Building Machine Learning Model is fun using Orange

Analytics Vidhya Content Team September 7, 2017

Introduction

With growing need of data science managers, we need tools which take out difficulty from doing data science and make it fun. Not everyone is willing to learn coding, even though they would want to learn / apply data science. This is where GUI based tools can come in handy.

Today, I will introduce you to another GUI based tool – Orange. This tool is great for beginners who wish to visualize patterns and understand their data without really knowing how to code.

In my previous article, I presented you with another GUI based tool KNIME. If you do not want to learn to code but still apply data science, you can try out any of these tools.

By the end of this tutorial, you'll be able to predict which person out of a certain set of people is eligible for a loan with Orange!

Table of Contents:

1. Why Orange?
2. Setting up your System:
3. Creating your first Workflow
4. Familiarizing yourself with the basics
    1. Problem Statement
    2. Importing the data files
    3. Understanding the data
5. How do you clean your data?
6. Training your first model

1. Why Orange?

Orange is a platform built for mining and analysis on a GUI based workflow. This signifies that you do not have to know how to code to be able to work using Orange and mine data, crunch numbers and derive insights.

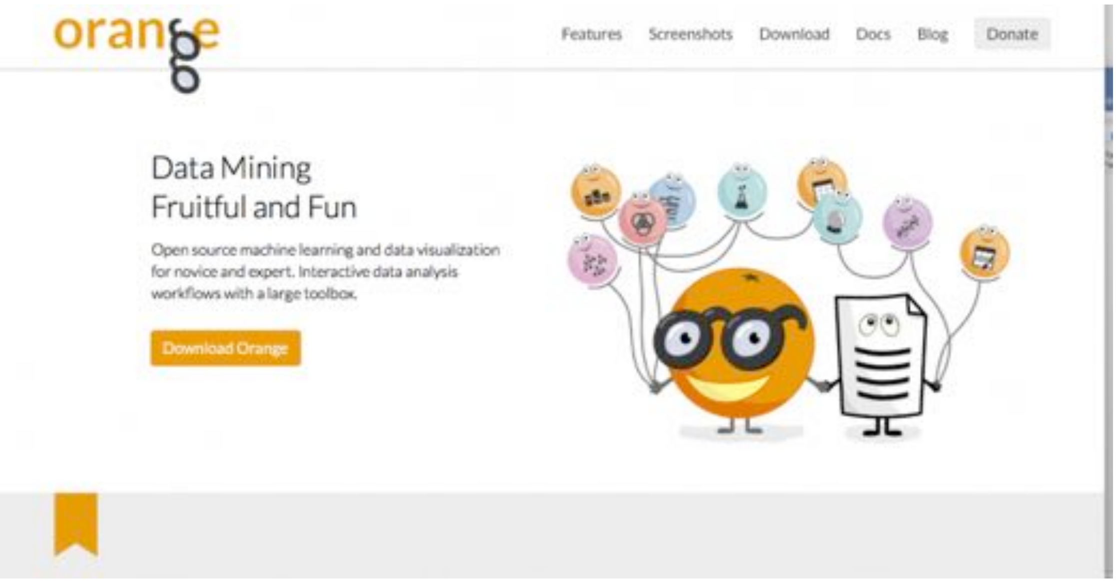
You can perform tasks ranging from basic visuals to data manipulations, transformations, and data mining. It consolidates all the functions of the entire process into a single workflow.

The best part and the differentiator about Orange is that it has some wonderful visuals. You can try silhouettes, heat-maps, geo-maps and all sorts of visualizations available.

2. Setting up your System

Orange comes built-in with the Anaconda tool if you’ve previously installed it. If not, follow these steps to download Orange.

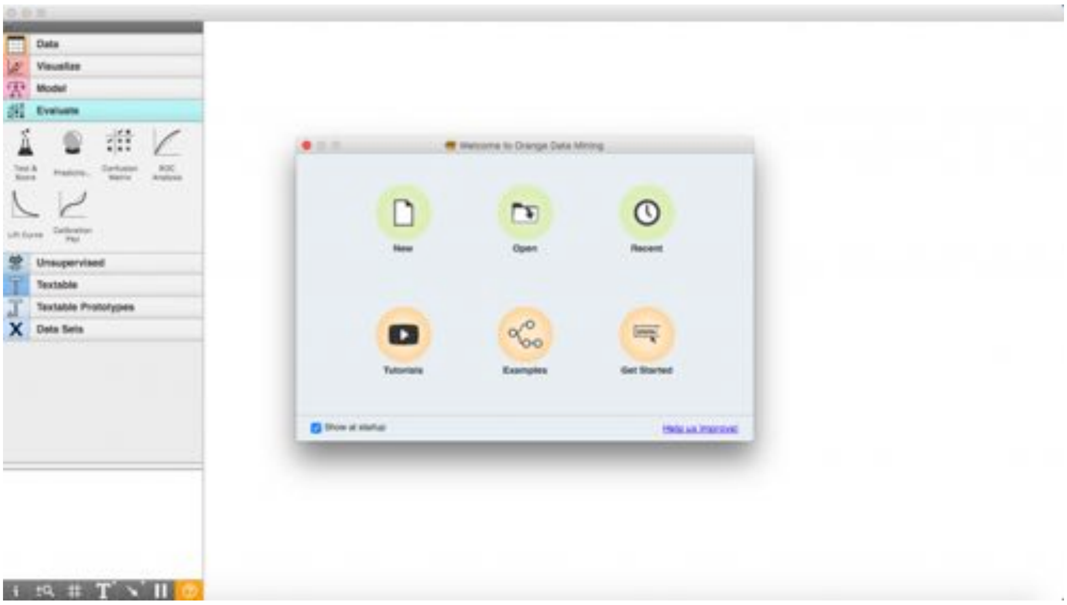
Step 1: Go to https://orange.biolab.si and click on Download.



Data Mining
Fruitful and Fun

Open source machine learning and data visualization for novice and expert. Interactive data analysis workflows with a large toolbox.

Step 2: Install the platform and set the working directory for Orange to store its files.



This is what the start-up page of Orange looks like. You have options that allow you to create new projects, open recent ones or view examples and get started.

Before we delve into how Orange works, let's define a few key terms to help us in our understanding:

1. **A widget** is the basic processing point of any data manipulation. It can do a number of actions based on what you choose in your widget selector on the left of the screen.
2. **A workflow** is the sequence of steps or actions that you take in your

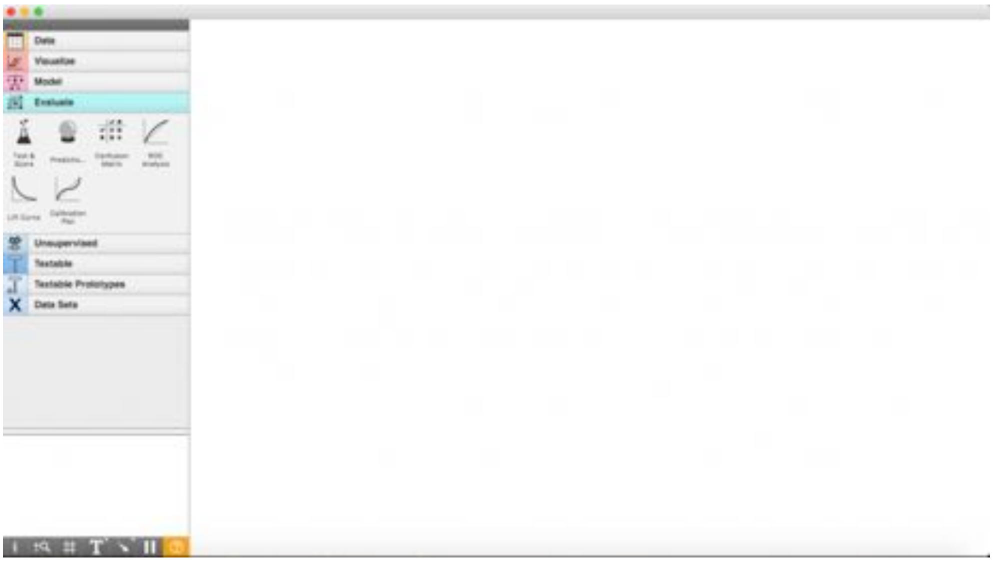
platform to accomplish a particular task.

You can also go to “Example Workflows” on your start-up screen to check out more workflows once you have created your first one.

For now, click on “New” and let’s start building your first workflow.

3. Creating Your First Workflow

This is the first step towards building a solution to any problem. We need to first understand what steps we need to take in order to achieve our final goal. After you clicked on “New” in the above step, this is what you should have come up with.



This is your blank Workflow on Orange. Now, you’re ready to explore and solve any problem by dragging any widget from the widget menu to your workflow.

4. Familiarising yourself with the basics

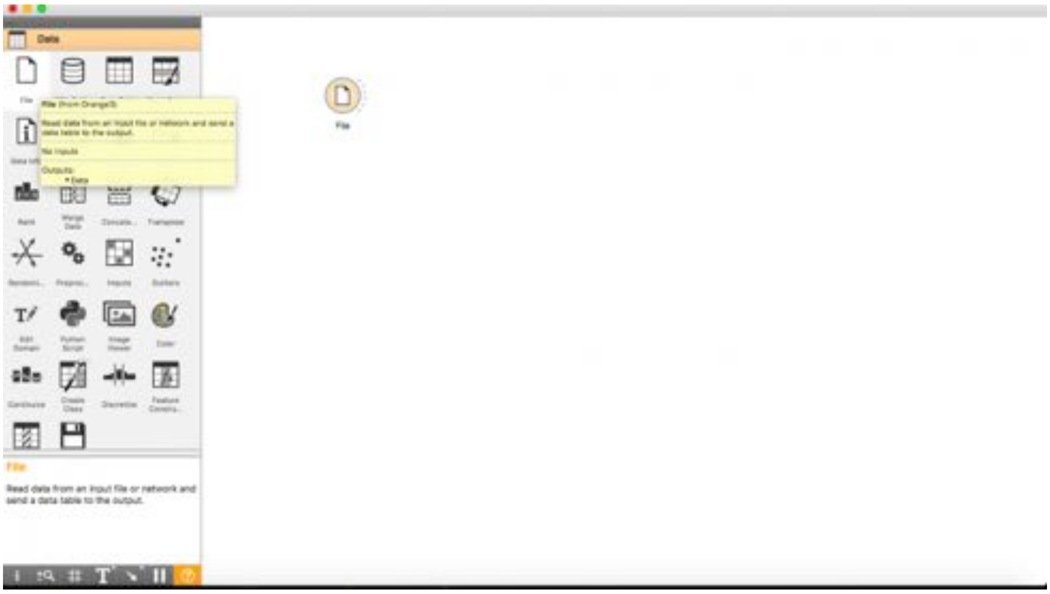
Orange is a platform that can help us solve most problems in Data Science today. Topics that range from the most basic visualizations to training models. You can even evaluate and perform unsupervised learning on datasets:

4.1 Problem

The problem we're looking to solve in this tutorial is the practice problem **Loan Prediction** that can be accessed via this link on **Datahack**.

4.2 Importing the data files

We begin with the first and the necessary step to understand our data and make predictions: **importing our data**



File (from Orange3)

Read data from an input file or network and send a data table to the output.

No inputs

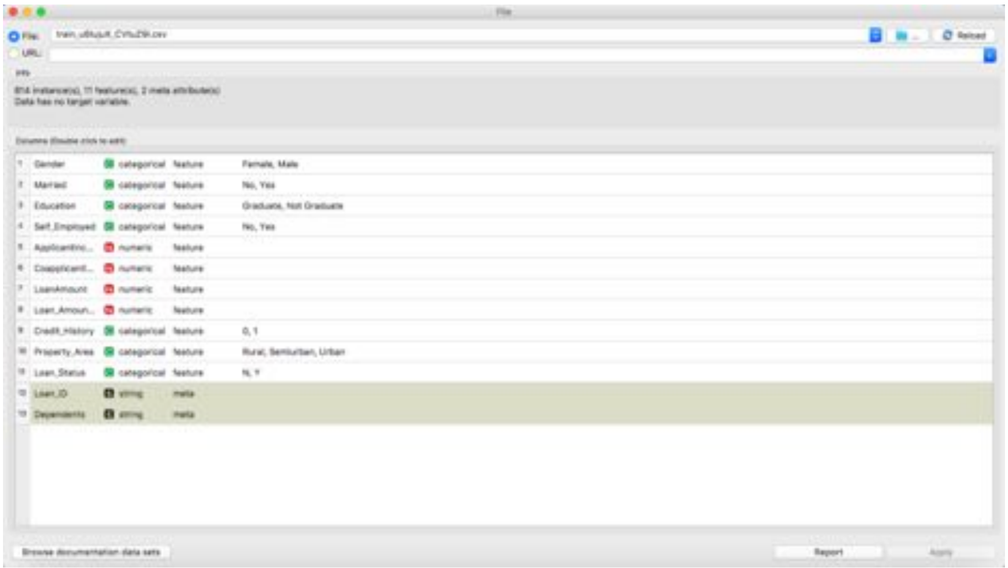
Outputs:
• Data

File

Read data from an input file or network and send a data table to the output.

Step 1: Click on the "Data" tab on the widget selector menu and drag the widget "File" to our blank workflow.

Step 2: Double click the "File" widget and select the file you want to load into the workflow. In this article, as we will be learning how to solve the practice problem Loan Prediction, I will import the training dataset from the same.



File: train_u6lujuX_CVtuZ9i.csv

URL:

Info

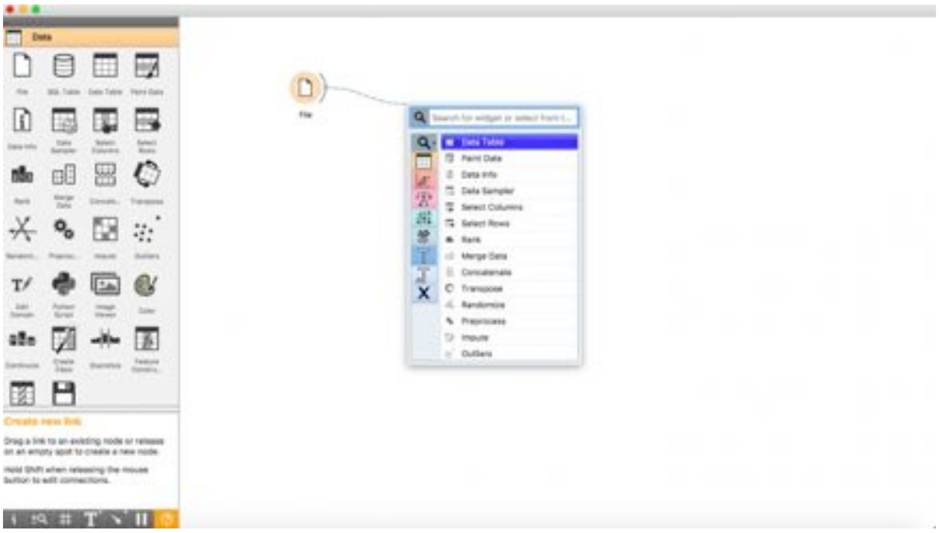
614 instance(s), 11 feature(s), 2 meta attribute(s)
Data has no target variable.

Columns (Double click to edit)

1	Gender	categorical	feature	Female, Male
2	Married	categorical	feature	No, Yes
3	Education	categorical	feature	Graduate, Not Graduate
4	Self_Employed	categorical	feature	No, Yes
5	ApplicantInc...	numeric	feature	
6	CoapplicantI...	numeric	feature	
7	LoanAmount	numeric	feature	
8	Loan_Amoun...	numeric	feature	
9	Credit_History	categorical	feature	0, 1
10	Property_Area	categorical	feature	Rural, Semiurban, Urban
11	Loan_Status	categorical	feature	N, Y
12	Loan_ID	string	meta	
13	Dependents	string	meta	

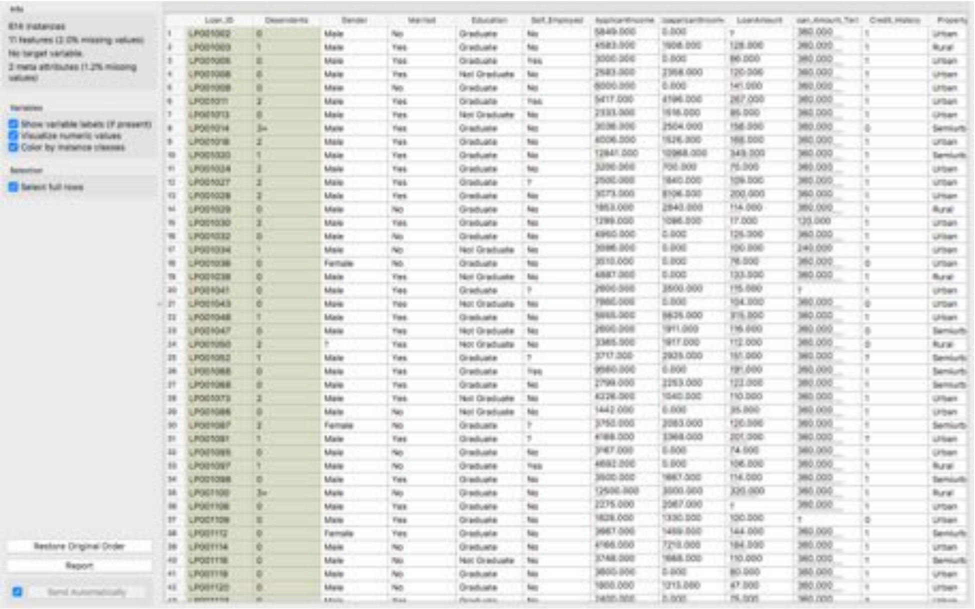
Step 3: Once you can see the structure of your dataset using the widget, go back by closing this menu.

Step 4: Now since we have the raw .csv details, we need to convert it to a format we can use in our mining. Click on the dotted line encircling the “File” widget and drag, and then click anywhere in the blank space.



Step 5: As we need a data table to better visualize our findings, we click on the “Data Table” widget.

Step 6: Now double click the widget to visualize your table.



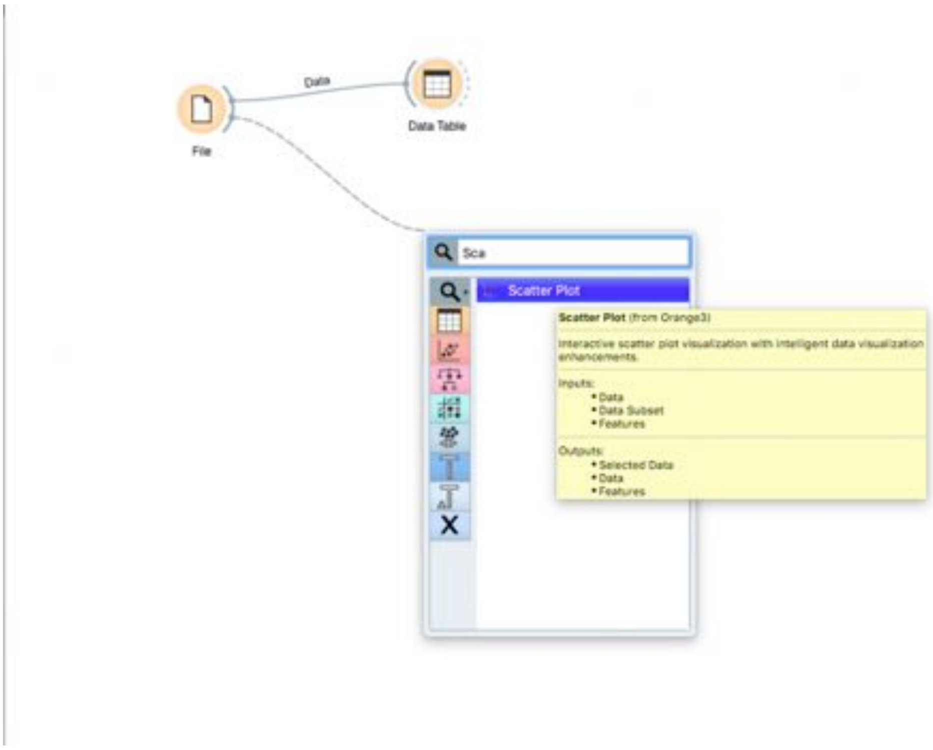
Neat! Isn't it?

Let's now visualize some columns to find interesting patterns in our data.

4.3 Understanding our Data

4.3.1 Scatter Plot

Click on the semicircle in front of the "File" widget and drag it to an empty space in the workflow and select the "Scatter Plot" widget.



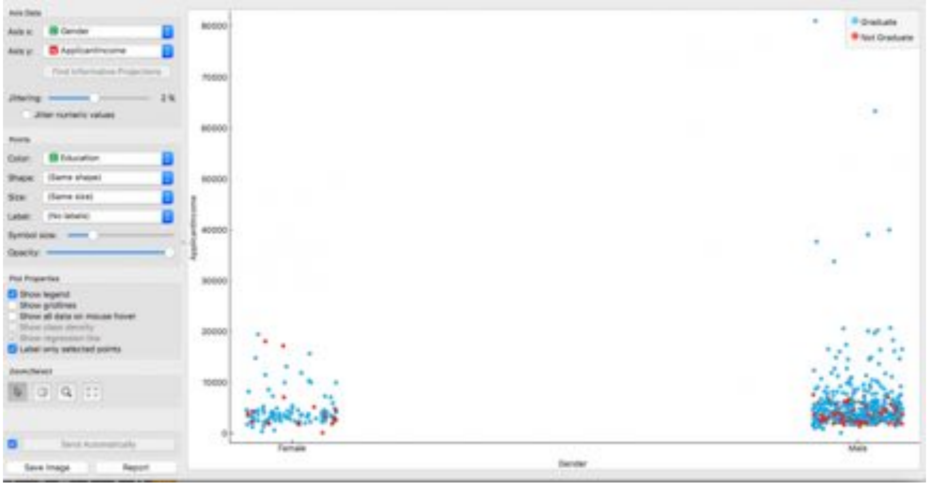
Scatter Plot (from Orange3)

Interactive scatter plot visualization with intelligent data visualization enhancements.

Inputs:
- Data
- Data Subset
- Features

Outputs:
- Selected Data
- Data
- Features

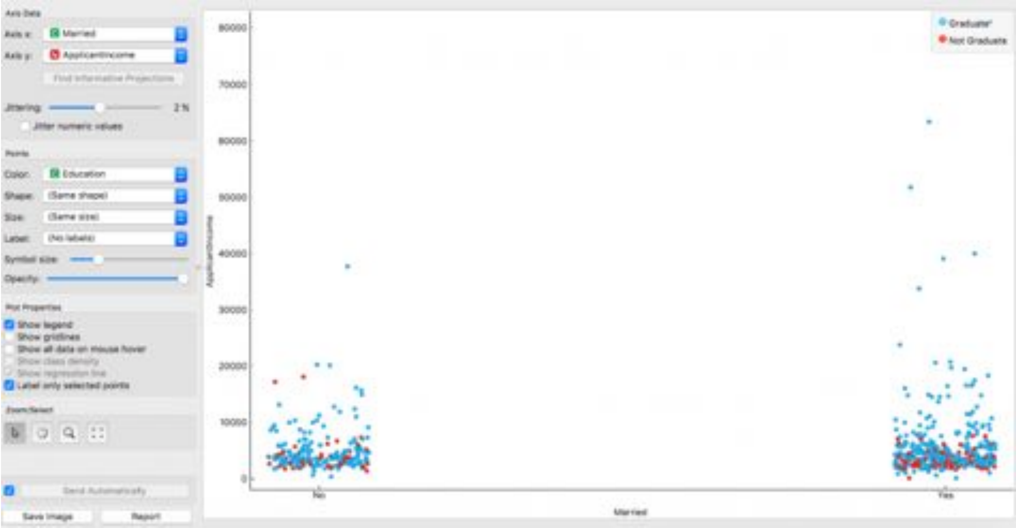
Once you create a Scatter Plot widget, double click it and explore your data like this! You can select the X and Y axes, colors, shapes, sizes and a lot of other manipulations.



Axis Data
Axis x: Gender
Axis y: ApplicantIncome
Jittering: 2 %
○ Jitter numeric values
Points
Color: Education
Shape: (Same shape)
Size: (Same size)
Label: (No labels)
Plot Properties
☑ Show legend
☐ Show gridlines
☐ Show all data on mouse hover
☐ Show class density
☐ Show regression line
☑ Label only selected points
☑ Send Automatically

The plot I've explored is a Gender by Income plot, with the colors set to the education levels. As we can see in males, the higher income group naturally belongs to the Graduates!

Although in females, we see that a lot of the graduate females are earning low or almost nothing at all. Any specific reason? Let's find out using the scatterplot.

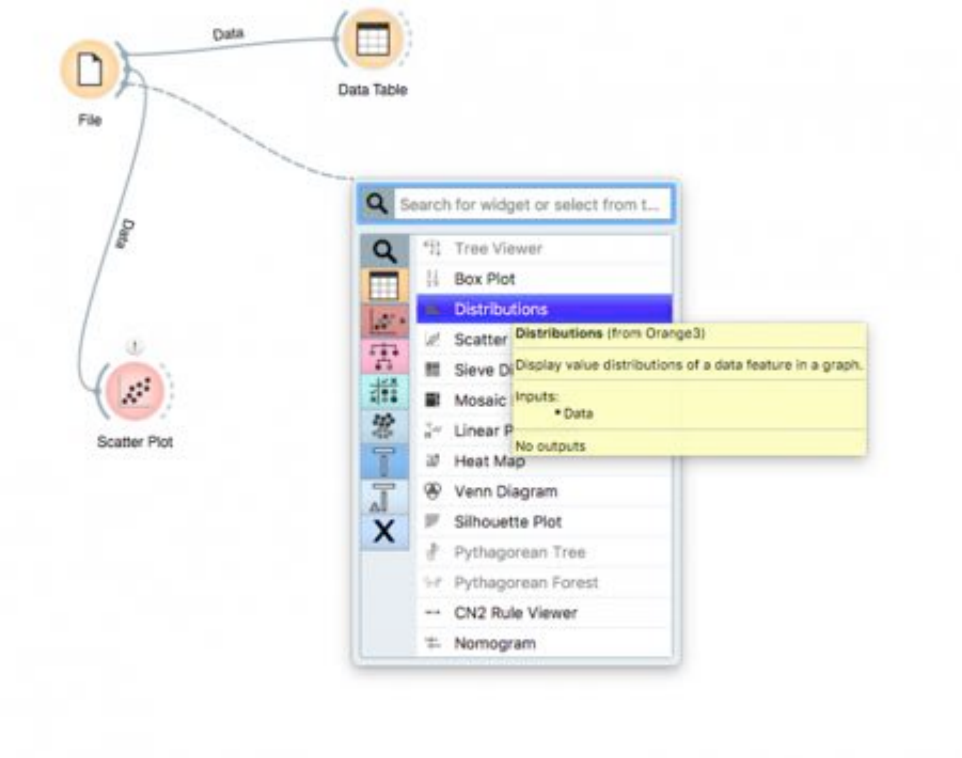


Axis Data
Axis x: Married
Axis y: ApplicantIncome
Jittering: 2 %
○ Jitter numeric values
Points
Color: Education
Shape: (Same shape)
Size: (Same size)
Label: (No labels)
Plot Properties
☑ Show legend
☐ Show gridlines
☐ Show all data on mouse hover
☐ Show class density
☐ Show regression line
☑ Label only selected points
☑ Send Automatically

One possible reason I found was marriage. A huge number graduates who were married were found to be in lower income groups; this may be due to family responsibilities or added efforts. Makes perfect sense, right?

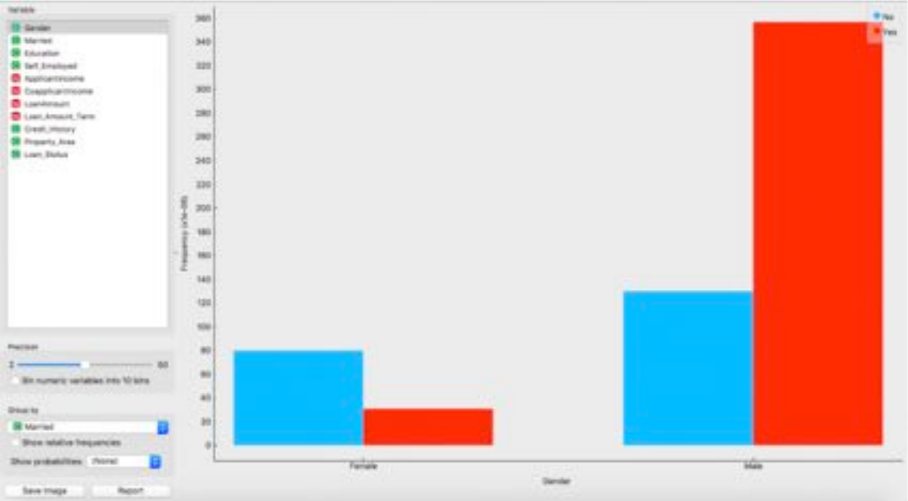
4.3.2 Distribution

Another way to visualize our distributions would be the "Distributions" widget. Click on the semi-circle again, and drag to find the widget "Distributions".



Distributions (from Orange3)
Display value distributions of a data feature in a graph.
Inputs:
• Data
No outputs

Now double click on it and visualize!



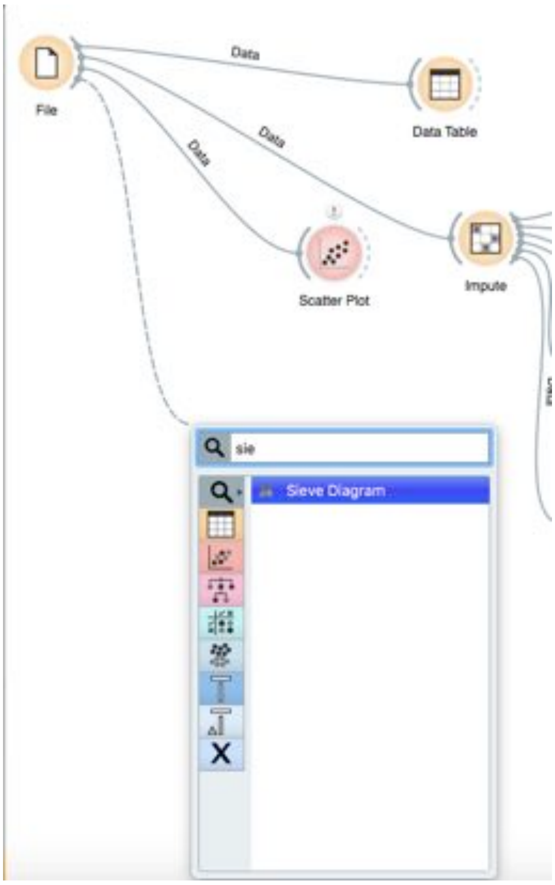
What we see is a very interesting distribution. We have in our dataset, more number of married males than females.

4.3.3 Sieve diagram

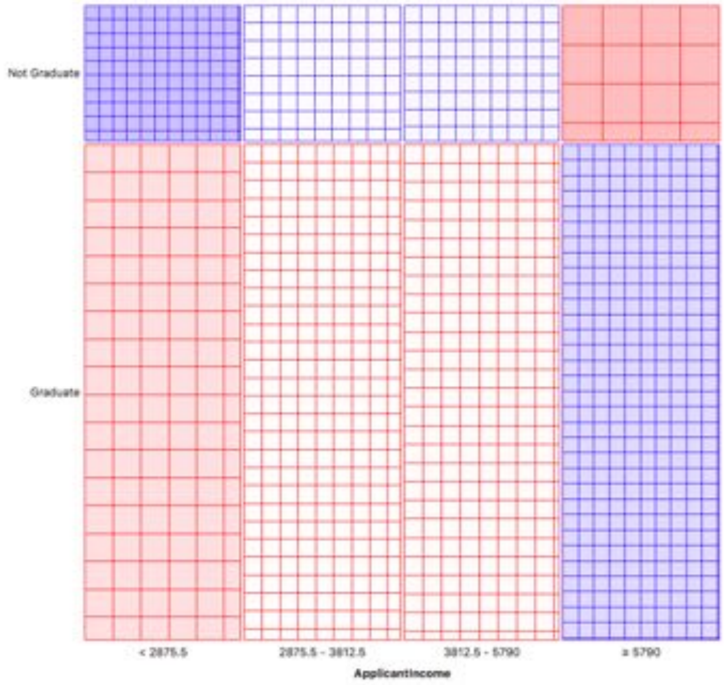
How does income relate to the education levels? Do graduates get paid more than non-grads?

Let's visualize using a sieve diagram.

Click and drag from the “File” widget and search for “Sieve Diagram”.



Once you place it, double click on it and select your axes!



This plot divides the sections of distribution into 4 bins. The sections can be investigated by hovering the mouse over it.

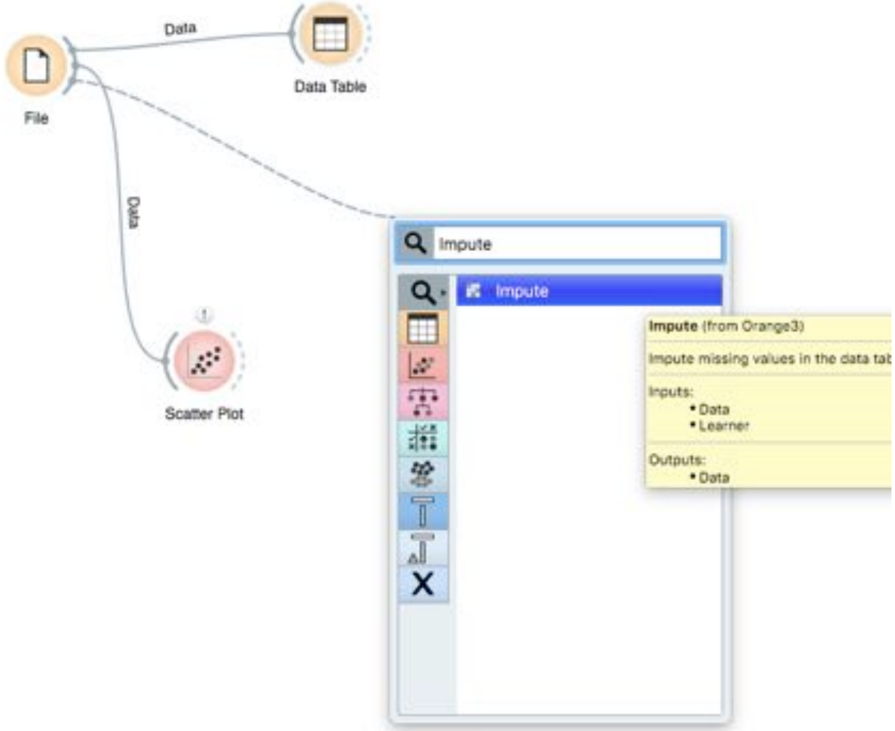
For example, graduates and non-graduates are divided 78% by 22%. Then subdivisions of 25% each are made by splitting the applicant incomes into 4 equal groups. Here the task for you, generate insight from these charts and share in the comment section.

Let’s now look at how to clean our data to start building our model.

5. How do you clean your data?

Here for cleaning purpose, we will impute missing values. Imputation is a very important step in understanding and making the best use of our data.

Click on the “File” widget and drag to find the “Impute” widget.



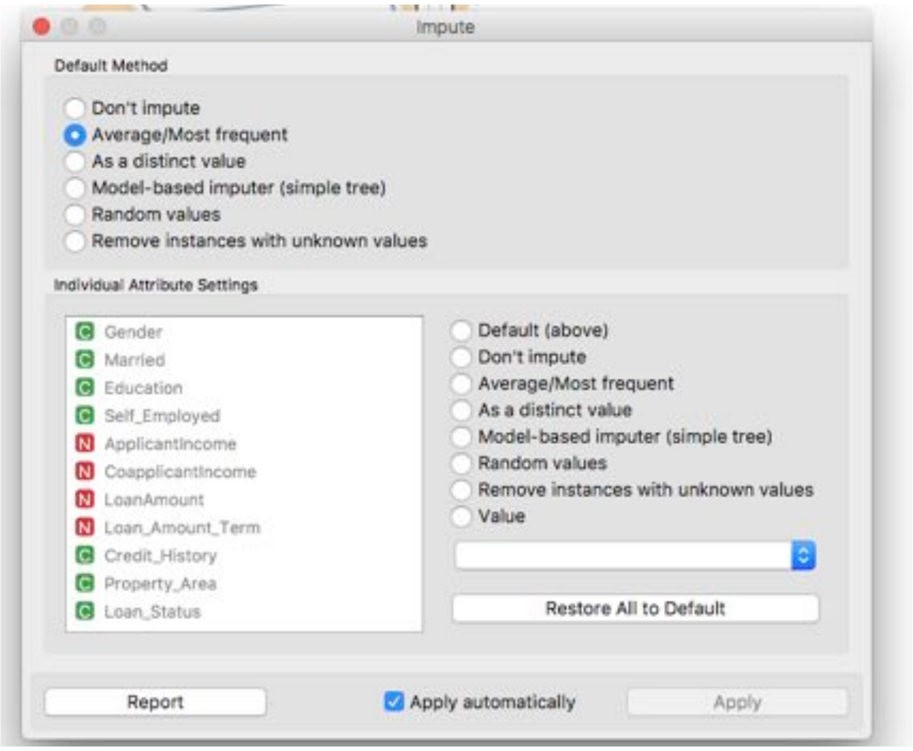
Impute (from Orange3)

Impute missing values in the data tab

Inputs:
- Data
- Learner

Outputs:
- Data

When you double click on the widget after placing it, you will see that there are a variety of imputation methods you can use. You can also use default methods or choose individual methods for each class separately.



Impute

Default Method

○ Don't impute
● Average/Most frequent
○ As a distinct value
○ Model-based imputer (simple tree)
○ Random values
○ Remove instances with unknown values

Individual Attribute Settings

C Gender
C Married
C Education
C Self_Employed
N ApplicantIncome
N CoapplicantIncome
N LoanAmount
N Loan_Amount_Term
C Credit_History
C Property_Area
C Loan_Status

○ Default (above)
○ Don't impute
○ Average/Most frequent
○ As a distinct value
○ Model-based imputer (simple tree)
○ Random values
○ Remove instances with unknown values
○ Value

☑ Apply automatically

Here, I have selected the default method to be Average for numerical values and Most Frequent for text based values (categorical).

You can select from a variety of imputations like:

1. Distinct Value
2. Random Values
3. Remove the rows with missing values
4. Model-Based

The other things you can include in your approach to training your model are Feature Extraction and Generation.For further understanding, follow this article on Data Exploration and Feature Engineering (https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/)

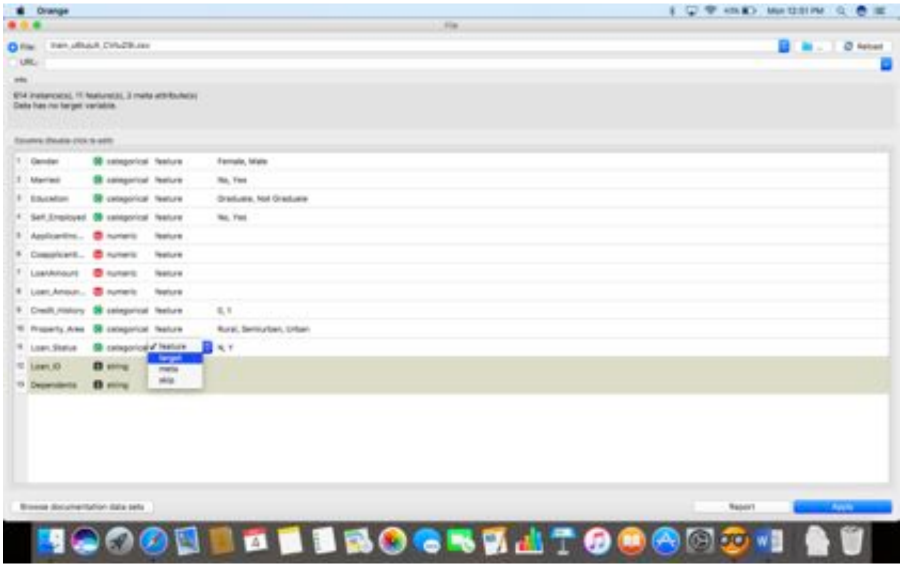
6. Training your First Model

Beginning with the basics, we will first train a linear model encompassing all the features just to understand how to select and build models.

Step 1: First, we need to set a target variable to apply Logistic Regression on it.

Step 2: Go to the “File” widget and double click it.

Step 3: Now, double click on the **Loan_Status** column and select it as the target variable. Click Apply.

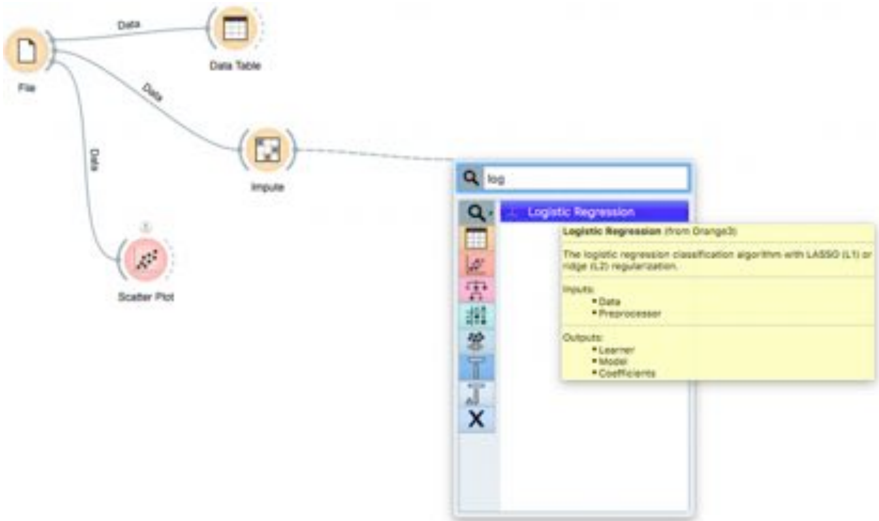

Orange
File
File: train_u6lujuX_CVtuZ9i.csv
Reload
URL:
Info
Browse documentation data sets
Report
Apply

614 instance(s), 11 feature(s), 2 meta attribute(s)
Data has no target variable.

Columns (Double click to edit)

1	Gender	categorical	feature	Female, Male
2	Married	categorical	feature	No, Yes
3	Education	categorical	feature	Graduate, Not Graduate
4	Self_Employed	categorical	feature	No, Yes
5	ApplicantIn...	numeric	feature	
6	Coapplicantl...	numeric	feature	
7	LoanAmount	numeric	feature	
8	Loan_Amoun...	numeric	feature	
9	Credit_History	categorical	feature	0, 1
10	Property_Area	categorical	feature	Rural, Semiurban, Urban
11	Loan_Status	categorical	✓ feature target meta skip	N, Y
12	Loan_ID	string		
13	Dependents	string		

Step 4: Once we have set our target variable, find the clean data from the “Impute” widget as follows and place the “Logistic Regression” widget.



Logistic Regression (from Orange3)

The logistic regression classification algorithm with LASSO (L1) or ridge (L2) regularization.

Inputs:
- Data
- Preprocessor

Outputs:
- Learner
- Model
- Coefficients

Step 5: Double click the widget and select the type of regularization you want to perform.

1. **Ridge Regression:**
   - Performs L2 regularization, i.e. adds penalty equivalent to **square of the magnitude** of coefficients
   - Minimization objective = LS Obj + α * (sum of square of

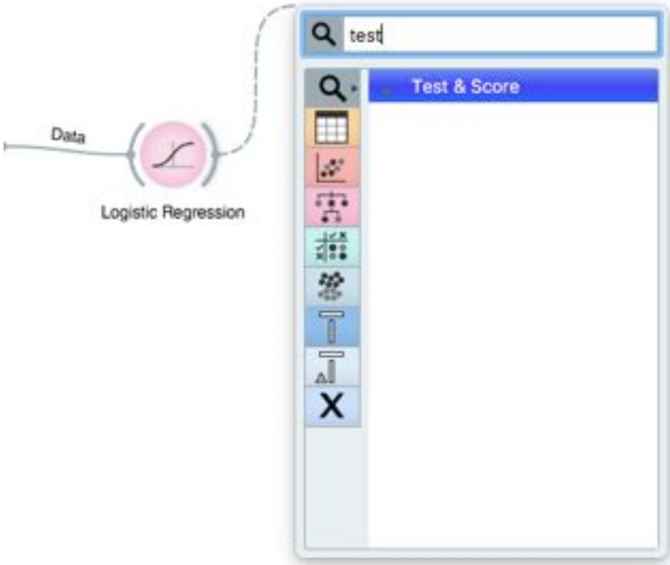
coefficients)

2. **Lasso Regression:**
   - Performs L1 regularization, i.e. adds penalty equivalent to **absolute value of the magnitude** of coefficients
   - Minimization objective = LS Obj + α * (sum of absolute value of coefficients)

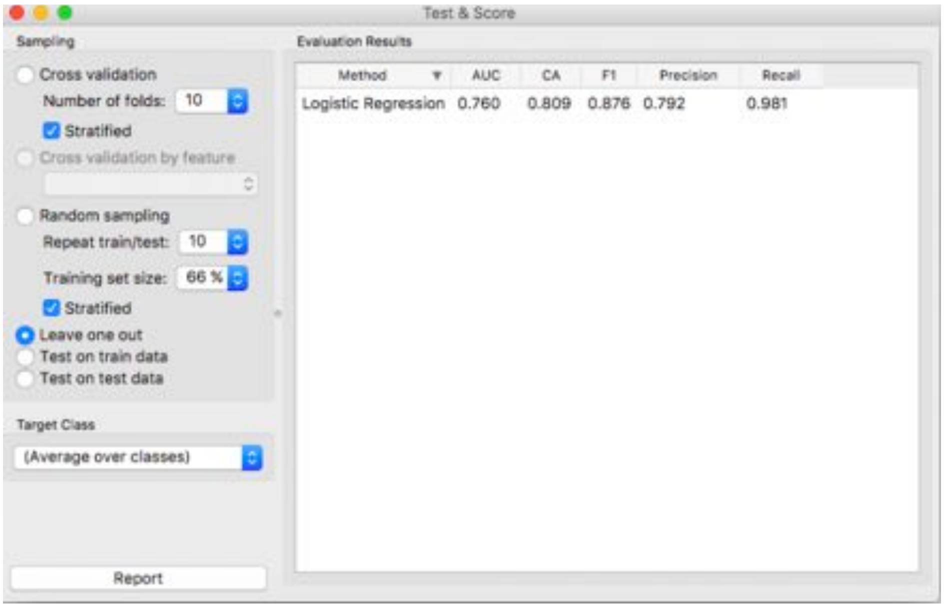
For a better understanding of these, please visit the link about Ridge and Lasso regressions https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-ridge-lasso-regression-python/

I have chosen Ridge for my analysis, you are free to choose between the two.

Step 6: Next, click on the "Impute" or the "Logistic Regression" widget and find the "Test and Score" widget. Make sure you connect both the **data and the model to the testing widget.**



Step 7: Now, click on the "Test and Score" widget to see how well your model is doing.



Test & Score

Sampling

○ Cross validation

Number of folds: 10

☑ Stratified

○ Cross validation by feature

○ Random sampling

Repeat train/test: 10

Training set size: 66 %

☑ Stratified

● Leave one out

○ Test on train data

○ Test on test data

Target Class

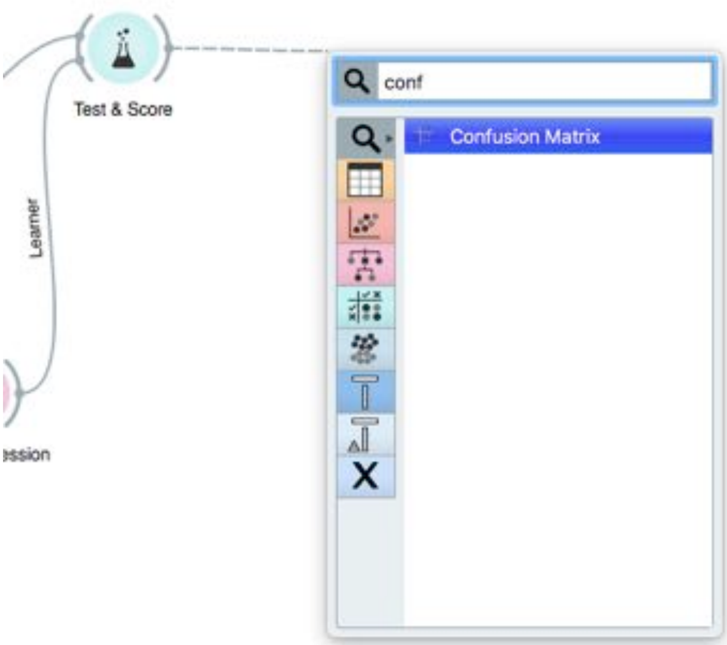
(Average over classes)

Report

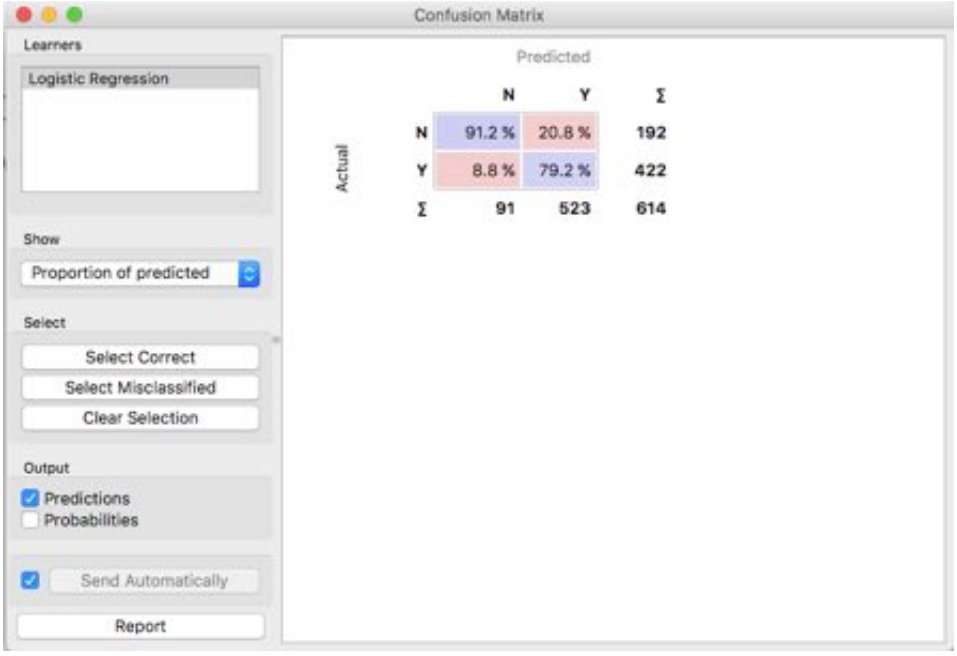
Evaluation Results

Method	AUC	CA	F1	Precision	Recall
Logistic Regression	0.760	0.809	0.876	0.792	0.981

Step 8: To visualize the results better, drag and drop from the “Test and Score” widget to fin d “Confusion Matrix”.



Step 9: Once you’ve placed it, click on it to visualize your findings!



Confusion Matrix

Learners

Logistic Regression

Show

Proportion of predicted

		Predicted		
		N	Y	Σ
Actual	N	91.2 %	20.8 %	192
	Y	8.8 %	79.2 %	422
	Σ	91	523	614

Output

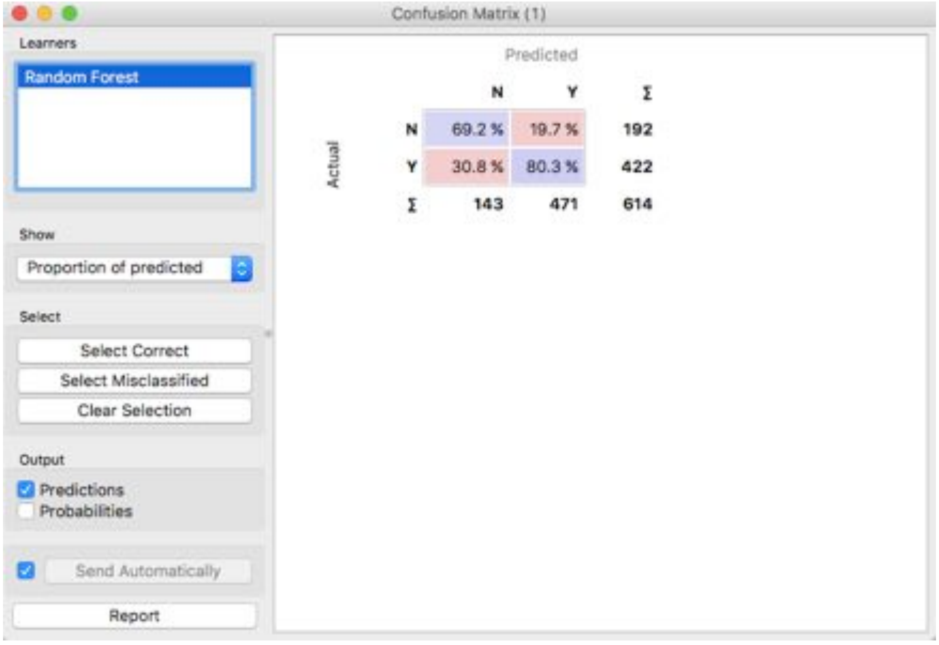
☑ Predictions

□ Probabilities

☑ Send Automatically

This way, you can test out different models and see how accurately they perform.

Let’s try to evaluate, how a Random Forest would do? Change the modeling method to Random Forest and look at the confusion matrix.



Confusion Matrix (1)

Learners

Random Forest

Show

Proportion of predicted

		Predicted		
		N	Y	Σ
Actual	N	69.2 %	19.7 %	192
	Y	30.8 %	80.3 %	422
	Σ	143	471	614

Output

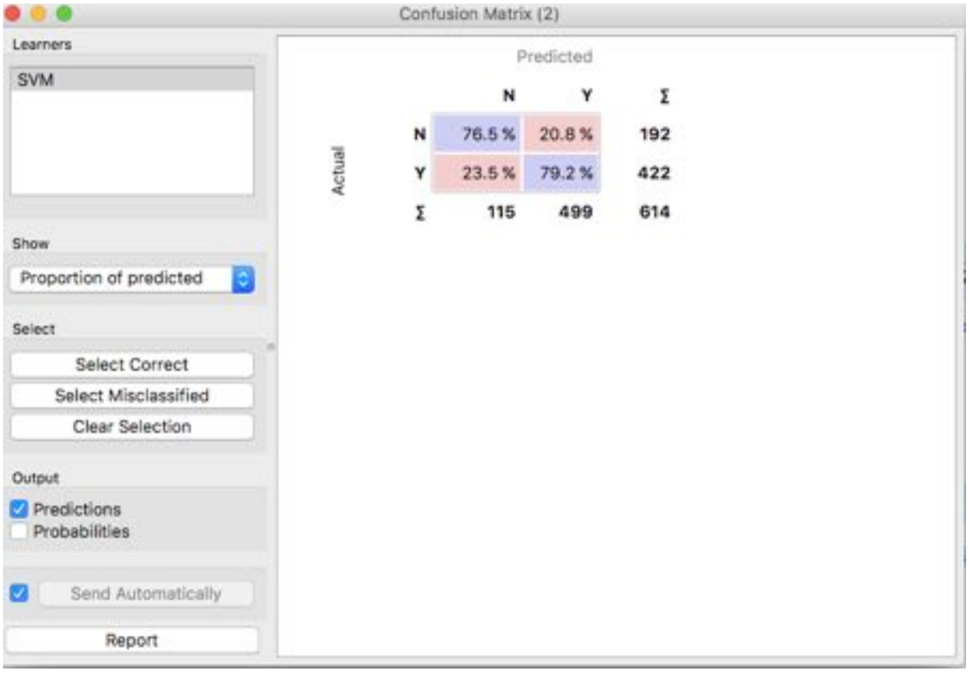
☑ Predictions

□ Probabilities

☑ Send Automatically

Looks decent, but the Logistic Regression performed better.

We can try again with a Support Vector Machine.



Show: Proportion of predicted

☑ Predictions
☐ Probabilities

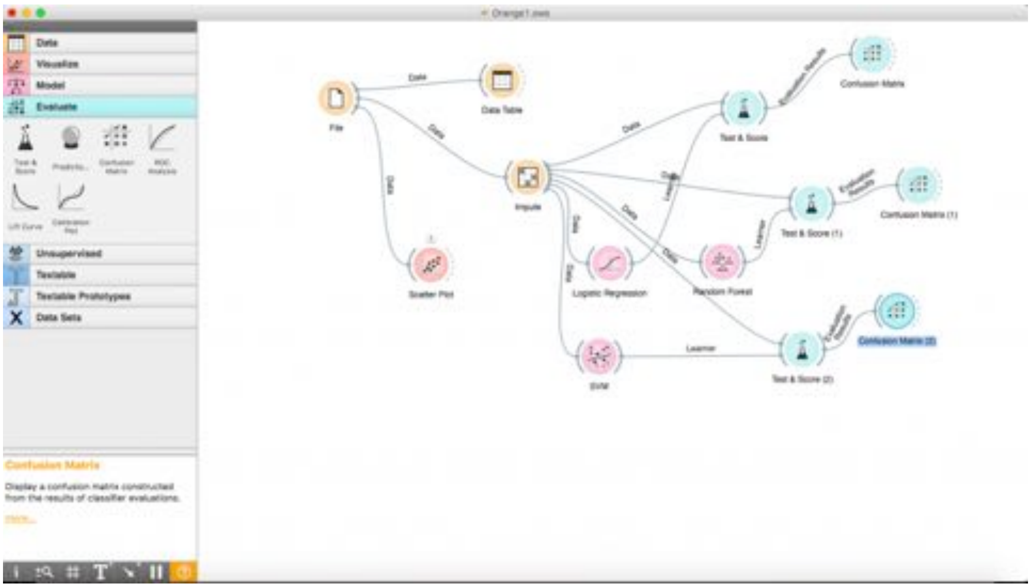
☑ Send Automatically

	N	Y	Σ
N	76.5 %	20.8 %	192
Y	23.5 %	79.2 %	422
Σ	115	499	614

Better than the Random Forest, but still not as good as the Logistic Regression model.

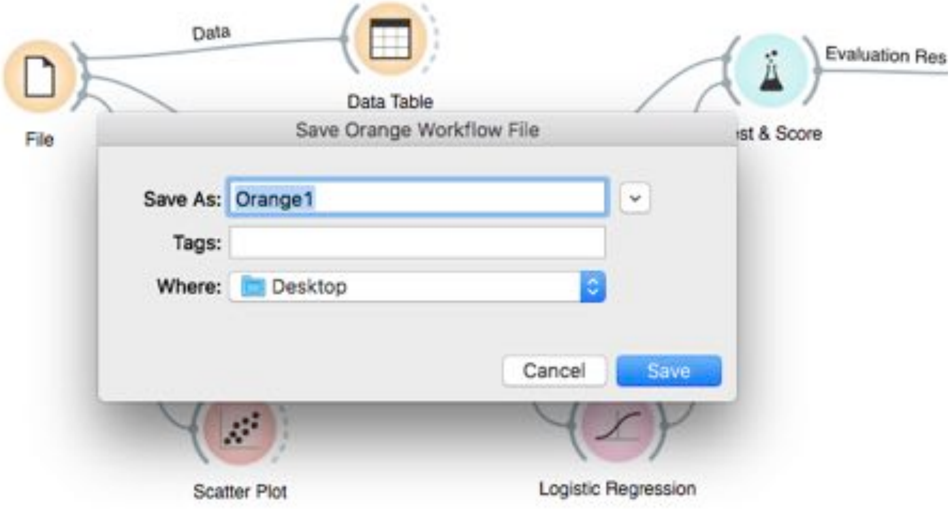
Sometimes the simpler methods are the better ones, isn't it?

This is how your final workflow would look after you are done with the complete process.



Display a confusion matrix constructed from the results of classifier evaluations.

For people who wish to work in groups, you can also export your workflows and send it to friends who can work alongside you!



Save As: Orange1

Tags:

Where: Desktop

The resulting file is of the (.ows) extension and can be opened in any other Orange setup.

End Notes

Orange is a platform that can be used for almost any kind of analysis but most importantly, for beautiful and easy visuals. In this article, we explored how to visualize a dataset. Predictive modeling was undertaken as well, using a logistic regression predictor, SVM, and a random forest predictor to find loan statuses for each person accordingly.

Hope this tutorial has helped you figure out aspects of the problem that you might not have understood or missed out on before. It is very important to understand the data science pipeline and the steps we take to train a model, and this should surely help you build better predictive models soon!

Learn, engage, compete, and get hired!