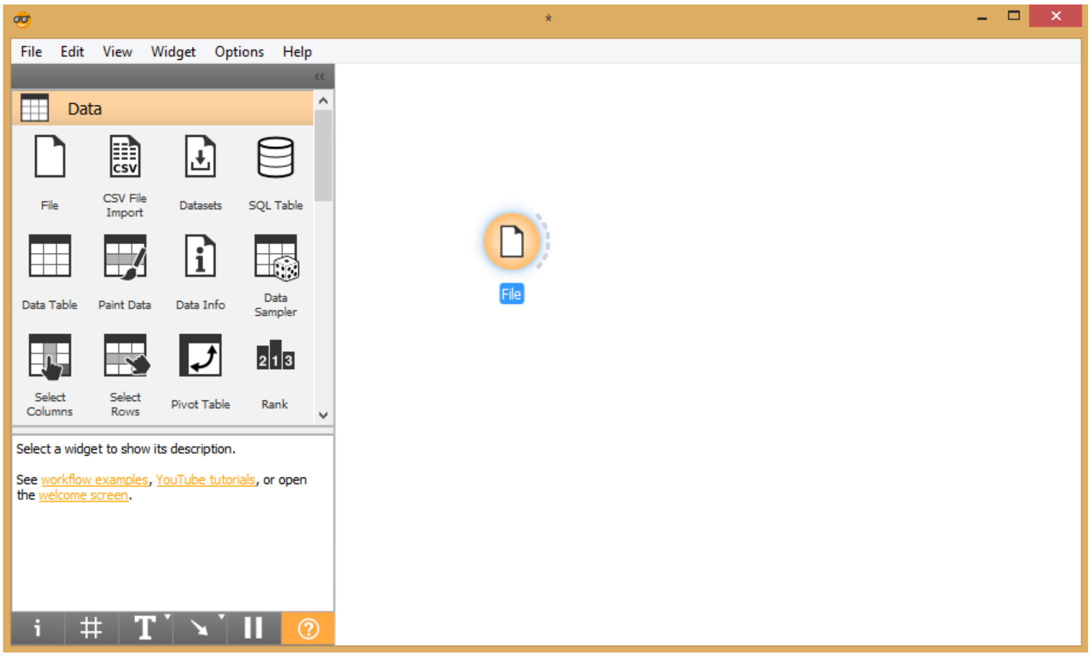
Research on Data Analytics Applications

1. ORANGE

Orange is an open source data mining tool that allows data visualization, data classification and clustering through visual programming. Extended functionalities can be used through python scripting. It has a very attractive and interactive graphical user interface.

It was easy downloading the software and I did not face any issues while installing it. I downloaded the latest version of Orange 3.27.1 from the link given below:
https://orange.biolab.si/download/#windows

After downloading I watched few tutorial videos and got to know about the basic functionalities in Orange and how widgets are used in it. Given below is the screenshot of the interactive user interface in Orange (Figure 1.1).

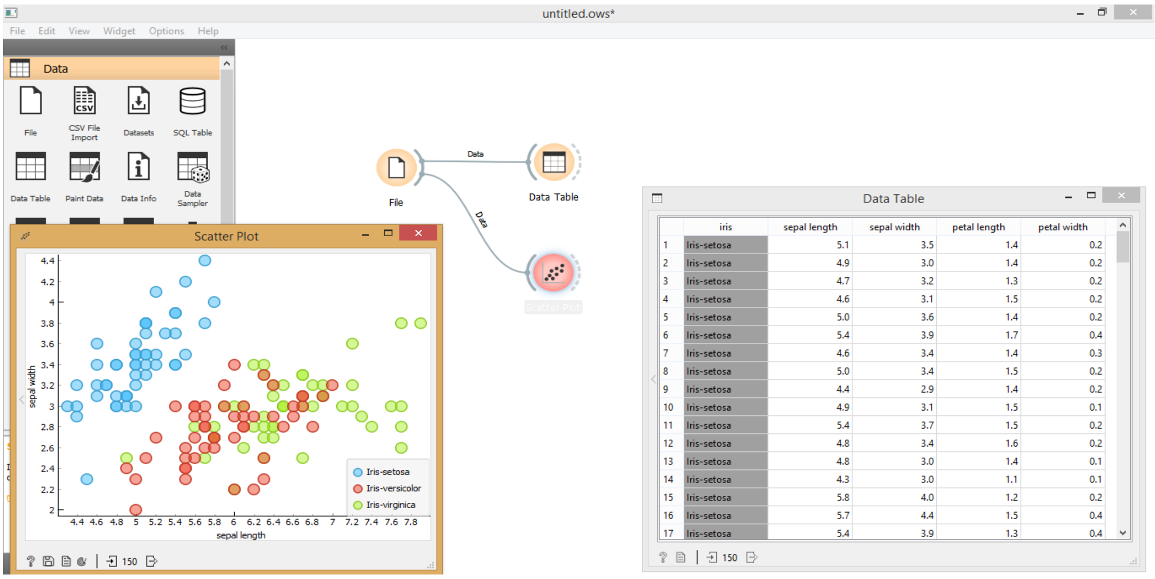


Select a widget to show its description.

See workflow examples, YouTube tutorials, or open the welcome screen.

Figure 1.1

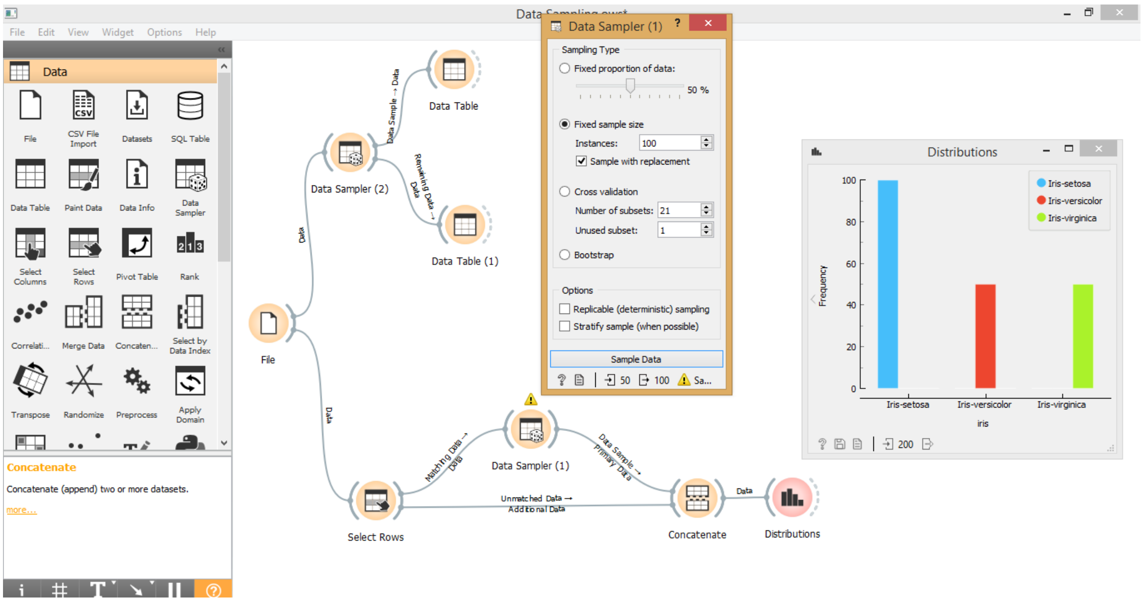
From my experience with using the tool, I think Orange is a great user-friendly data mining tool for beginners. Orange widgets are used for creating schemas for different processes such as clustering, visualization, classification etc. When you select a widget, the description about that particular widget is given in the bottom white box of the Tool Dock which is really helpful and informative. Schemas start with the File widget where in data is read and can be given to other widgets. We can also import a data table from a CSV formatted file using the CSV File Import widget and outputs the dataset. In the small example screenshot given below (Figure 1.2), the file widget is used to read the data which is then sent to both Data Table widget and Scatter Plot widget. We can select widgets from the tool dock or drag and drop or by just right clicking. It is easy to connect widgets by just dragging lines.



	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.1	3.5	1.4	0.2
2	Iris-setosa	4.9	3.0	1.4	0.2
3	Iris-setosa	4.7	3.2	1.3	0.2
4	Iris-setosa	4.6	3.1	1.5	0.2
5	Iris-setosa	5.0	3.6	1.4	0.2
6	Iris-setosa	5.4	3.9	1.7	0.4
7	Iris-setosa	4.6	3.4	1.4	0.3
8	Iris-setosa	5.0	3.4	1.5	0.2
9	Iris-setosa	4.4	2.9	1.4	0.2
10	Iris-setosa	4.9	3.1	1.5	0.1
11	Iris-setosa	5.4	3.7	1.5	0.2
12	Iris-setosa	4.8	3.4	1.6	0.2
13	Iris-setosa	4.8	3.0	1.4	0.1
14	Iris-setosa	4.3	3.0	1.1	0.1
15	Iris-setosa	5.8	4.0	1.2	0.2
16	Iris-setosa	5.7	4.4	1.5	0.4
17	Iris-setosa	5.4	3.9	1.3	0.4

Figure 1.2

I felt that Data Sampler widget is a very useful one which can be used for various data sampling methods such as Fixed proportion of data, Fixed sample size, Cross Validation and Bootstrap. It has two outputs with sampled dataset and a dataset for remaining data (which are instances remaining in the input excluding the sampled dataset). Data Sampler can also be used for Over/Undersampling by increasing the number of instances of the sample with replacement and then by using Concatenate widget, matched and unmatched data can be appended and visualized with Distributions widget. Given below is the screenshot of an example using Data Sampler (Figure 1.3).



Concatenate

Concatenate (append) two or more datasets.

more...

Data Sampler (1)

Sampling Type

○ Fixed proportion of data:

50 %

● Fixed sample size

Instances: 100

☑ Sample with replacement

○ Cross validation

Number of subsets: 21

Unused subset: 1

○ Bootstrap

Options

□ Replicable (deterministic) sampling

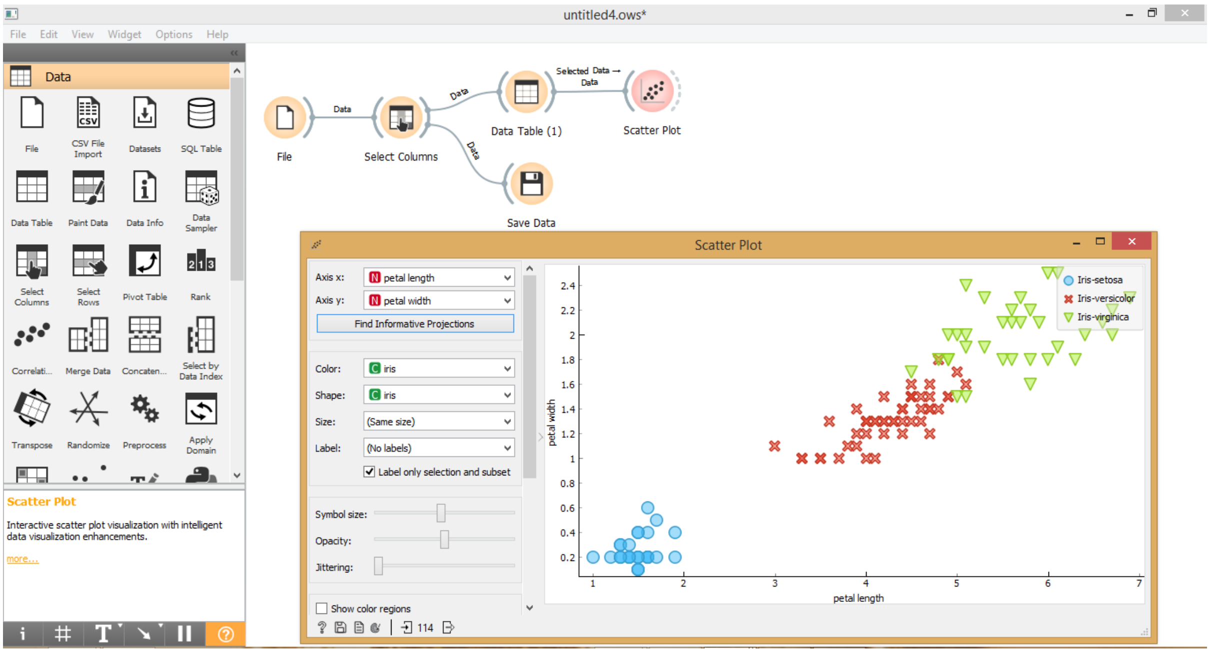
□ Stratify sample (when possible)

Sample Data

50 100 Sa...

Figure 1.3

Data can be plotted with different colour, shape, size and labels for interactive data visualization. We can also change the way the plot is displayed according to the x and y axes. Another useful point is that our data can be saved in whichever format (such as csv, xlsx, tab etc) we want using Save Data widget. Given below is the screenshot of an example (Figure 1.4).



Scatter Plot

Interactive scatter plot visualization with intelligent data visualization enhancements.

more...

Axis x: petal length

Axis y: petal width

Color: iris

Shape: iris

Size: (Same size)

Label: (No labels)

☑ Label only selection and subset

Symbol size:

Opacity:

Jittering:

□ Show color regions

Figure 1.4

Another widget which I found interested is the Hierarchical clustering widget that can be used to group items from a matrix of distances and shows a corresponding dendogram. Given below is the screenshot of an example of the same (Figure 1.5).

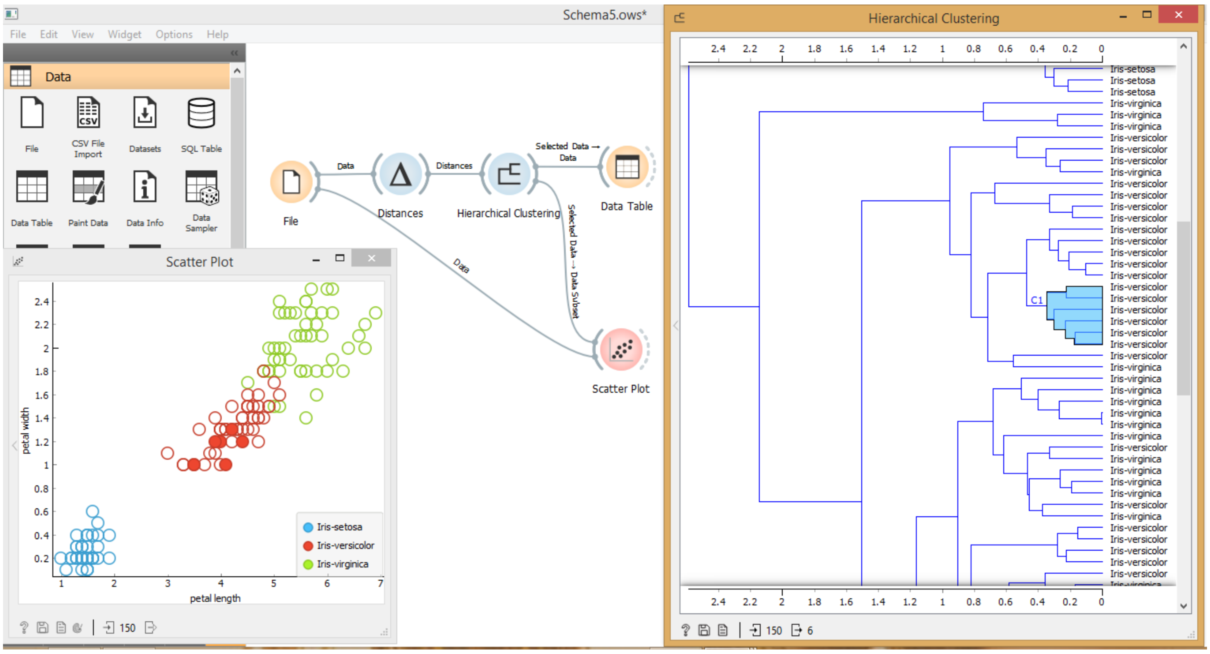
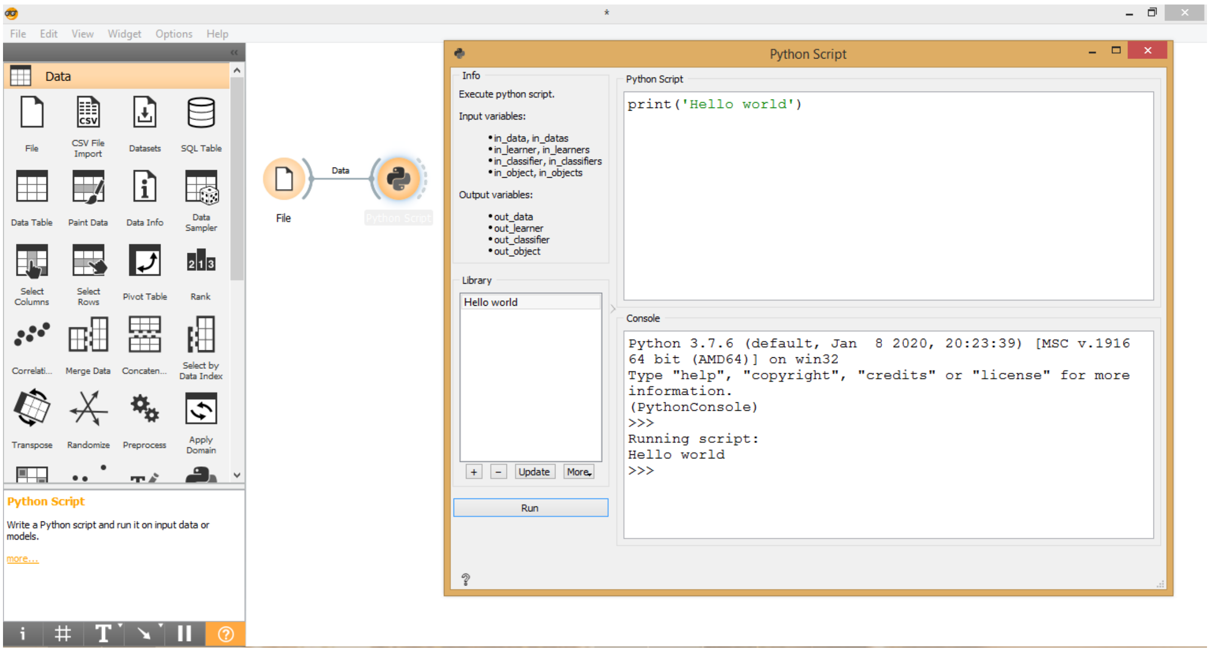


Figure 1.5

Python Script widget is where programming comes into place. It is used to extend functionalities for advanced users by running a python script that includes some input and output variables in its local namespace. Screenshot of the same is given in the Figure 1.6.



Write a Python script and run it on input data or models.

Execute python script.

Input variables:

- in_data, in_datas
- in_learner, in_learners
- in_classifier, in_classifiers
- in_object, in_objects

Output variables:

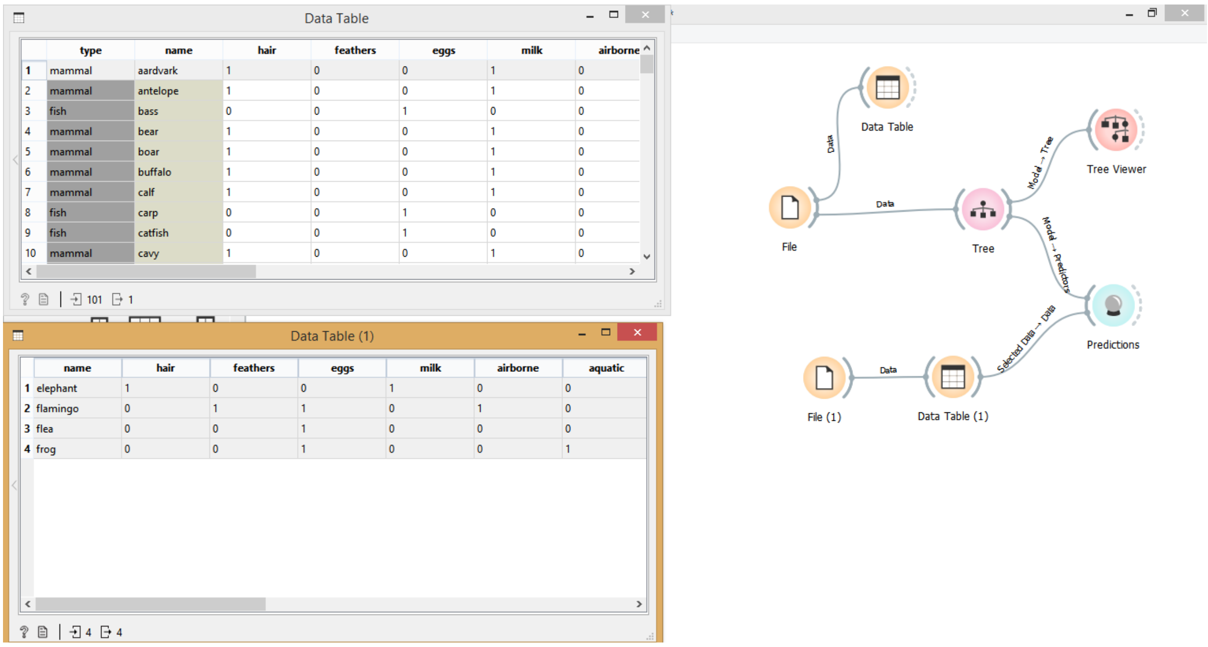
- out_data
- out_learner
- out_classifier
- out_object

```
print('Hello world')
```

```
Python 3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
(PythonConsole)
>>>
Running script:
Hello world
>>>
```

Figure 1.6

Tree widget and Predictions widget are other widgets in Orange that I learned and tried out to do predictions. Predictions widget accepts 2 inputs in which one is test data and other one is Predictors which is the output from any model widget. Here I used Tree widget as the model for classification. Tree is a very useful algorithm that can handle both discrete and continuous datasets which divides data into nodes by class purity. Screenshot of an example schema using zoo dataset for prediction is given in the screenshot (Figure 1.7). Both the test data table and model predictor data table are also shown in the Figure 1.7. Zoo dataset was the data used for model classification and I created a test data with four instances to predict the type. Here the target attribute is set as type which is the column that needs to be predicted whereas all other attributes are used for predicting.



Data Table

	type	name	hair	feathers	eggs	milk	airborne
1	mammal	aardvark	1	0	0	1	0
2	mammal	antelope	1	0	0	1	0
3	fish	bass	0	0	1	0	0
4	mammal	bear	1	0	0	1	0
5	mammal	boar	1	0	0	1	0
6	mammal	buffalo	1	0	0	1	0
7	mammal	calf	1	0	0	1	0
8	fish	carp	0	0	1	0	0
9	fish	catfish	0	0	1	0	0
10	mammal	cavy	1	0	0	1	0

101 1

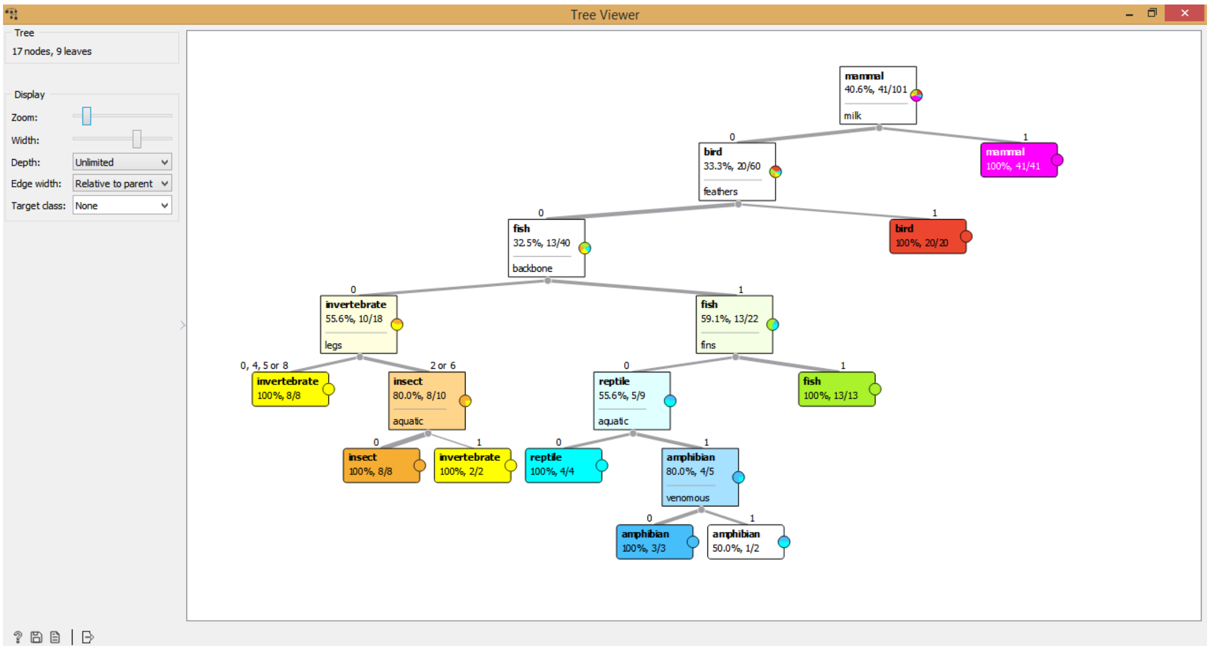
Data Table (1)

	name	hair	feathers	eggs	milk	airborne	aquatic
1	elephant	1	0	0	1	0	0
2	flamingo	0	1	1	0	1	0
3	flea	0	0	1	0	0	0
4	frog	0	0	1	0	0	1

4 4

Figure 1.7

The output of Tree can be viewed using Tree Viewer (Figure1.8).

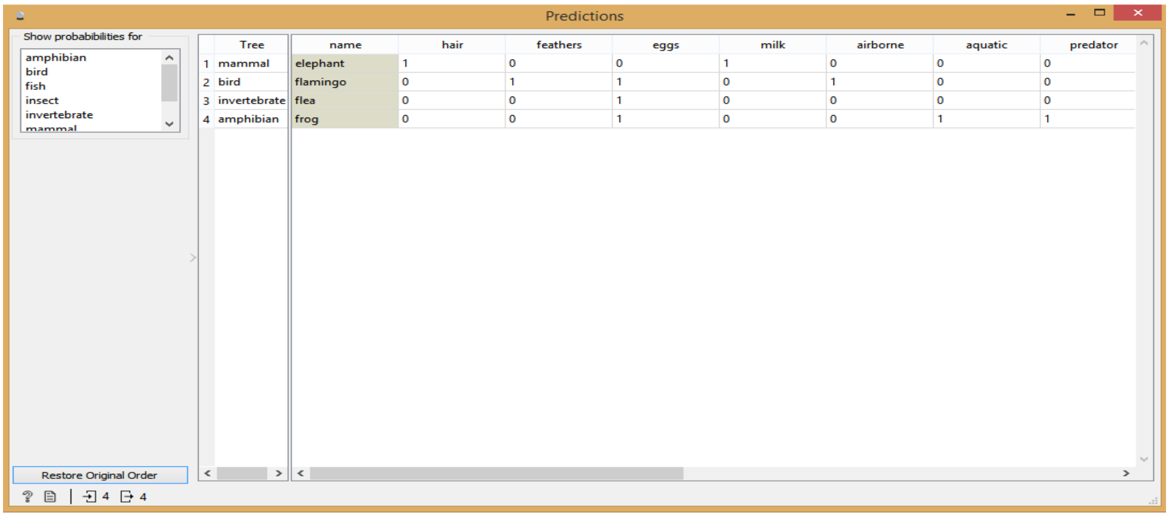

Tree Viewer

Tree
17 nodes, 9 leaves

Display
Zoom:
Width:
Depth: Unlimited
Edge width: Relative to parent
Target class: None

Figure 1.8

The output of Predictions is given in the screenshot below (Figure 1.9). Here the type of four instances were predicted based on other attributes such as hair, feathers, eggs, milk, airborne, aquatic, predator etc. The output of Prediction of the model is that elephant is a mammal, flamingo is a bird, flea is an invertebrate and frog is an amphibian.



Predictions

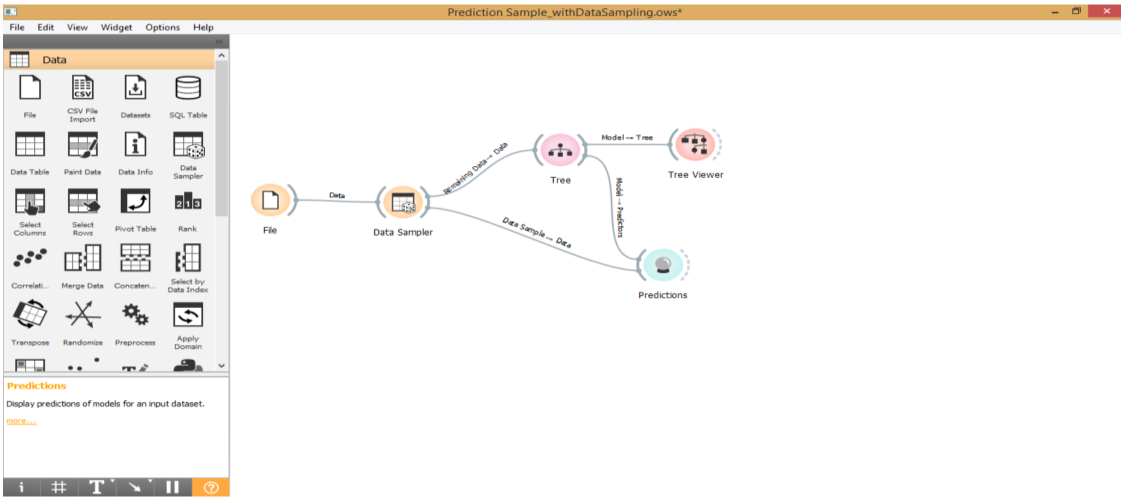
Show probabilities for

amphibian
bird
fish
insect
invertebrate
mammal

	Tree	name	hair	feathers	eggs	milk	airborne	aquatic	predator
1	mammal	elephant	1	0	0	1	0	0	0
2	bird	flamingo	0	1	1	0	1	0	0
3	invertebrate	flea	0	0	1	0	0	0	0
4	amphibian	frog	0	0	1	0	0	1	1

Figure 1.9

The same prediction can also be done by sampling data and using the sampled data as test data and the remaining data as the input for Tree model. Screenshot of the same schema using Data Sampler is shown in the Figure 1.10.



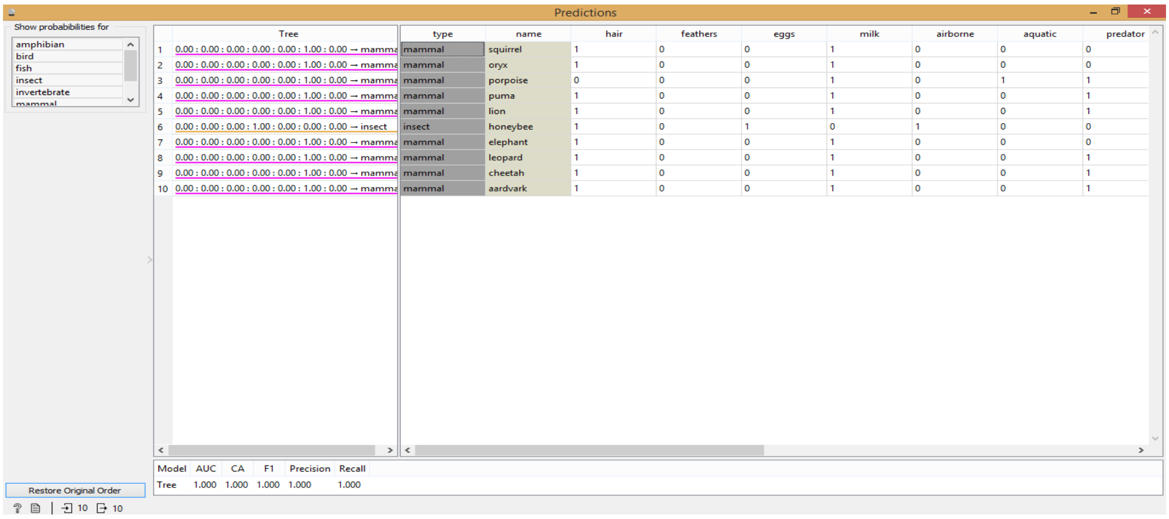
Predictions

Display predictions of models for an input dataset.

more...

Figure 1.10

The output of Predictions of the model is that the type is being predicted for the sampled test data. The screenshot is given below in the Figure 1.11.



	Tree	type	name	hair	feathers	eggs	milk	airborne	aquatic	predator
1	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 → mamma	mammal	squirrel	1	0	0	1	0	0	0
2	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 → mamma	mammal	oryx	1	0	0	1	0	0	0
3	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 → mamma	mammal	porpoise	0	0	0	1	0	1	1
4	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 → mamma	mammal	puma	1	0	0	1	0	0	1
5	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 → mamma	mammal	lion	1	0	0	1	0	0	1
6	0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 → insect	insect	honeybee	1	0	1	0	1	0	0
7	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 → mamma	mammal	elephant	1	0	0	1	0	0	0
8	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 → mamma	mammal	leopard	1	0	0	1	0	0	1
9	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 → mamma	mammal	cheetah	1	0	0	1	0	0	1
10	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 → mamma	mammal	aardvark	1	0	0	1	0	0	1

Model	AUC	CA	F1	Precision	Recall
Tree	1.000	1.000	1.000	1.000	1.000

Figure 1.11

My experience with using Orange was really good. It has a really attractive and interactive user interface which helped me understand it easily and I was able to try out different widgets and got to know about a general idea of what the tool is and why it is used.

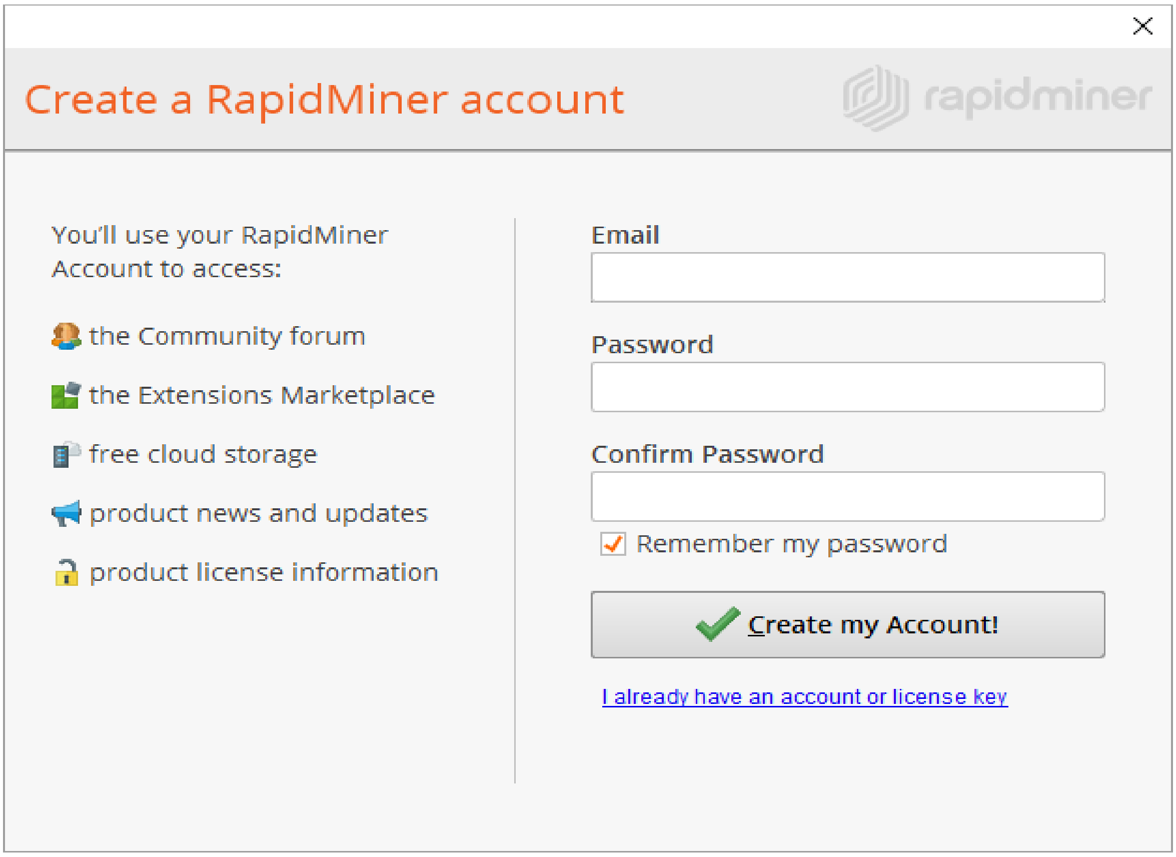
2. RAPIDMINER STUDIO

RapidMiner Studio is an extensive data science platform with visual workflow design and complete automation.

It was easy downloading the software but to install and launch, it took a huge amount of time. I downloaded the latest version of RapidMiner Studio 9.8 from the link given below:

https://rapidminer.com/get-started/

It was very slow while launching the software and asked to create an account to setup. Screenshot of the same is given below (Figure 2.1). Even after the set up and restarting the software, it takes time to open up. I also watched few youtube videos to get a basic understanding of the software.



Create a RapidMiner account

You'll use your RapidMiner Account to access:

- the Community forum
- the Extensions Marketplace
- free cloud storage
- product news and updates
- product license information

Email

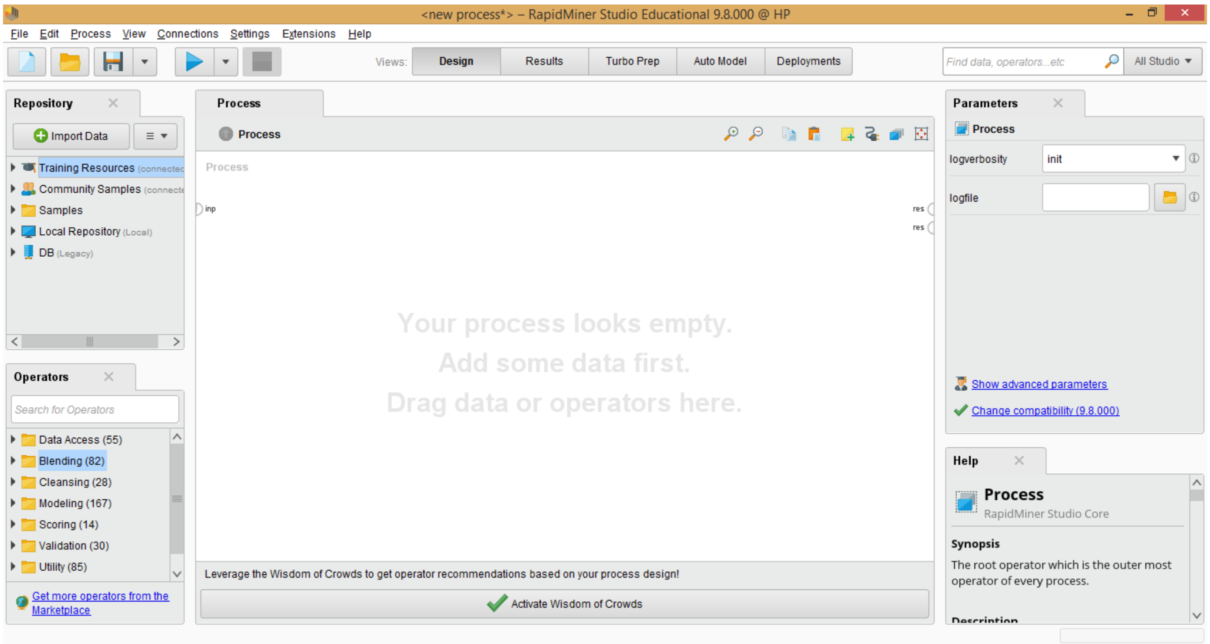
Password

Confirm Password

☑ Remember my password

Figure 2.1

The user interface of RapidMiner Studio is somewhat user friendly but can be a bit confusing for beginners. For experts, this tool can be really helpful with a great visual workflow and automation. It can also extend functionalities using R and Python code. Its UI consists of Repository, Operators, Process panel, Views, Global search area, Ports, Parameters and Help. In Views, Design and Results can be defined as the work areas which are equipped to carry out specific tasks for data preparation, modelling etc and to access specific functionalities. The Process panel is used as a work area to design and build any process. Repository is a storage within RapidMiner Studio for data and RapidMiner processes. Operators are building blocks that are used to create RapidMiner processes. Ports are input and output channels for operators and processes. Parameters are used to modify operator behaviour. These are few of the panels that I worked on to understand the basic functionalities. Given below is the screenshot of the RapidMiner Studio UI (Figure 2.2).



<new process*> – RapidMiner Studio Educational 9.8.000 @ HP
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Find data, operators...etc
All Studio
Repository
Import Data
Training Resources (connected)
Community Samples (connected)
Samples
Local Repository (Local)
DB (Legacy)
Operators
Search for Operators
Data Access (55)
Blending (82)
Cleansing (28)
Modeling (167)
Scoring (14)
Validation (30)
Utility (85)
Get more operators from the Marketplace
Process
Process
Process
inp
res
res
Parameters
Process
logverbosity
init
logfile
Show advanced parameters
Change compatibility (9.8.000)
Help
Process
RapidMiner Studio Core
Synopsis
Description
Activate Wisdom of Crowds

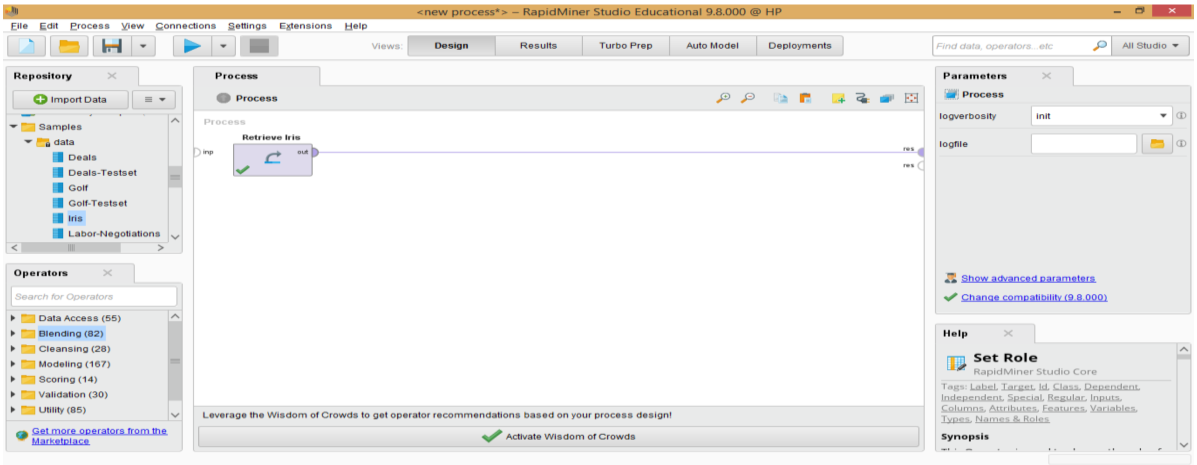
Your process looks empty.
Add some data first.
Drag data or operators here.

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

The root operator which is the outer most operator of every process.

Figure 2.2

RapidMiner Studio has in-built datasets such as iris, titanic etc where the input file needs to be just dragged and dropped. The output of the file operator must be given to the output port and then we have to run the process in order to view the results. The drag + drop visual interface not only speeds up but also automates the creation of predictive models. Given below is the screenshot of the same (Figure 2.3).

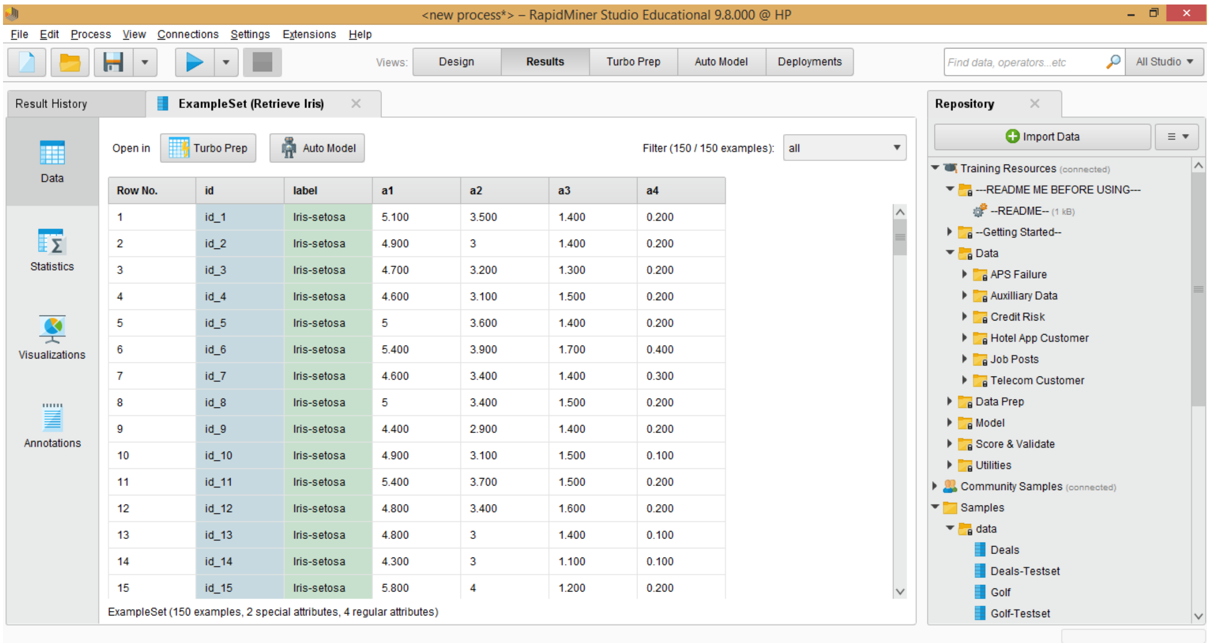


Tags: Label, Target, Id, Class, Dependent, Independent, Special, Regular, Inputs, Columns, Attributes, Features, Variables, Types, Names & Roles

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Figure 2.3

In the Results view, it is possible to view the data table (Figure 2.4), statistics (Figure 2.5) and data can be visualized using Scatterplot (Figure 2.6). Here rows are called Examples and columns are called Attributes.

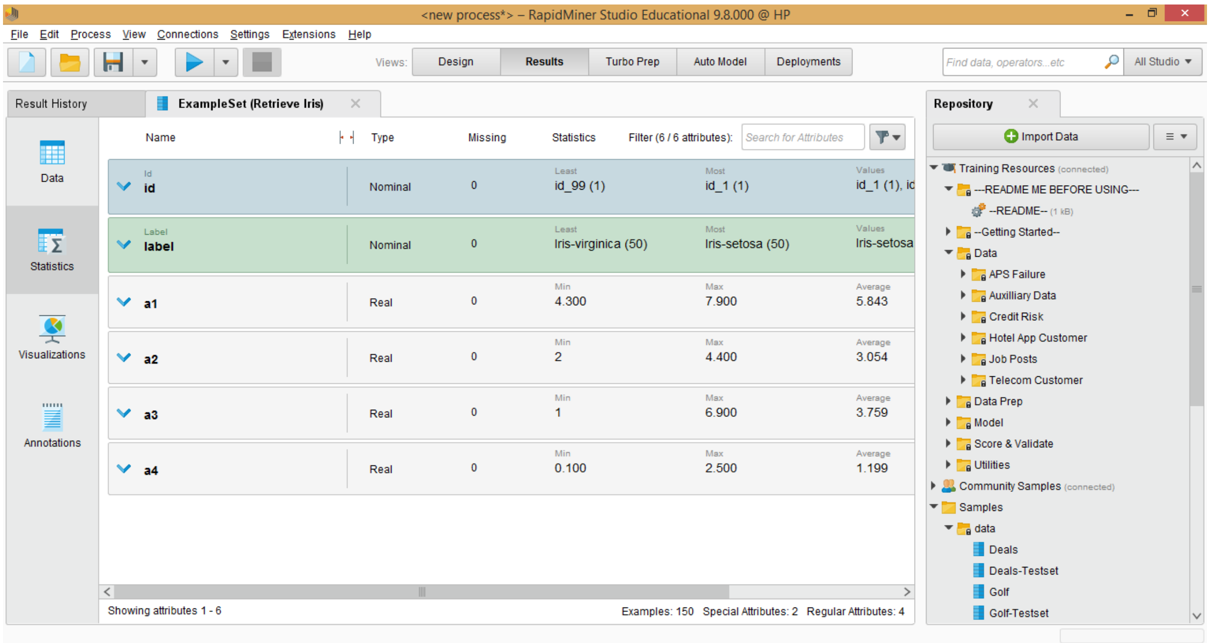

<new process*> – RapidMiner Studio Educational 9.8.000 @ HP
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Find data, operators...etc
All Studio
Result History
ExampleSet (Retrieve Iris)
Data
Statistics
Visualizations
Annotations
Open in Turbo Prep Auto Model
Repository
Import Data
Training Resources (connected)
--README ME BEFORE USING--
--README-- (1 kB)
--Getting Started--
Data
APS Failure
Auxiliary Data
Credit Risk
Hotel App Customer
Job Posts
Telecom Customer
Data Prep
Model
Score & Validate
Utilities
Community Samples (connected)
Samples
data
Deals
Deals-Testset
Golf
Golf-Testset

Filter (150 / 150 examples): all

Row No.	id	label	a1	a2	a3	a4
1	id_1	Iris-setosa	5.100	3.500	1.400	0.200
2	id_2	Iris-setosa	4.900	3	1.400	0.200
3	id_3	Iris-setosa	4.700	3.200	1.300	0.200
4	id_4	Iris-setosa	4.600	3.100	1.500	0.200
5	id_5	Iris-setosa	5	3.600	1.400	0.200
6	id_6	Iris-setosa	5.400	3.900	1.700	0.400
7	id_7	Iris-setosa	4.600	3.400	1.400	0.300
8	id_8	Iris-setosa	5	3.400	1.500	0.200
9	id_9	Iris-setosa	4.400	2.900	1.400	0.200
10	id_10	Iris-setosa	4.900	3.100	1.500	0.100
11	id_11	Iris-setosa	5.400	3.700	1.500	0.200
12	id_12	Iris-setosa	4.800	3.400	1.600	0.200
13	id_13	Iris-setosa	4.800	3	1.400	0.100
14	id_14	Iris-setosa	4.300	3	1.100	0.100
15	id_15	Iris-setosa	5.800	4	1.200	0.200

ExampleSet (150 examples, 2 special attributes, 4 regular attributes)

Figure 2.4


<new process*> – RapidMiner Studio Educational 9.8.000 @ HP
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Find data, operators...etc
All Studio
Result History
ExampleSet (Retrieve Iris)
Data
Statistics
Visualizations
Annotations
Search for Attributes
Repository
Import Data
Training Resources (connected)
--README ME BEFORE USING--
--README-- (1 kB)
--Getting Started--
Data
APS Failure
Auxiliary Data
Credit Risk
Hotel App Customer
Job Posts
Telecom Customer
Data Prep
Model
Score & Validate
Utilities
Community Samples (connected)
Samples
data
Deals
Deals-Testset
Golf
Golf-Testset

Filter (6 / 6 attributes):

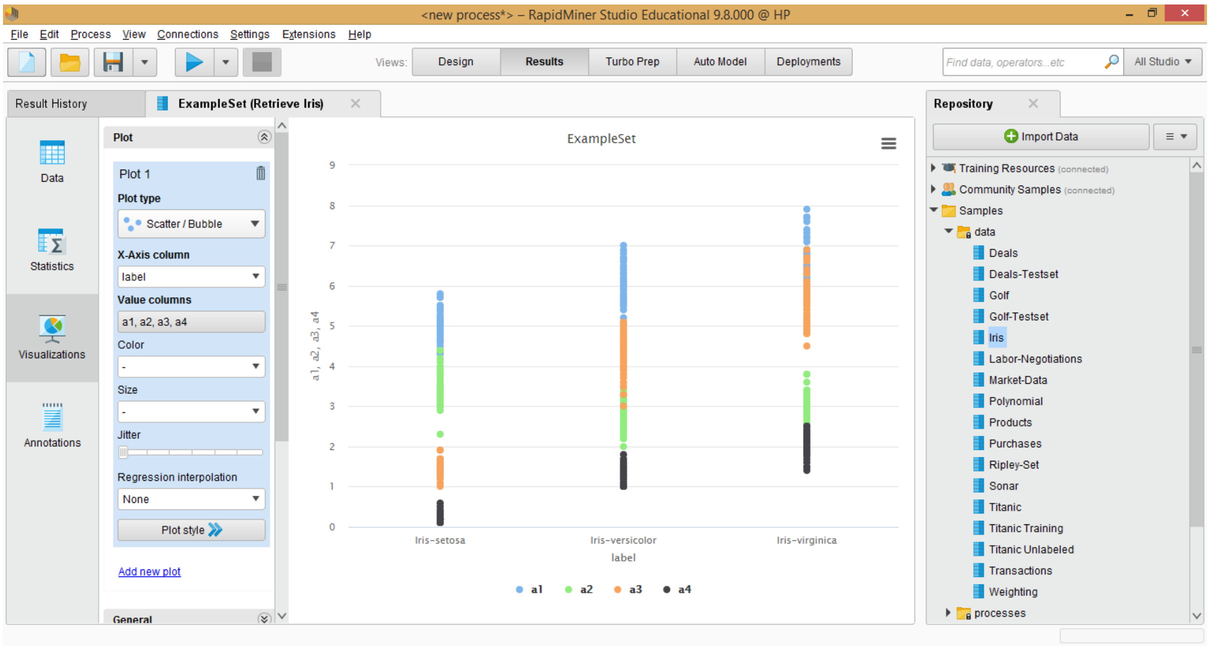
Name	Type	Missing	Statistics		
Id id	Nominal	0	Least id_99 (1)	Most id_1 (1)	Values id_1 (1), id
Label label	Nominal	0	Least Iris-virginica (50)	Most Iris-setosa (50)	Values Iris-setosa
a1	Real	0	Min 4.300	Max 7.900	Average 5.843
a2	Real	0	Min 2	Max 4.400	Average 3.054
a3	Real	0	Min 1	Max 6.900	Average 3.759
a4	Real	0	Min 0.100	Max 2.500	Average 1.199

Showing attributes 1 - 6

Examples: 150 Special Attributes: 2 Regular Attributes: 4

Figure 2.5

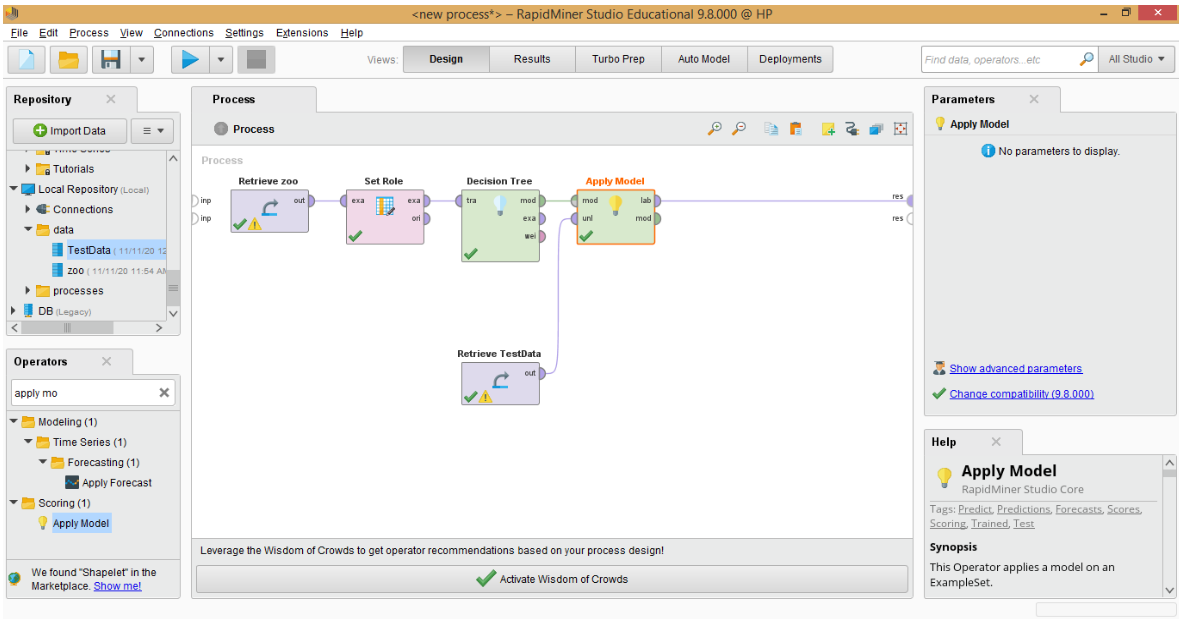
There are a numerous variety of plot types under visualizations which is actually a good point about it. In the Scatterplot, x-axis columns and value columns can be changed accordingly to attain different plots. Here with a variety of plots, colour and size can also be changed which makes it more fun to work on these visualization models. Given below is an example (Figure 2.6).



Plot type: Scatter / Bubble
X-Axis column: label
Value columns: a1, a2, a3, a4
Color: -
Size: -
Jitter
Regression interpolation: None
Plot style

Figure 2.6

After getting a basic idea of RapidMiner Studio, I tried to predict a model by importing a dataset. Screenshot of the design view is given below (Figure 2.7). I took zoo dataset and Set Role operator was used for setting the "type" column (prediction attribute) in dataset as the label (prediction attribute role). Then the output from the Set Role that is the train data is given as input to the Decision Tree modelling operator whose output is shown in the Figure 2.8. A test data is also imported so as to predict the model using the scoring operator called Apply Model.



We found "Shapelet" in the Marketplace. Show me!

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

No parameters to display.

This Operator applies a model on an ExampleSet.

Figure 2.7

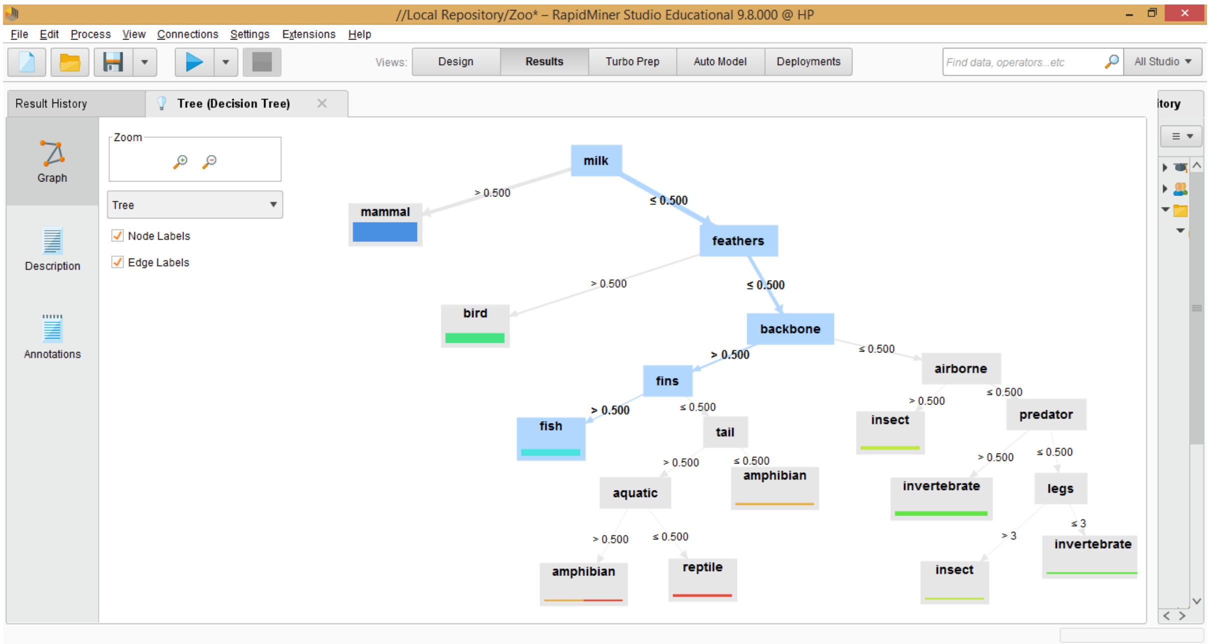
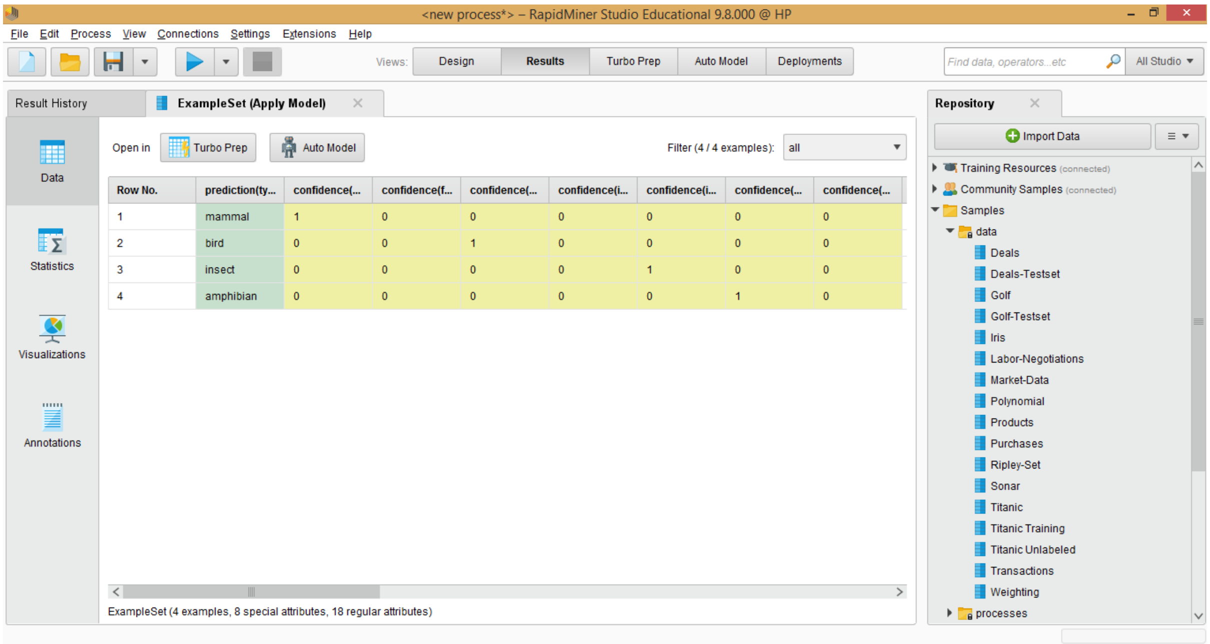


Figure 2.8

The result view shows the test data prediction that is the output of the Apply Model in the figure given below (Figure 2.9). It shows predictions and confidences for the inputs.



ExampleSet (Apply Model)

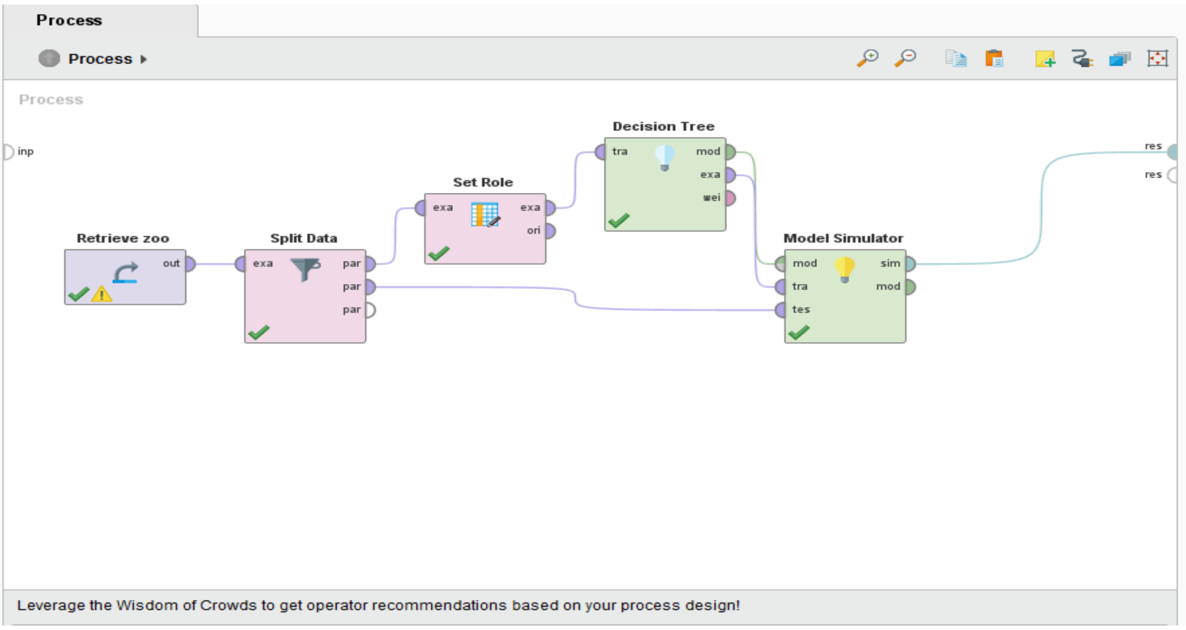
Open in Turbo Prep Auto Model Filter (4 / 4 examples): all

Row No.	prediction(ty...	confidence(...	confidence(f...	confidence(...	confidence(i...	confidence(i...	confidence(...	confidence(...
1	mammal	1	0	0	0	0	0	0
2	bird	0	0	1	0	0	0	0
3	insect	0	0	0	0	1	0	0
4	amphibian	0	0	0	0	0	1	0

ExampleSet (4 examples, 8 special attributes, 18 regular attributes)

Figure 2.9

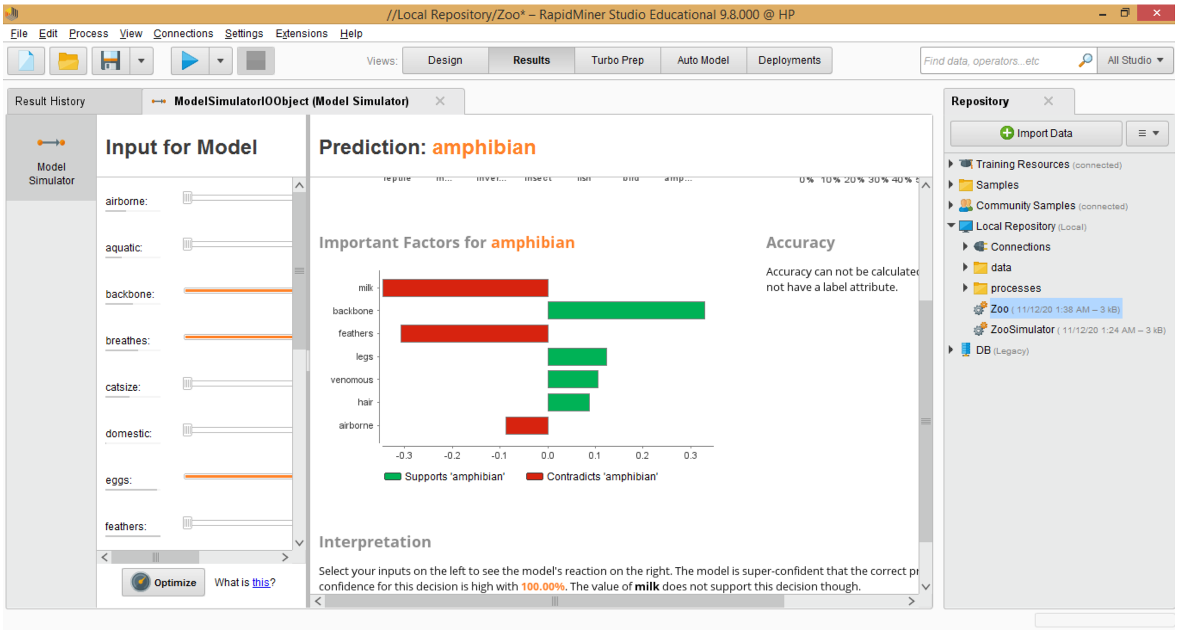
The above example was to predict a test/sampled data wherein the data was separately given as input to Apply Model operator. But in the same example given below (Figure 2.10), I used Split Data operator to split data instances with the required ratio so that from the two outputs of Split Data operator, one is used as train data and other is used as test data. The train data is then given as input to the Split role operator where the "type" column is set as label and then given to the Decision Tree operator. The output from the Decision Tree and the test data is given as input to the Model Simulator operator.



Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Figure 2.10

The output of the Model Simulator is given below in the Figure 2.11 which shows the predictions, confidences and explanations for the inputs. Model Simulator is an operator that provides an easy real time modelling method so that we can make changes in output and have a much more detailed output regarding the attributes.



Accuracy can not be calculatec
not have a label attribute.

Select your inputs on the left to see the model's reaction on the right. The model is super-confident that the correct pr
confidence for this decision is high with 100.00%. The value of **milk** does not support this decision though.

Figure 2.11

My experience with using RapidMiner was good. Initially I took time to grasp things but after watching some tutorials and trying out few examples, it was really interesting. RapidMiner Studio might be somewhat difficult for a beginner but for experts, I think this tool would really help them to extend functionalities and build advanced models. It has a good user interface which helped me understand the tool after few days of working on it. I was able to try out different operators and got to know about a general idea of what the tool is and why it is used.

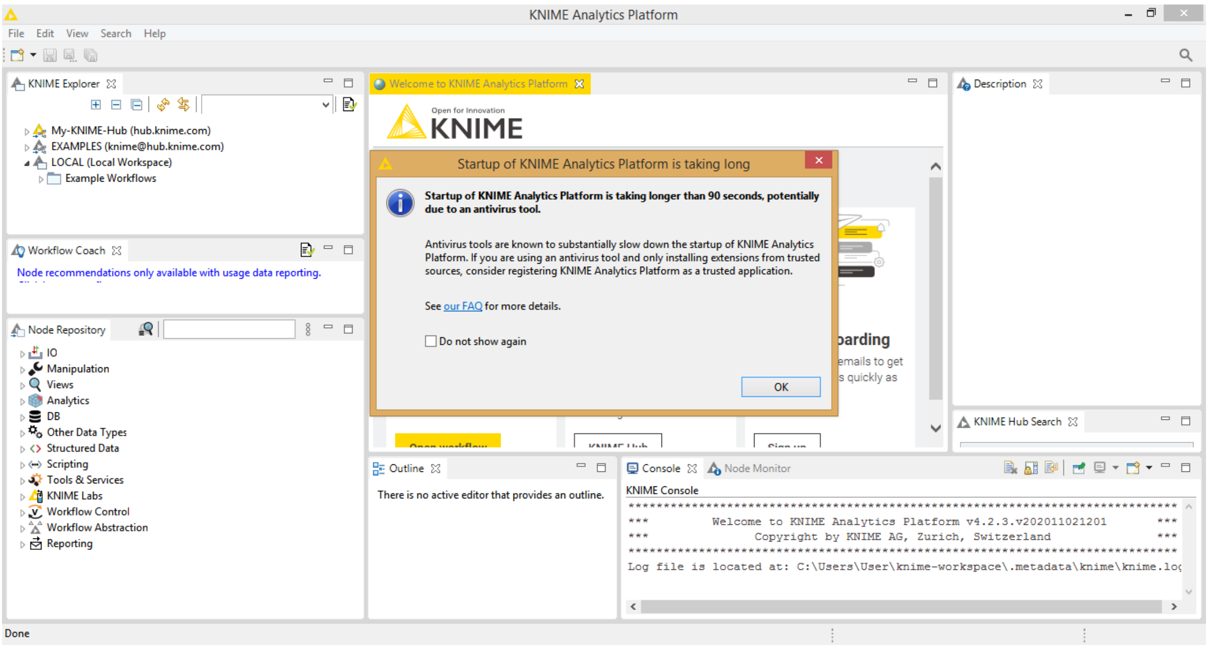
3. KNIME

The Konstanz Information Miner, known as KNIME is a free open source data analytics platform which can be used for integration and reporting also.

It was easy downloading the software but to install and launch, it took a huge amount of time. I downloaded the latest version of Knime 4.2.3 from the link given below:

https://www.knime.com/download-installer/2/64bit

It was very slow while installing the software and then a dialogue box popped up saying that the startup took so long because of Antivirus tool. Screenshot of the same is given below (Figure 3.1). I also watched few youtube videos to acquire a basic knowledge of the software.



Node recommendations only available with usage data reporting.

Startup of KNIME Analytics Platform is taking longer than 90 seconds, potentially due to an antivirus tool.

Antivirus tools are known to substantially slow down the startup of KNIME Analytics Platform. If you are using an antivirus tool and only installing extensions from trusted sources, consider registering KNIME Analytics Platform as a trusted application.

See our FAQ for more details.

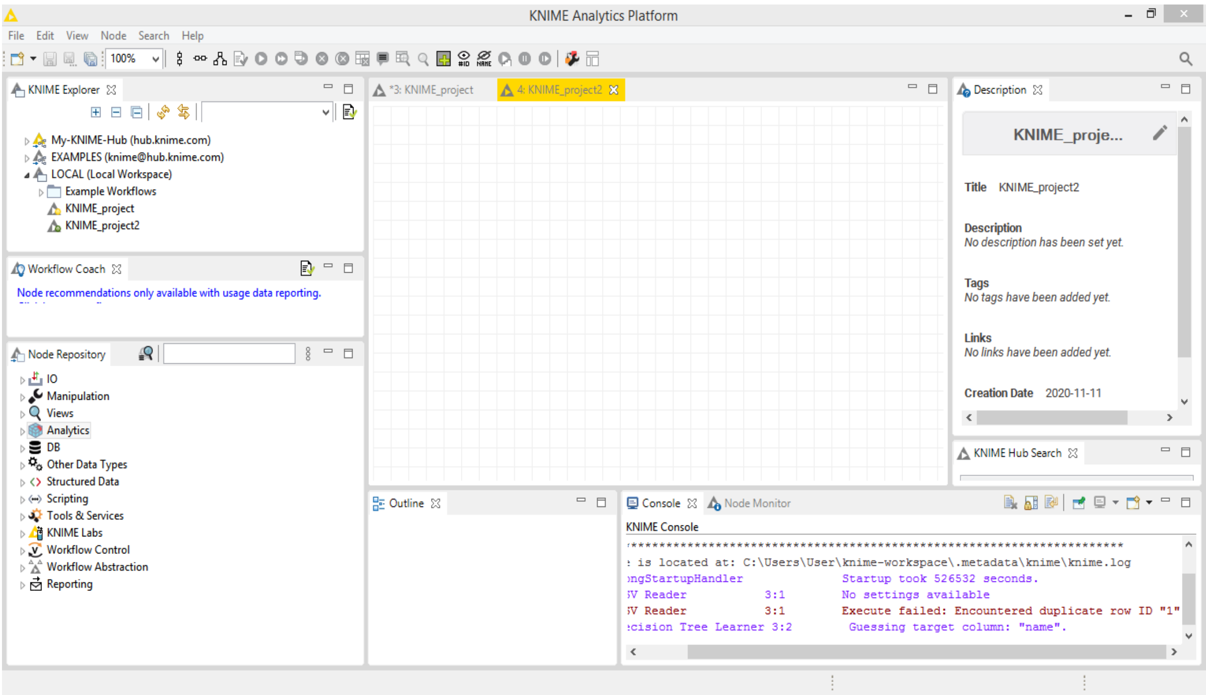
□ Do not show again

There is no active editor that provides an outline.

```
*******************************************************************************
***          Welcome to KNIME Analytics Platform v4.2.3.v202011021201       ***
***                 Copyright by KNIME AG, Zurich, Switzerland               ***
*******************************************************************************
Log file is located at: C:\Users\User\knime-workspace\.metadata\knime\knime.log
```

Figure 3.1

The user interface of Knime is not user friendly and it was kind of difficult for me at the beginning to understand its workflow and the working of nodes. UI of Knime consists of different panels such as Knime Explorer, Workflow Editor, Node Repository, Description, Knime console, Knime hub search and few more. Knime Explorer allows you to browse your workflows and to act upon them. Also, lot of pre-built examples by Knime are available in it. I went through some of them, but it was not that helpful for me. Local workspace is available to create either a New Knime workflow or a New workflow group. When a New Knime workflow is created, the workflow editor opens up which is a canvas for editing the currently active workflow. Node Repository is a collection of nodes which contains prewritten programs. It allows to drag and drop multiple nodes to the workflow editor or double clicking the node will also bring nodes to create the workflow. Description panel displays the information of a node. Given below is the screenshot of the user interface of Knime analytics platform (Figure 3.2).



Node recommendations only available with usage data reporting.

KNIME_proje...

Title KNIME_project2

Description
No description has been set yet.

Tags
No tags have been added yet.

Links
No links have been added yet.

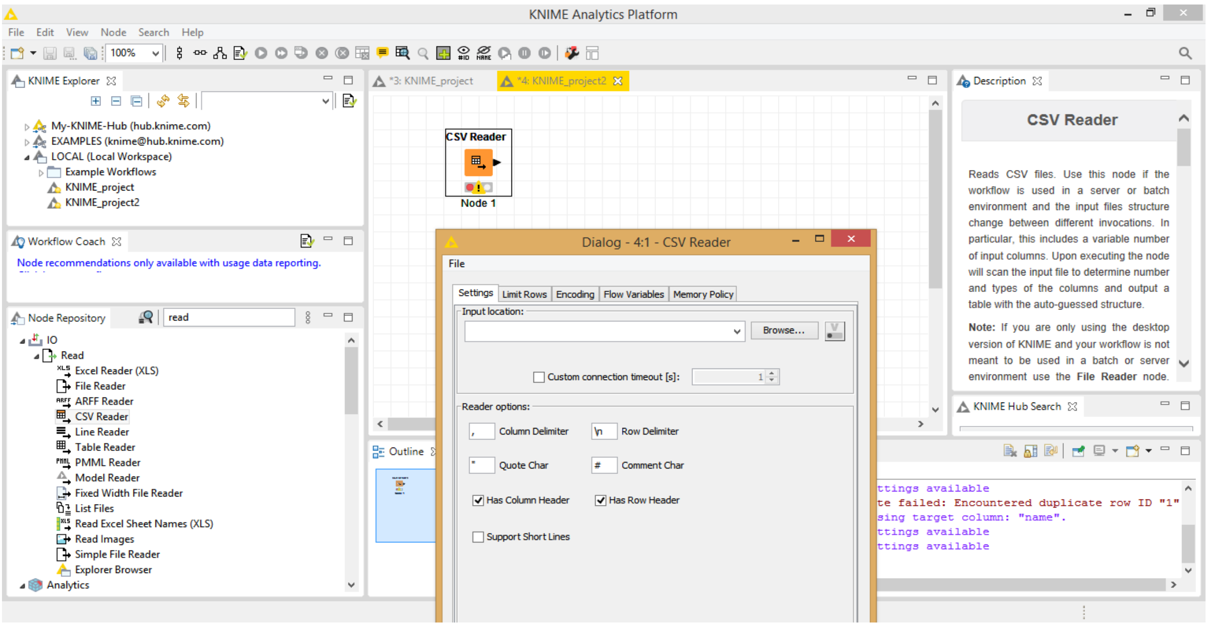
Creation Date 2020-11-11

KNIME Console

```
*************************************************************************
: is located at: C:\Users\User\knime-workspace\.metadata\knime\knime.log
ongStartupHandler            Startup took 526532 seconds.
SV Reader            3:1        No settings available
SV Reader            3:1        Execute failed: Encountered duplicate row ID "1"
ecision Tree Learner 3:2         Guessing target column: "name".
```

Figure 3.2

Nodes may contain one, two or more input ports on the left-hand side of the node and the output port is on the right-hand side. The node state determines the status of the node i.e., Red colour states that the node is not configured, Yellow colour states that the node is configured (ready to be executed), Green colour states that the node is executed and red cross symbol states that it is an error. Given below is an example of a CSV Reader node (Figure 3.3) which was dragged and dropped into the workflow editor. It then needs to be configured where the csv file is given as input. The node state is actually an interesting feature, but it is asked to reset, configure and re-execute everytime when changes were made. At times, I found it hard to move the nodes according to my wish. Overall, I was not satisfied with the user interface. When using Knime, my laptop got hang few times. I'm not sure whether it is my laptop's problem, but I did find it slow downing all other applications also.



Node recommendations only available with usage data reporting.

Dialog - 4:1 - CSV Reader

File

Settings | Limit Rows | Encoding | Flow Variables | Memory Policy

Input location:

Browse...

□ Custom connection timeout [s]: 1

Reader options:

, Column Delimiter	\n Row Delimiter

" Quote Char	# Comment Char

☑ Has Column Header	☑ Has Row Header

□ Support Short Lines

Description

CSV Reader

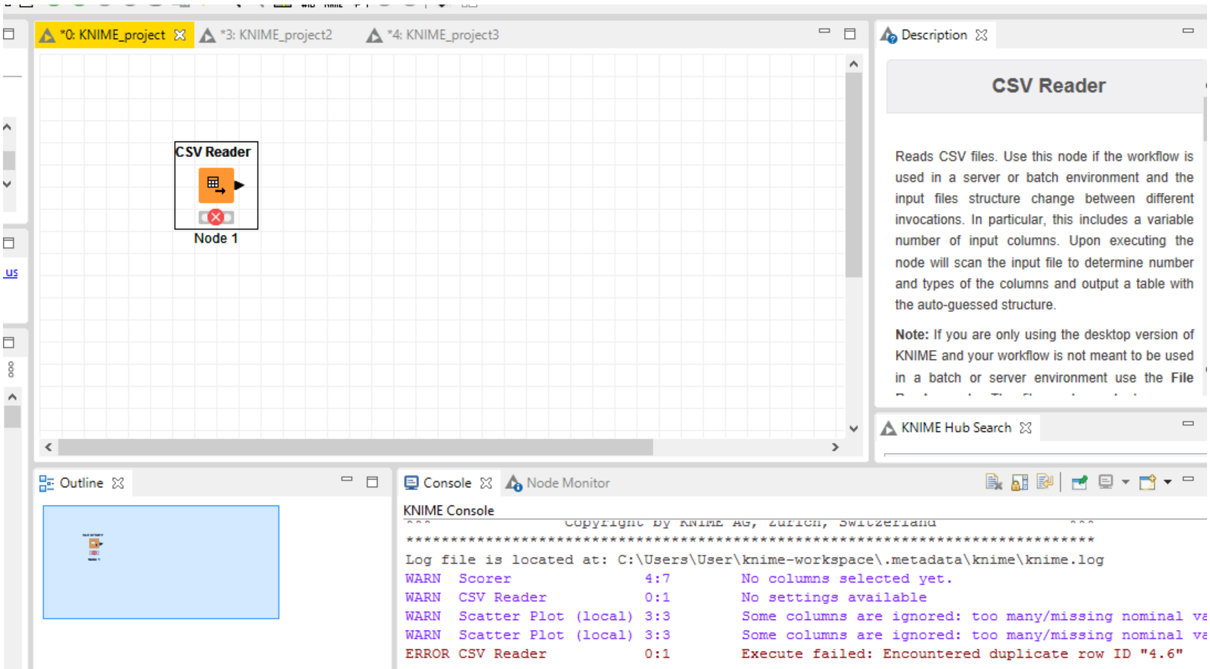
Reads CSV files. Use this node if the workflow is used in a server or batch environment and the input files structure change between different invocations. In particular, this includes a variable number of input columns. Upon executing the node will scan the input file to determine number and types of the columns and output a table with the auto-guessed structure.

Note: If you are only using the desktop version of KNIME and your workflow is not meant to be used in a batch or server environment use the **File Reader** node.

```
ttings available
te failed: Encountered duplicate row ID "1"
sing target column: "name".
ttings available
ttings available
```

Figure 3.3

Here in an example, a CSV Reader node was used to input the iris dataset. At first, the red cross error symbol was shown because here in Knime, each row in the dataset should have a unique id. The dataset I uploaded initially did not have any unique ids and that is why it showed an error message. Screenshot of the same is given in the Figure 3.4.



Description

CSV Reader

Reads CSV files. Use this node if the workflow is used in a server or batch environment and the input files structure change between different invocations. In particular, this includes a variable number of input columns. Upon executing the node will scan the input file to determine number and types of the columns and output a table with the auto-guessed structure.

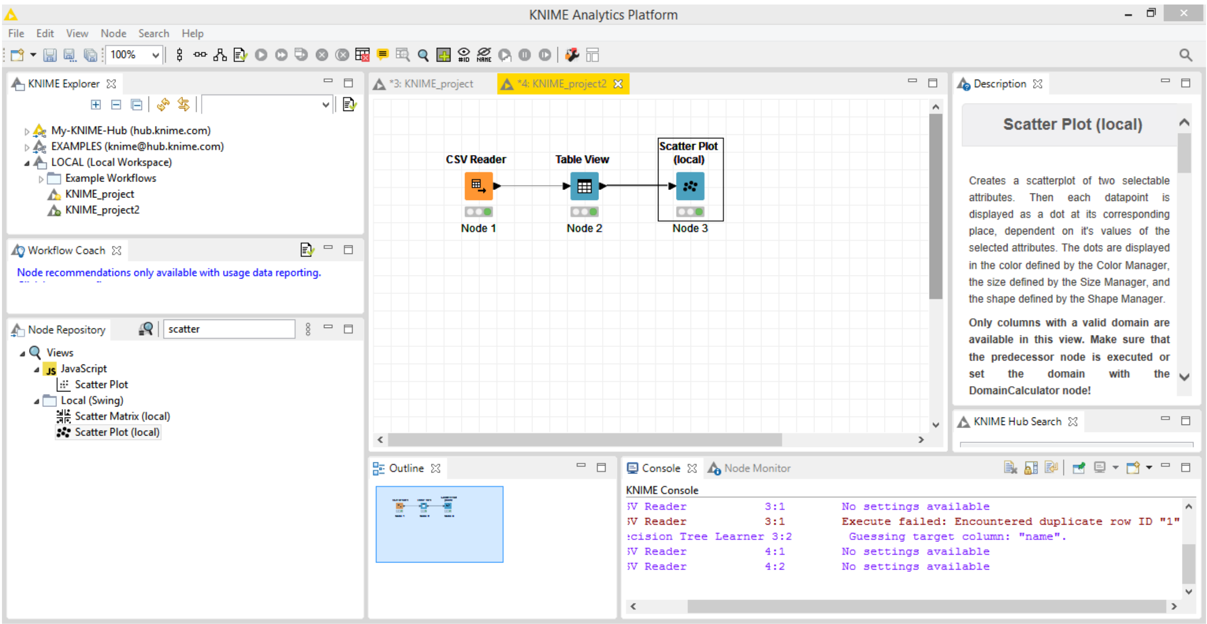
Note: If you are only using the desktop version of KNIME and your workflow is not meant to be used in a batch or server environment use the **File**

KNIME Console

```
***           Copyright by KNIME AG, Zurich, Switzerland           ***
**********************************************************************
Log file is located at: C:\Users\User\knime-workspace\.metadata\knime\knime.log
WARN  Scorer               4:7       No columns selected yet.
WARN  CSV Reader           0:1       No settings available
WARN  Scatter Plot (local) 3:3       Some columns are ignored: too many/missing nominal va
WARN  Scatter Plot (local) 3:3       Some columns are ignored: too many/missing nominal va
ERROR CSV Reader           0:1       Execute failed: Encountered duplicate row ID "4.6"
```

Figure 3.4

The CSV Reader is then configured with the right dataset, executed and the state is turned into Green. An example using iris dataset is shown in the Figure 3.5. After that the CSV Reader is connected to a Table View where the data table is viewed. And then it is connected to a scatter plot to visualize the data. All the nodes have turned into Green state which means all are executed.



Scatter Plot (local)

Creates a scatterplot of two selectable attributes. Then each datapoint is displayed as a dot at its corresponding place, dependent on it's values of the selected attributes. The dots are displayed in the color defined by the Color Manager, the size defined by the Size Manager, and the shape defined by the Shape Manager.

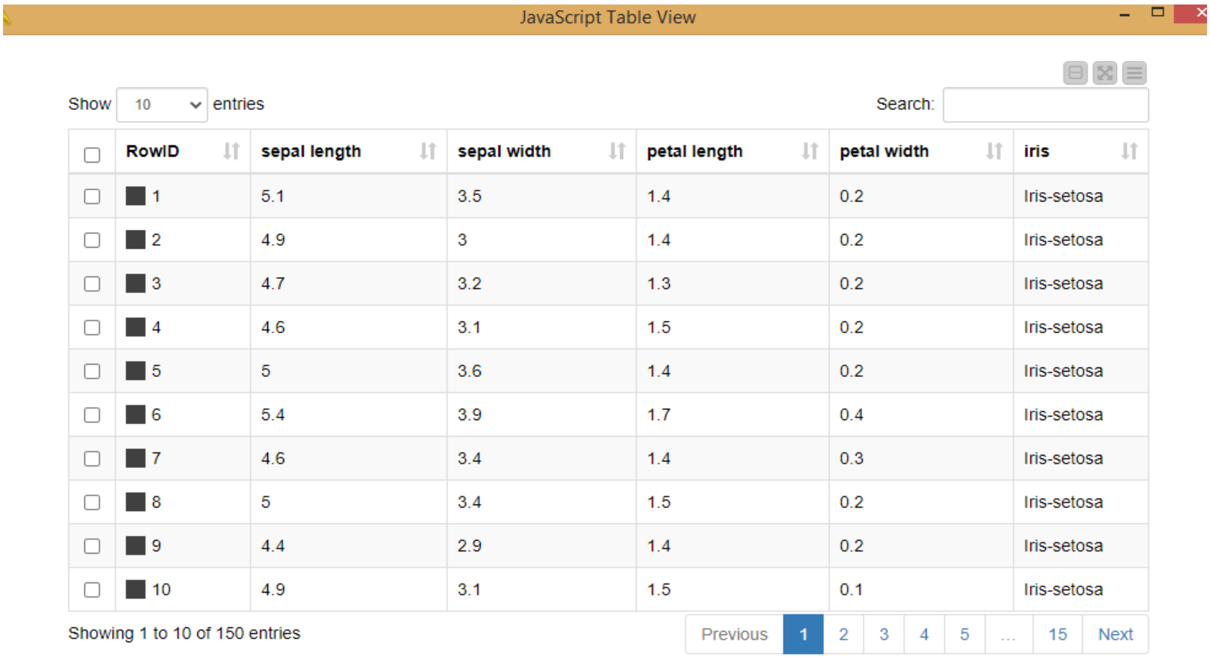
Only columns with a valid domain are available in this view. Make sure that the predecessor node is executed or set the domain with the DomainCalculator node!

KNIME Console

```
;V Reader          3:1      No settings available
;V Reader          3:1      Execute failed: Encountered duplicate row ID "1"
:cision Tree Learner 3:2     Guessing target column: "name".
;V Reader          4:1      No settings available
;V Reader          4:2      No settings available
```

Figure 3.5

The Table view and scatterplot of the executed workflow is given in the Figure 3.6 and Figure 3.7 respectively.



Show 10 entries Search:

□	RowID	sepal length	sepal width	petal length	petal width	iris
□	■ 1	5.1	3.5	1.4	0.2	Iris-setosa
□	■ 2	4.9	3	1.4	0.2	Iris-setosa
□	■ 3	4.7	3.2	1.3	0.2	Iris-setosa
□	■ 4	4.6	3.1	1.5	0.2	Iris-setosa
□	■ 5	5	3.6	1.4	0.2	Iris-setosa
□	■ 6	5.4	3.9	1.7	0.4	Iris-setosa
□	■ 7	4.6	3.4	1.4	0.3	Iris-setosa
□	■ 8	5	3.4	1.5	0.2	Iris-setosa
□	■ 9	4.4	2.9	1.4	0.2	Iris-setosa
□	■ 10	4.9	3.1	1.5	0.1	Iris-setosa

Showing 1 to 10 of 150 entries

Figure 3.6

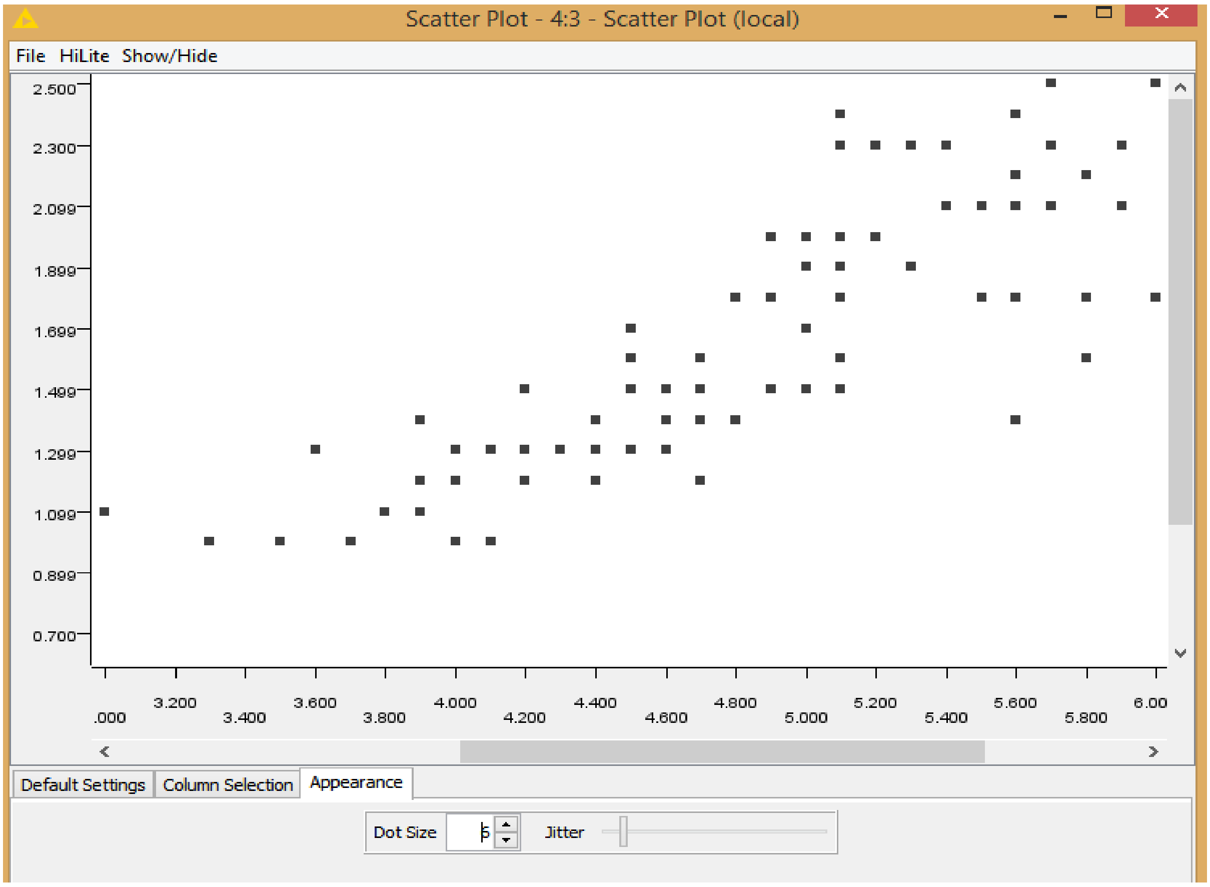
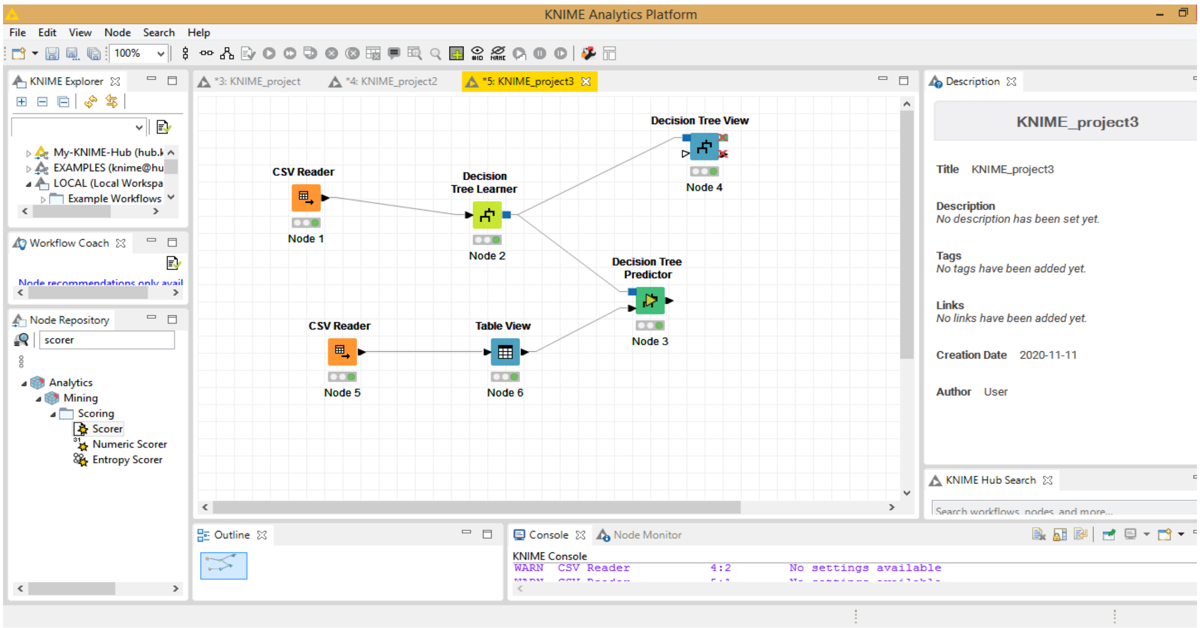


Figure 3.7

An example for prediction is shown below (Figure 3.8). Here a CSV Reader i.e., Node 1 is given the iris dataset as input (train data). It is then connected to the Decision Tree Learner (its output is given in Figure 3.9) where classification modelling is done. To get a more clear and simplified view of the output of the Decision Tree Learner, Decision Tree Viewer is used. The test data is given as input through a CSV Reader i.e., Node 5. It is connected to Table View where the data table can be viewed clearly. Both the train and test data are given to the Decision Tree Predictor in order to predict the type of iris plant species.



KNIME_project3

Title KNIME_project3

Description
No description has been set yet.

Tags
No tags have been added yet.

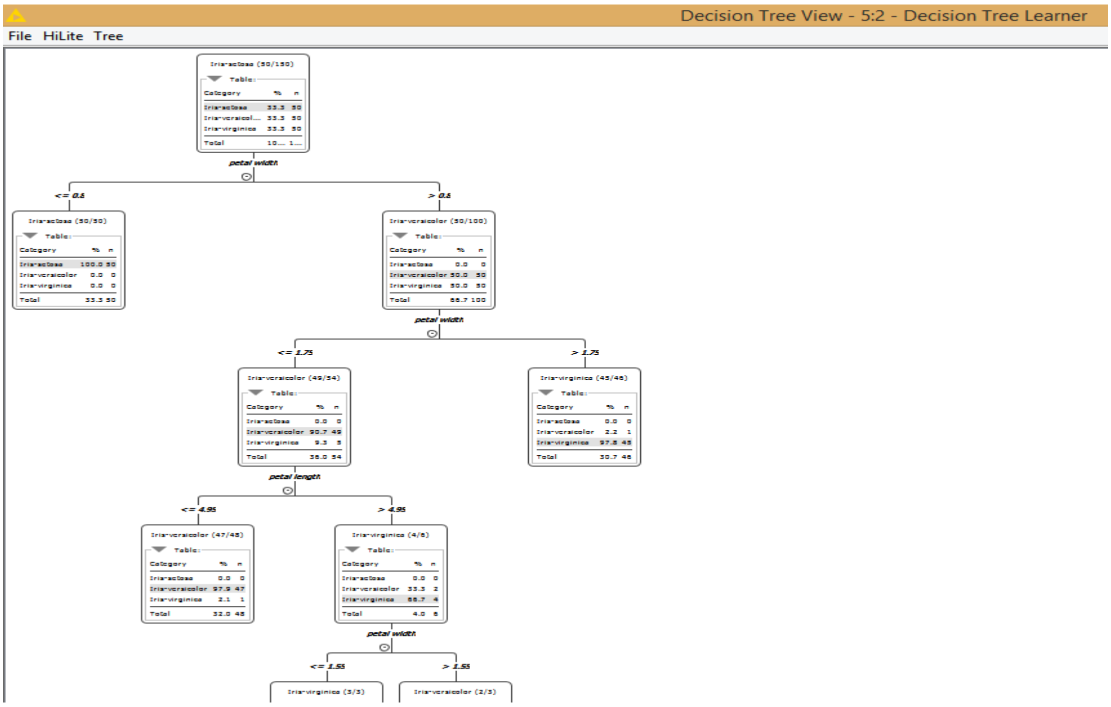
Links
No links have been added yet.

Creation Date 2020-11-11

Author User

```
KNIME Console
WARN  CSV Reader            4:2        No settings available
```

Figure 3.8



Iris-setosa (50/150)

Table:

Category	%	n
Iris-setosa	33.3	50
Iris-versicol...	33.3	50
Iris-virginica	33.3	50
Total	10...	1...

Iris-setosa (50/50)

Table:

Category	%	n
Iris-setosa	100.0	50
Iris-versicolor	0.0	0
Iris-virginica	0.0	0
Total	33.3	50

Iris-versicolor (50/100)

Table:

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	50.0	50
Iris-virginica	50.0	50
Total	66.7	100

Iris-versicolor (49/54)

Table:

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	90.7	49
Iris-virginica	9.3	5
Total	36.0	54

Iris-virginica (45/46)

Table:

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	2.2	1
Iris-virginica	97.8	45
Total	30.7	46

Iris-versicolor (47/48)

Table:

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	97.9	47
Iris-virginica	2.1	1
Total	32.0	48

Iris-virginica (4/6)

Table:

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	33.3	2
Iris-virginica	66.7	4
Total	4.0	6

Figure 3.9

The simplified view of Decision Tree Learner is given by Decision Tree Viewer(Figure 3.10).

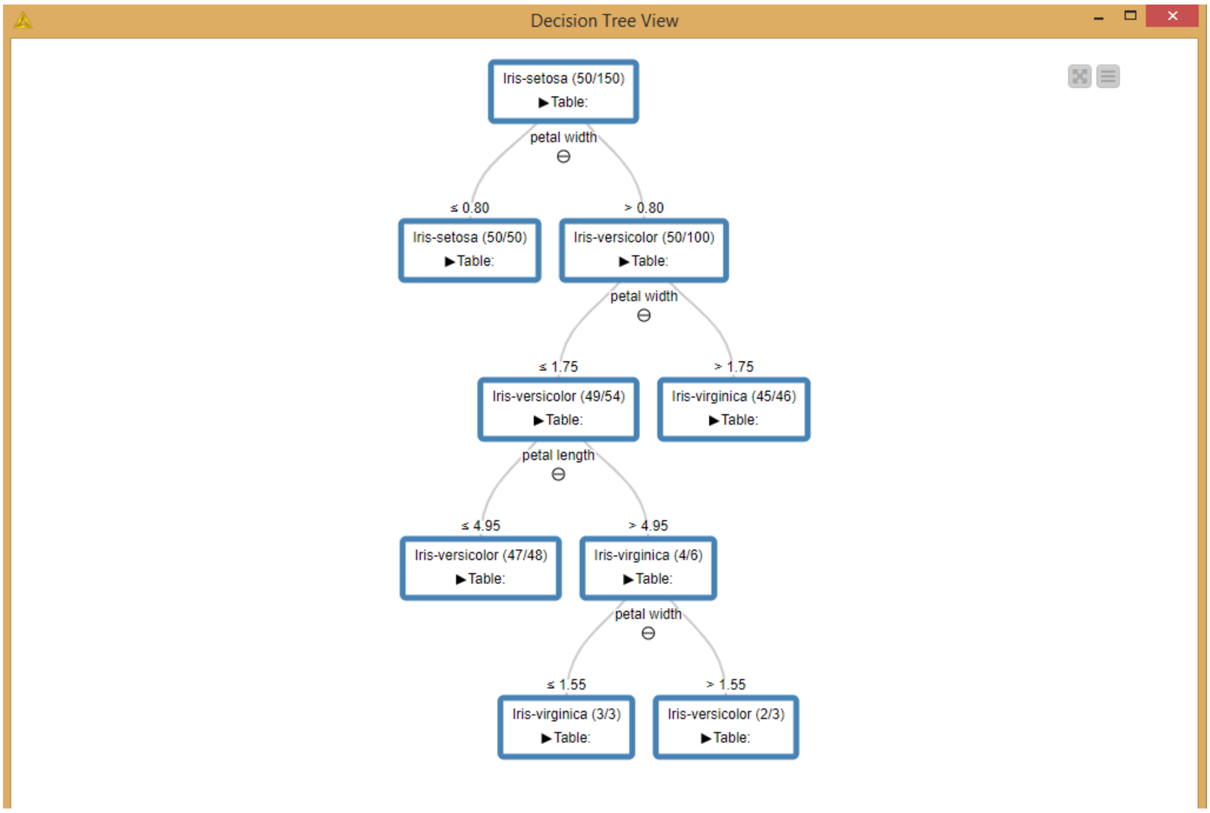
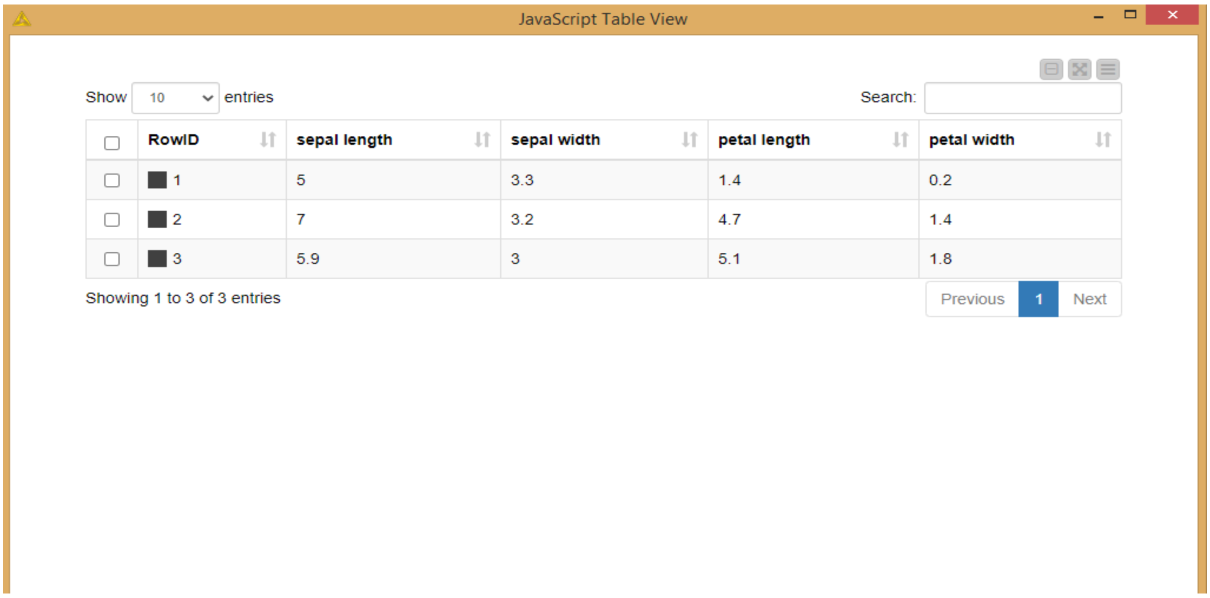


Figure 3.10

The Table View of the test data (with three instances) is given below (Figure 3.11).

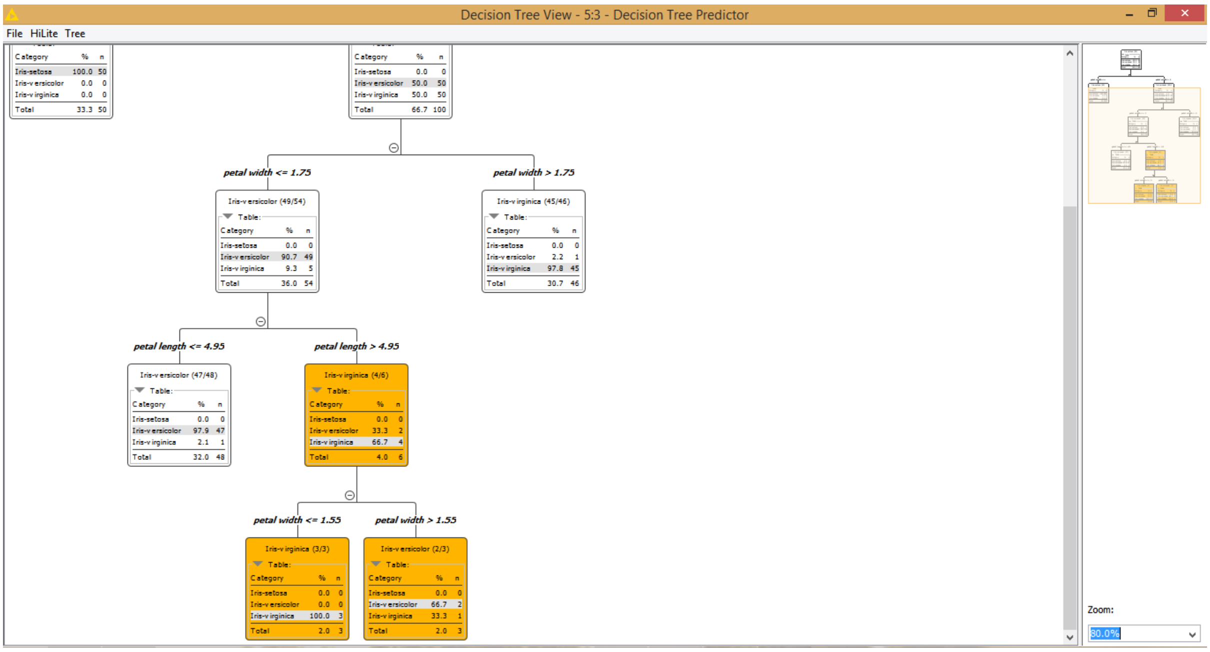


□	RowID	sepal length	sepal width	petal length	petal width
□	■ 1	5	3.3	1.4	0.2
□	■ 2	7	3.2	4.7	1.4
□	■ 3	5.9	3	5.1	1.8

Showing 1 to 3 of 3 entries

Figure 3.11

The output of Decision Tree Predictor is given in the Figure 3.12 where the type of three iris species are predicted. The highlighted sections are the predicted output based on the conditions from the trained dataset. Another thing about Knime is that it has separate predictors for each models.



Category	%	n
Iris-setosa	100.0	50
Iris-versicolor	0.0	0
Iris-virginica	0.0	0
Total	33.3	50

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	50.0	50
Iris-virginica	50.0	50
Total	66.7	100

Iris-versicolor (49/54)

Table:

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	90.7	49
Iris-virginica	9.3	5
Total	36.0	54

Iris-virginica (45/46)

Table:

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	2.2	1
Iris-virginica	97.8	45
Total	30.7	46

Iris-versicolor (47/48)

Table:

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	97.9	47
Iris-virginica	2.1	1
Total	32.0	48

Iris-virginica (4/6)

Table:

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	33.3	2
Iris-virginica	66.7	4
Total	4.0	6

Iris-virginica (3/3)

Table:

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	0.0	0
Iris-virginica	100.0	3
Total	2.0	3

Iris-versicolor (2/3)

Table:

Category	%	n
Iris-setosa	0.0	0
Iris-versicolor	66.7	2
Iris-virginica	33.3	1
Total	2.0	3

Figure 3.12

My experience with using Knime was not good. It takes a lot of time to launch the software and it slowed down all other applications. It was not user-friendly for me as I did take a lot of time to get to know about the software. After watching some tutorials and trying out few examples, I was able to do small basic examples. Knime is difficult for a beginner but for experts, I think this tool might be handy them to extend functionalities and build advanced models using Java, R and Python. It does not have a good user interface which made me spend a lot of time to get a general basic idea. After doing few examples, I was able to try out different nodes and got to know about a general idea of what the tool is and why it is used.

REFERENCES

1. File — Orange Visual Programming 3 documentation [WWW Document], n.d. URL https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/data/file.html (accessed 11.11.20).
2. Getting Started with Orange 01: Welcome to Orange, 2015. https://www.youtube.com/watch?v=HXjnDIgGDuI&list=PLmNPvQr9Tf-ZSDLwOzxpvY-HrE0yv-8Fy&index=1
3. Foong, N.W., 2020. Data Science Made Easy: Data Modeling and Prediction using Orange [WWW Document]. Medium. URL https://towardsdatascience.com/data-science-made-easy-data-modeling-and-prediction-using-orange-f451f17061fa (accessed 11.11.20).
4. Predictive Analytics Software | RapidMiner Studio [WWW Document], n.d. . RapidMiner. URL https://rapidminer.com/products/studio/ (accessed 11.11.20).
5. URL https://docs.rapidminer.com/latest/studio/getting-started/ui-overview.html (accessed 11.11.20).
6. Intro to the RapidMiner Studio GUI | RapidMiner, 2019. https://www.youtube.com/watch?v=Gg01mmR3-g&list=PLssWC2d9JhOZLbQNZ80uOxLypglgWqbJA
7. KNIME, 2020. Wikipedia.https://en.wikipedia.org/wiki/KNIME
8. KNIME User Interface Walkthrough | KNIME Tutorial, 2015. https://www.youtube.com/watch?v=W0L8Feq51kE