Case Study Report

For the Subject

Data Warehousing Data Mining
(Trimester VIII)

Submitted by

Shivansh Kushwaha

S1032180745

Souurabh Gavhane

S1032180909

Pratik Singh

S1032181204

Saish Malunde

S1032181324

Under the Guidance of

Prof. Sheetal Girase

School of Computer Engineering and Technology
MIT World Peace University, Kothrud,
Pune 411 038, Maharashtra - India
2020-2021

List of Figures

Contents

Chapter 1
Introduction of the Tool

Orange data mining software was originally developed by scientists at the University of Ljubljana (Slovenia) in 1997 using the Python, Cython, C++ and C programming languages. The software's graphical environment and interfaces have been developed using the Python and Qt3 libraries (Demsar et al.,2013). This software, with the latest version presents on March 6, 2017 with Orange 3.4.0 has a simple interface on which users create a data analysis workflow by placing graphical components(widgets).

Orange is an open-source data mining tool with very strong data visualization capabilities. It allows you to use a GUI (Orange Canvas) to drag and drop modules and connect them to evaluate and test various machine learning algorithms on your data. This hands-on tutorial will go through setting up Orange and getting familiar with its GUI components. We do this by exploring a sample data set with some visualization widgets included with Orange.

1.1 Features:

1.1.1 **Canvas**: Graphical front-end for data analysis.

1.1.2 **Widgets:**

- **Data**: Widgets for data input, data filtering, sampling, imputation, feature manipulation.
- **Visualize:** Widgets for common visualization and multivariate visualization.
- **Classify:** A set of supervised machines learning algorithms.
- **Regression:** A set of supervised machines learning algorithms.
- **Evaluate:** Cross-validation, sampling-based procedures, reliability estimation and scoring of prediction methods.
- **Unsupervised:** Unsupervised learning algorithms for clustering and data projection techniques.
- **Add-ons:**
    - **Associate:** Widgets for mining frequent item sets and association rule learning.

- **Bioinformatics:** Widgets for gene set analysis, enrichment and access to pathway libraries.
- **Data fusion:** Widgets for fusing different data sets, collective matrix factorization of talent factors.
- **Educational:** Widgets for teaching machine learning concepts.
- **Geo:** Widgets for working with geospatial data.
- **Image analytics:** Widgets for working with images and ImageNet embeddings.
- **Network:** Widgets for graph and network analysis.
- **Text mining:** Widgets for natural language processing (NLP) and text mining.
- **Spectroscopy:** Widgets for analyzing and visualization of spectral datasets.

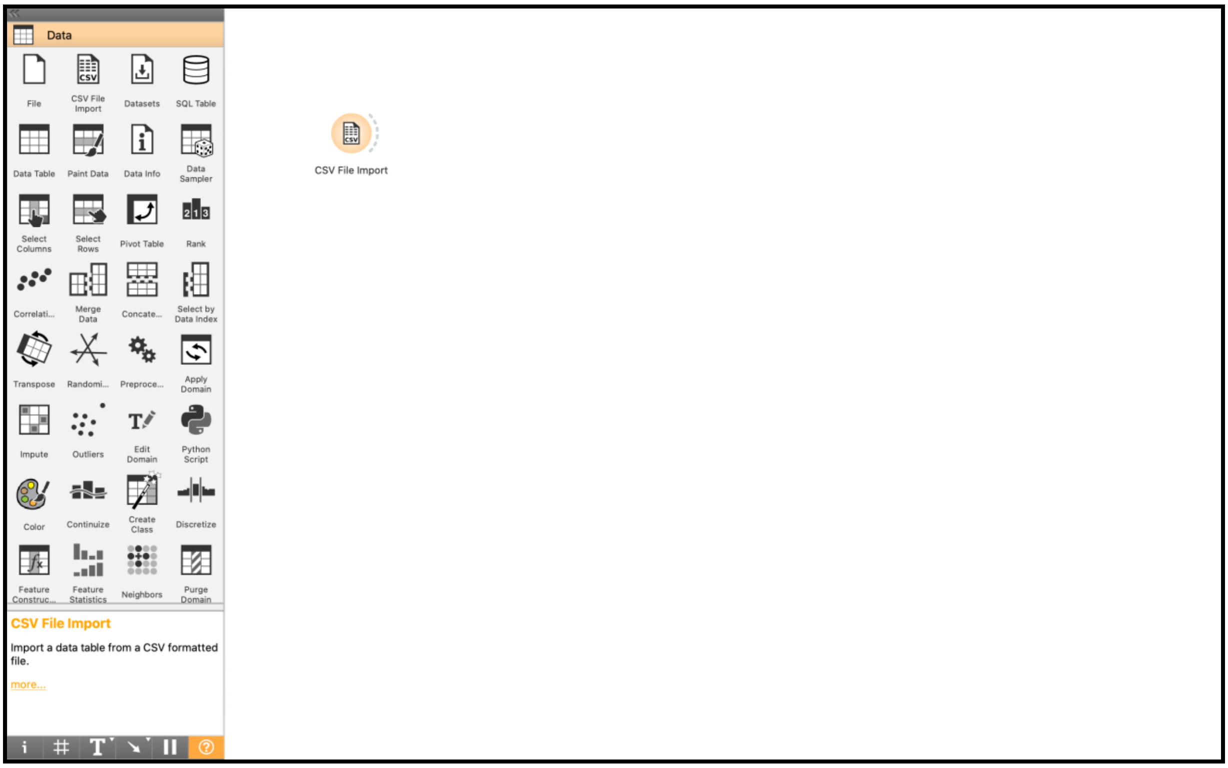


Figure 1.1: File widget dropped on the Canvas.

Chapter 2

How To perform Supervised/Unsupervised Learning

2.1 Supervised learning:

If you're learning a task under supervision, someone is present judging whether you're getting the right answer. Similarly, in supervised learning, that means having a full set of labeled data while training an algorithm.

Fully labeled means that each example in the training dataset is tagged with the answer the algorithm should come up with on its own. So, a labeled dataset of flower images would tell the model which photos were of roses, daisies and daffodils. When shown a new image, the model compares it to the training examples to predict the correct label.

There are two main areas where supervised learning is useful: classification problems and regression problems.

2.1.1Supervised learning in Orange:

- **Classification:** Much of Orange is devoted to machine learning methods for classification, or supervised data mining. These methods rely on data with class-labeled instances, like that of senate voting.
  Orange implements functions for construction of classification models, their evaluation and scoring.
  For supervised learning, Orange uses learners. These are objects that receive the data and return classifiers. Learners are passed to evaluation routines
  Classification Models:

    - Learners and Classifiers
    - Probabilistic Classification
    - Logistic Regression
    - Random Forrest
    - Cross-Validation
    - Handful of Classifiers
    - k-Nearest Neighbors
    - Naive Bayes
    - Support Vector Machines
    - Linear Support Vector Machines
    - Nu-Support Vector Machines
    - Classification Tree
    - Simple Tree

- Majority Classifier
- Neural Network
- CN2 Rule Induction
- Calibration and threshold optimization

- **Regression:** Regression in Orange is, from the interface, very similar to classification. These both require class-labeled data. Just like in classification, regression is implemented with learners and regression models (regressors). Regression learners are objects that accept data and return regressors. Regression models are given data items to predict the value of continuous class.

  Regression Models:

  - Linear Regression
  - Polynomial
  - Mean
  - Random Forest
  - Simple Random Forest
  - Regression Tree
  - Neural Network
  - Handful of Regressors
  - Cross Validation

2.2 Unsupervised learning:

Clean, perfectly labeled datasets aren't easy to come by. And sometimes, researchers are asking the algorithm questions they don't know the answer to. That's where unsupervised learning comes in.

In unsupervised learning, a deep learning model is handed a dataset without explicit instructions on what to do with it. The training dataset is a collection of examples without a specific desired outcome or correct answer. The neural network then attempts to automatically find structure in the data by extracting useful features and analyzing its structure.

2.2.1 Unsupervised learning in Orange:

- **Clustering:** Without being an expert ornithologist, it's possible to look at a collection of bird photos and separate them roughly by species, relying on cues like feather color, size or beak shape. That's how the most common application for unsupervised learning, clustering, works: the deep learning model looks for training data that are similar to each other and groups them together.

- **Association:** Fill an online shopping cart with diapers, applesauce and sippy cups and the site just may recommend that you add a bib and a baby monitor to your order. This is an example of association, where certain features of a data sample correlate with other features. By looking at a couple key attributes of a data point, an unsupervised learning model can predict the other attributes with which they're commonly associated.

- **Autoencoders:** Autoencoders take input data, compress it into a code, then try to recreate the input data from that summarized code. It's like starting with Moby Dick, creating a SparkNotes version and then trying to rewrite the original story using only SparkNotes for reference. While a neat deep learning trick, there are fewer real-world cases where a simple autocoder is useful. But add a layer of complexity and the possibilities multiply: by using both noisy and clean versions of an image during training, autoencoders can remove noise from visual data like images, video or medical scans to improve picture quality.

- **Anomaly Detection:** Banks detect fraudulent transactions by looking for unusual patterns in customer's purchasing behavior. For instance, if the same credit card is used in California and Denmark within the same day, that's cause for suspicion. Similarly, unsupervised learning can be used to flag outliers in a dataset.

2.3 Semi-Supervised learning:

Semi-supervised learning is, for the most part, just what it sounds like: a training dataset with both labeled and unlabeled data. This method is particularly useful when extracting relevant features from the data is difficult, and labeling examples is a time-intensive task for experts.

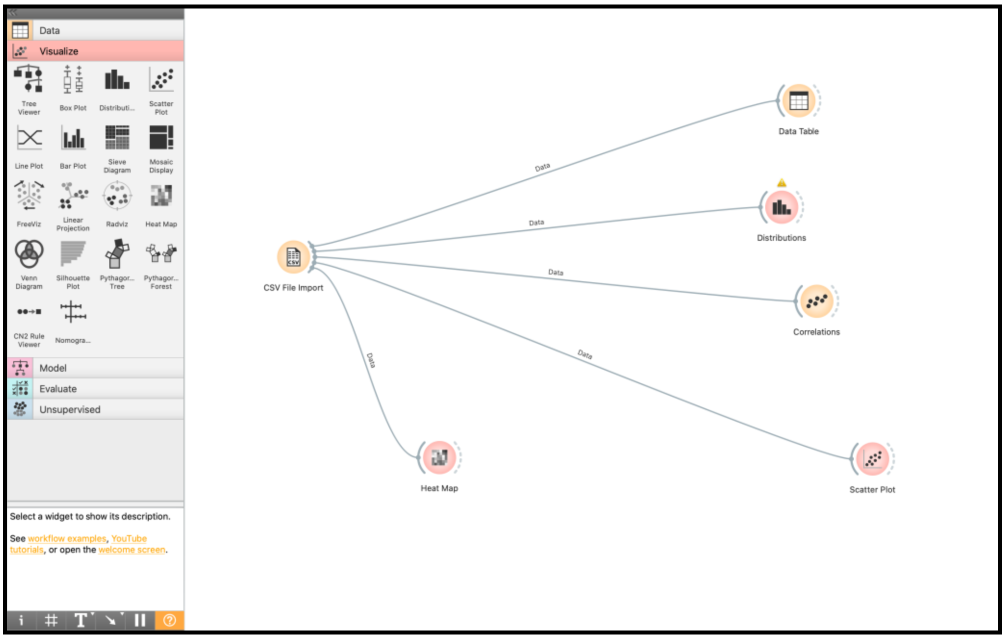


Figure 2.1: Workflow on canvas.

Chapter 3

How to perform Pre-processing/ Clustering.

3.1 Pre-processing of data-set:

- **Pre-processing using Python Script:**

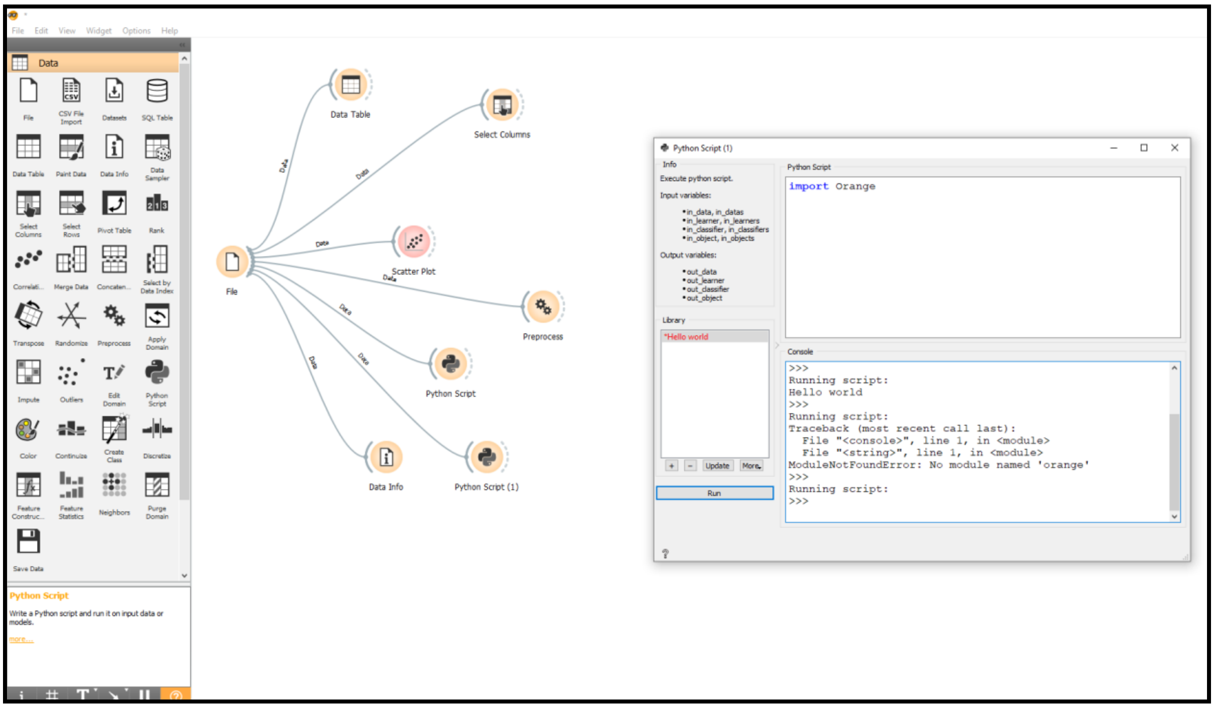


Figure 3.1: Python Script widget on canvas.

Python Script provided by the Orange console can be used to write all the functions or pre-processing such as eliminating null values from the dataset etc.

This can be achieved either by using the provided widget i.e., Python Script or on Anaconda by using "**import Orange**".

- **Pre-processing using Preprocess widget.**

Orange console also provides a widget called **Preprocess**. Using this widget, we can select the process we want to perform on our data-set such as:

  - Impute Missing Values
  - Normalize Features
  - Randomize

- Remove Sparse Features
- Select Relevant Features
- Select Random Features
- Continuize Discrete Variables

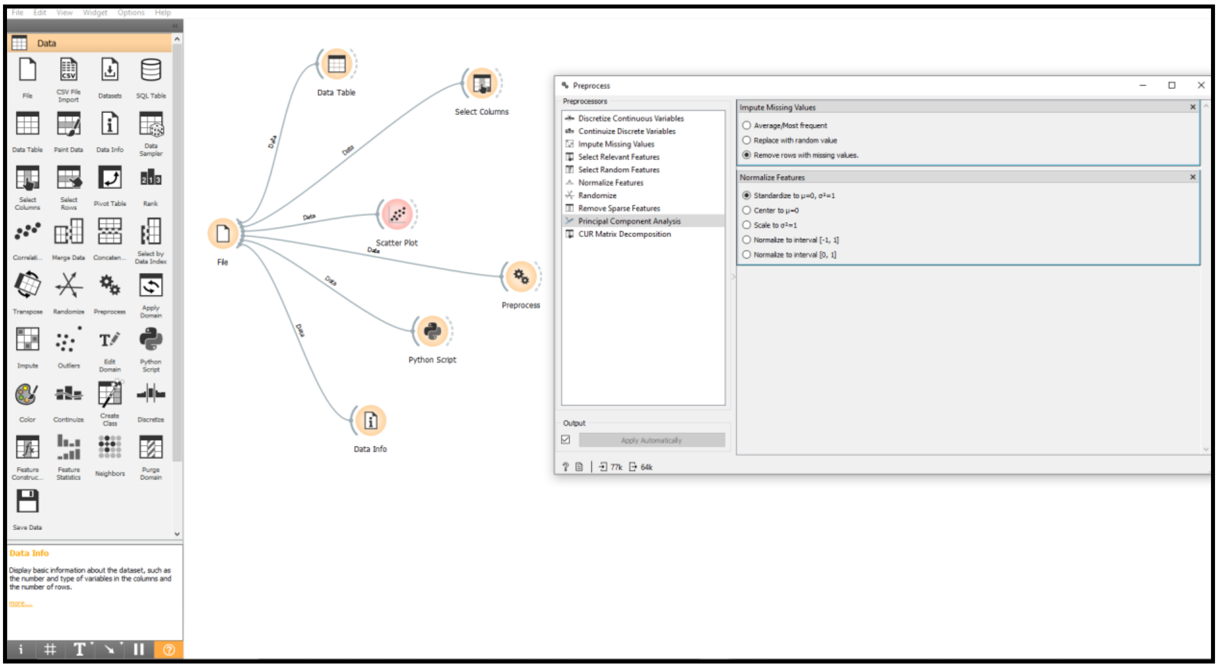


Figure 3.2: Preprocess widget on canvas.

3.2 Clustering:

Clustering is the process of making a group of abstract objects into classes of similar objects.

Points to Remember

- A cluster of data objects can be treated as one group.
- While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups.
- The main advantage of clustering over classification is that, it is adaptable to changes and helps single out useful features that distinguish different groups.

Clustering can be executed on console’s Python script using the following algorithm of hierarchical clustering.

Code to execute sample Iris data-set on console:

```
import Orange
from Orange import data, distance
from Orange.clustering import hierarchical
data = data. Table('iris')
dist_matrix = distance. Euclidean(data)
hierar = hierarchical. HierarchicalClustering(n_clusters=3)
hierar. linkage = hierarchical. AVERAGE
hierar.fit(dist_matrix)
hierar.labels
```

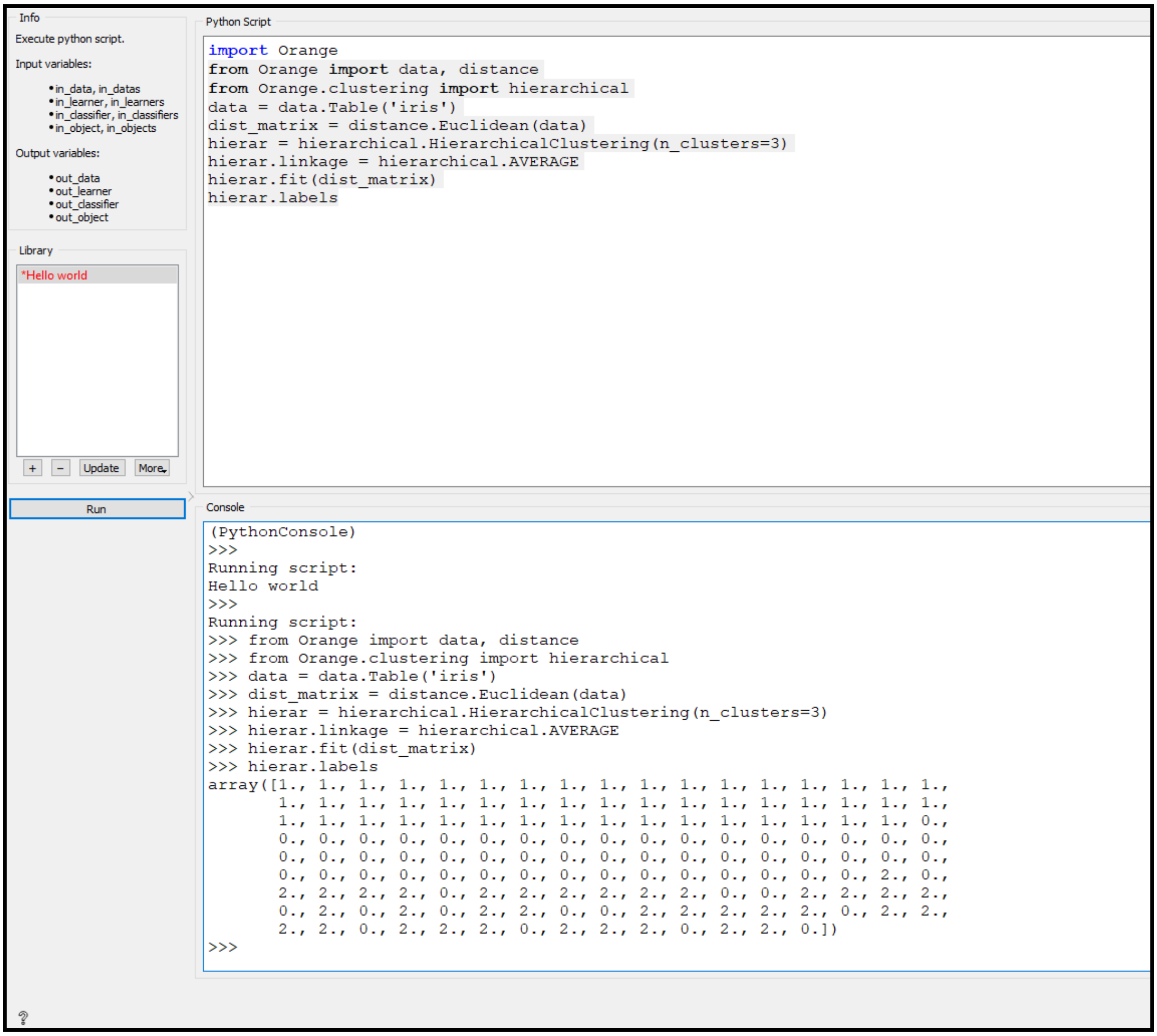


Figure 3.3: Python Script widget on canvas, running clustering algorithm.

Chapter 4
Description of the Dataset Used

There are two data sets used to discover and utilise various features of Orange Data Mining Tool.

- Animal rescue centre dataset.
  This dataset is recorded by Austin City Animal Shelter, Texas, USA. It features all the details of the rescued animal of all sorts and their outcome.
  It helped in keeping a record of stray animals and lost animals which later on provided the establishment of probability on adoption of animals according to their feature.

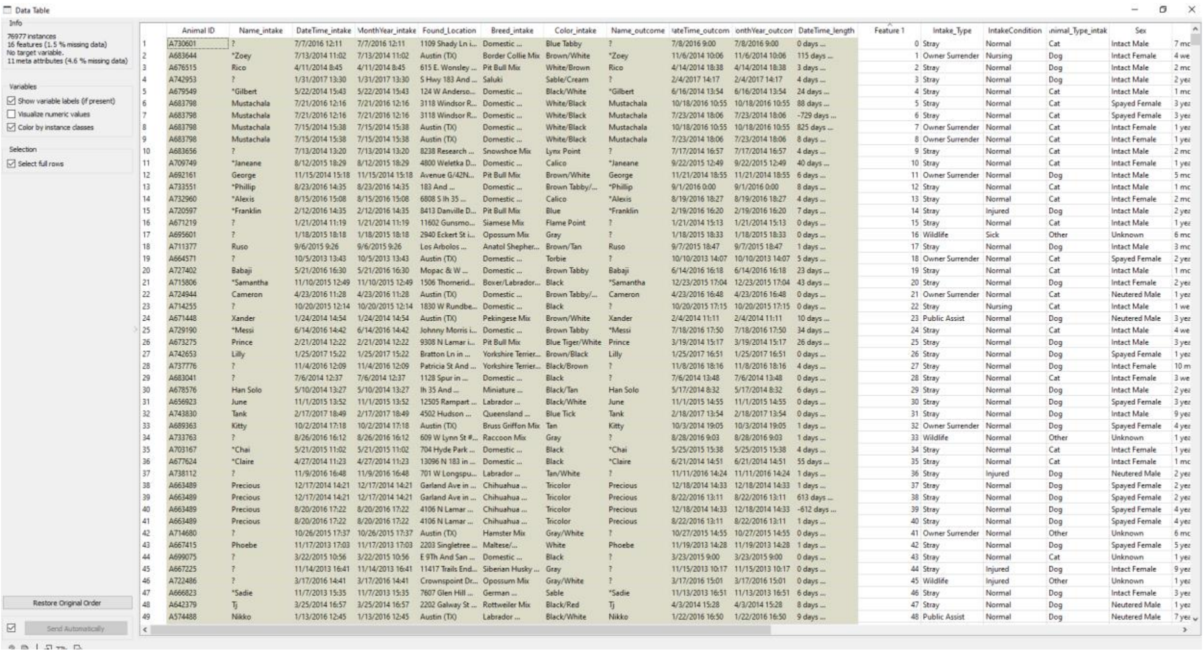
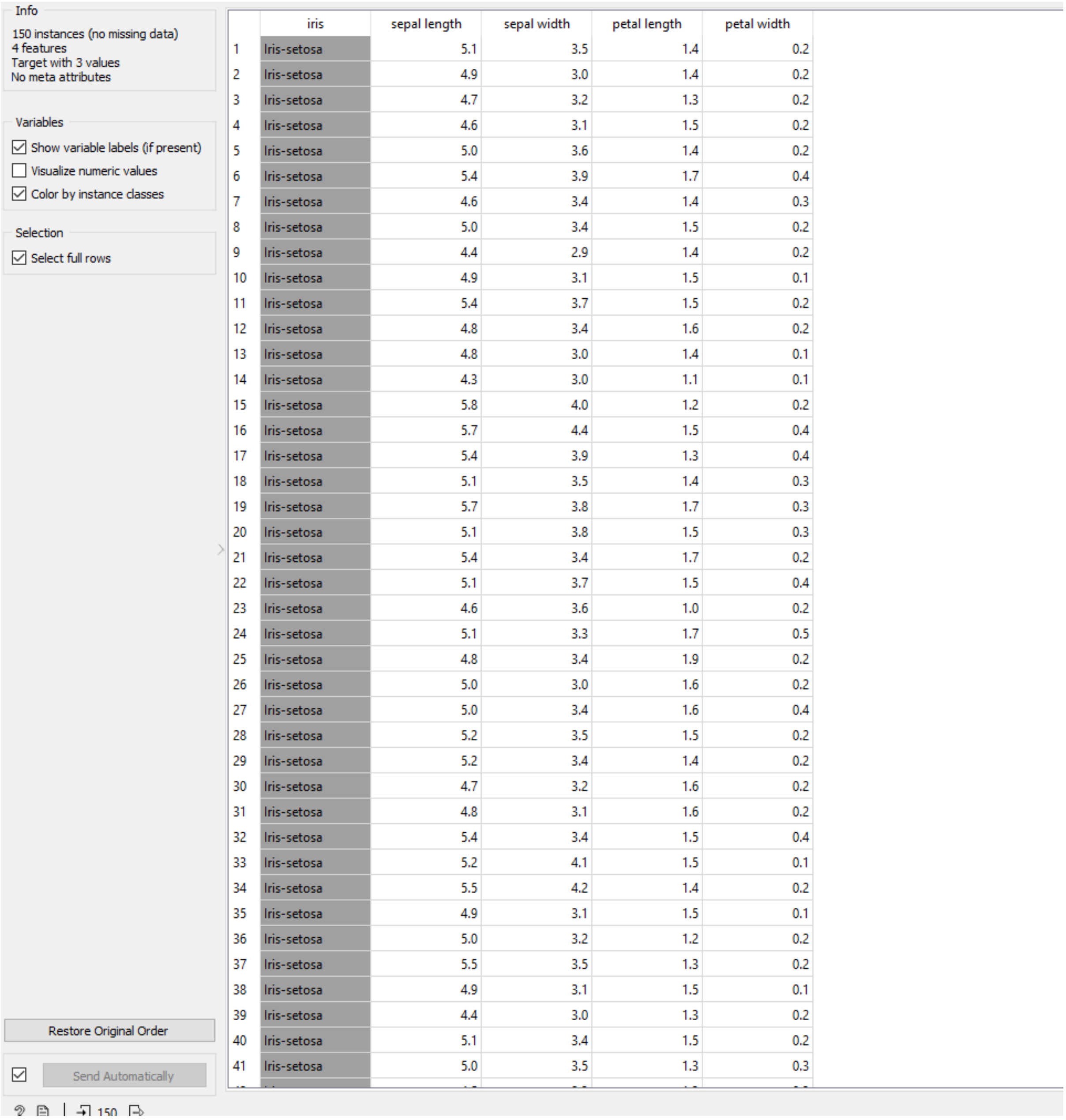


Figure 4.1: Animal Rescue Centre Dataset.

- Iris dataset (sample dataset of the console).
  This dataset is provided by the console as a sample to help in learning the variety of features on the software.



	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
18	Iris-setosa	5.1	3.5	1.4	0.3
19	Iris-setosa	5.7	3.8	1.7	0.3
20	Iris-setosa	5.1	3.8	1.5	0.3
21	Iris-setosa	5.4	3.4	1.7	0.2
22	Iris-setosa	5.1	3.7	1.5	0.4
23	Iris-setosa	4.6	3.6	1.0	0.2
24	Iris-setosa	5.1	3.3	1.7	0.5
25	Iris-setosa	4.8	3.4	1.9	0.2
26	Iris-setosa	5.0	3.0	1.6	0.2
27	Iris-setosa	5.0	3.4	1.6	0.4
28	Iris-setosa	5.2	3.5	1.5	0.2
29	Iris-setosa	5.2	3.4	1.4	0.2
30	Iris-setosa	4.7	3.2	1.6	0.2
31	Iris-setosa	4.8	3.1	1.6	0.2
32	Iris-setosa	5.4	3.4	1.5	0.4
33	Iris-setosa	5.2	4.1	1.5	0.1
34	Iris-setosa	5.5	4.2	1.4	0.2
35	Iris-setosa	4.9	3.1	1.5	0.1
36	Iris-setosa	5.0	3.2	1.2	0.2
37	Iris-setosa	5.5	3.5	1.3	0.2
38	Iris-setosa	4.9	3.1	1.5	0.1
39	Iris-setosa	4.4	3.0	1.3	0.2
40	Iris-setosa	5.1	3.4	1.5	0.2
41	Iris-setosa	5.0	3.5	1.3	0.3

Figure 4.2: Iris Dataset.

We used two datasets as the Animal rescue centre dataset is non-categorical and Isis being categorical, visualization was effectively executed.

Chapter 5
Implementation

- **Data Input:** Orange can read files in native tab-delimited format, or can load data from any of the major standard spreadsheet file types, like CSV and Excel. Native format starts with a header row with feature (column) names. The second header row gives the attribute type, which can be continuous, discrete, time, or string. The third header line contains meta information to identify dependent features (class), irrelevant features (ignore) or meta features (meta).

```
import Orange
data = Orange.data.Table("grp_data")
```

- **Saving Data:** Data objects can be saved to a file:

```
data.save("new_data.tab")
```

- **Exploring data:** Data table stores information on data instances as well as on data domain. Domain holds the names of attributes, optional classes, their types and, and if categorical, the value names.

```
import Orange

data = Orange.data. Table("grp_data")
n = len(data.domain.attributes)
n_cont = sum(1 for a in data.domain.attributes if a.is_continuous)
n_disc = sum(1 for a in data.domain.attributes if a.is_discrete)
print("%d attributes: %d continuous, %d discrete" % (n, n_cont, n_disc))

print(
    "First three attributes:",
    ", ".join(data.domain.attributes[i].name for i in range(3)),
)

print("Class:", data.domain.class_var.name)
```

- **Missing Values:** The particular data instance included missing data (represented with '?') for the first and the fourth attribute. In the original dataset file, the missing values are, by default, represented with a blank space.

```
>>> import numpy as np
>>> data = Orange.data.Table("iris")
>>> data[2]
```

```
[?, y, y, ?, y, ... | sepal]
>>> np.isnan(data[2][0])
True
>>> np.isnan(data[2][1])
False
```

- **Data Instances:** Data table stores data instances (or examples). These can be indexed or traversed as any Python list. Data instances can be considered as vectors, accessed through element index, or through feature name.

```
import Orange

data = Orange.data.Table("iris")
print("First three data instances:")
for d in data[:3]:
    print(d)

print("25-th data instance:")
print(data[24])

name = "sepal width"
print("Value of '%s' for the first instance:" % name, data[0][name])
print("The 3rd value of the 25th data instance:", data[24][2])
```

- **Mean:** The Iris dataset we have used above has four continuous attributes. Here's a script that computes their mean:

```
average = lambda x: sum(x) / len(x)

data = Orange.data.Table("iris")
print("%-15s %s" % ("Feature", "Mean"))
for x in data.domain.attributes:
    print("%-15s %.2f" % (x.name, average([d[x] for d in data])))
```

- **Data Selection and Sampling**: Besides the name of the data file, `Orange.data.Table` can accept the data domain and a list of data items and returns a new dataset. This is useful for any data subsetting:

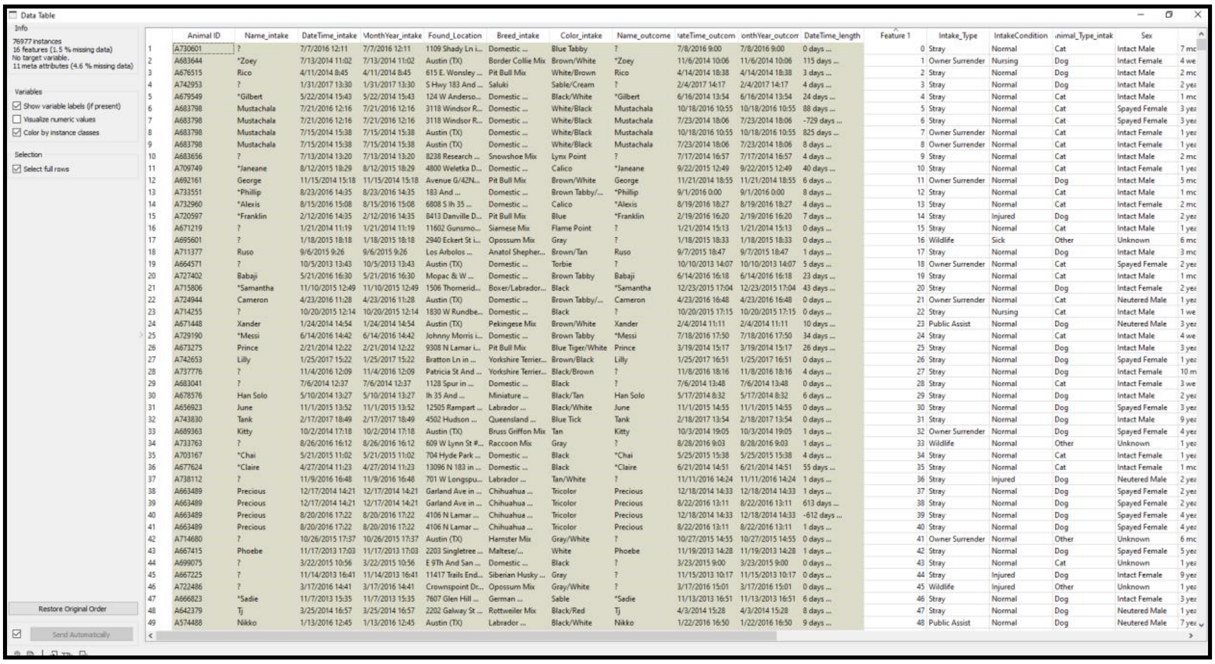
```
data = Orange.data.Table("iris.tab")
print("Dataset instances:", len(data))
subset = Orange.data.Table(data.domain, [d for d in data if d["petal
length"] > 3.0])
print("Subset size:", len(subset))
```

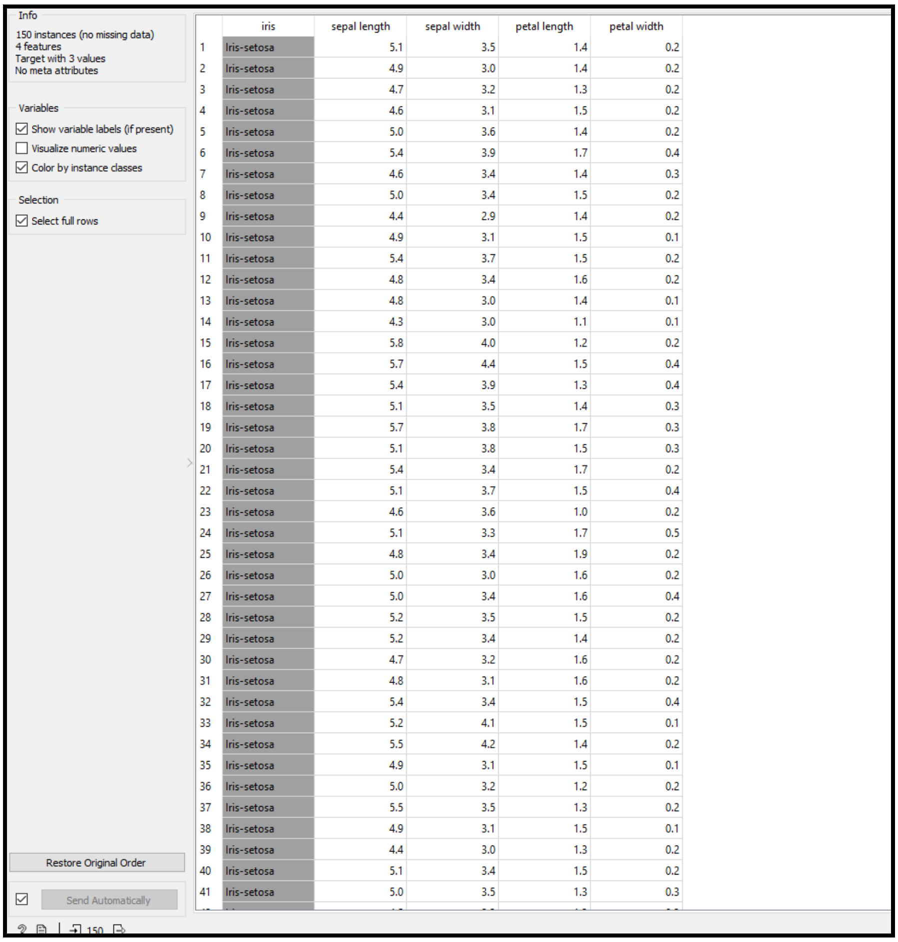
- The code outputs: (Data Selection and Sampling).

```
Dataset instances: 150
Subset size: 99
```

- **Classification Tree:** Orange includes three implemenations of classification trees. TreeLearner is home-grown and properly handles multinominal and missing values. The one from scikit-learn, SklTreeLearner, is faster. Another home-grown, SimpleTreeLearner, is simpler and still faster.

```
>>> import Orange
>>> iris = Orange.data.Table('iris')
>>> tr = Orange.classification.TreeLearner()
>>> classifier = tr(data)
>>> printed_tree = classifier.print_tree()
>>> for i in printed_tree.split('\n'):
>>>     print(i)
[50.  0.  0.] petal length ≤ 1.9
[ 0. 50. 50.] petal length > 1.9
[ 0. 49.  5.]     petal width ≤ 1.7
[ 0. 47.  1.]         petal length ≤ 4.9
   [0. 2. 4.]         petal length > 4.9
   [0. 0. 3.]             petal width ≤ 1.5
   [0. 2. 1.]             petal width > 1.5
   [0. 2. 0.]                 sepal length ≤ 6.7
   [0. 0. 1.]                 sepal length > 6.7
[ 0.  1. 45.]     petal width > 1.7
```





	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
18	Iris-setosa	5.1	3.5	1.4	0.3
19	Iris-setosa	5.7	3.8	1.7	0.3
20	Iris-setosa	5.1	3.8	1.5	0.3
21	Iris-setosa	5.4	3.4	1.7	0.2
22	Iris-setosa	5.1	3.7	1.5	0.4
23	Iris-setosa	4.6	3.6	1.0	0.2
24	Iris-setosa	5.1	3.3	1.7	0.5
25	Iris-setosa	4.8	3.4	1.9	0.2
26	Iris-setosa	5.0	3.0	1.6	0.2
27	Iris-setosa	5.0	3.4	1.6	0.4
28	Iris-setosa	5.2	3.5	1.5	0.2
29	Iris-setosa	5.2	3.4	1.4	0.2
30	Iris-setosa	4.7	3.2	1.6	0.2
31	Iris-setosa	4.8	3.1	1.6	0.2
32	Iris-setosa	5.4	3.4	1.5	0.4
33	Iris-setosa	5.2	4.1	1.5	0.1
34	Iris-setosa	5.5	4.2	1.4	0.2
35	Iris-setosa	4.9	3.1	1.5	0.1
36	Iris-setosa	5.0	3.2	1.2	0.2
37	Iris-setosa	5.5	3.5	1.3	0.2
38	Iris-setosa	4.9	3.1	1.5	0.1
39	Iris-setosa	4.4	3.0	1.3	0.2
40	Iris-setosa	5.1	3.4	1.5	0.2
41	Iris-setosa	5.0	3.5	1.3	0.3

Chapter 6
Visualisation

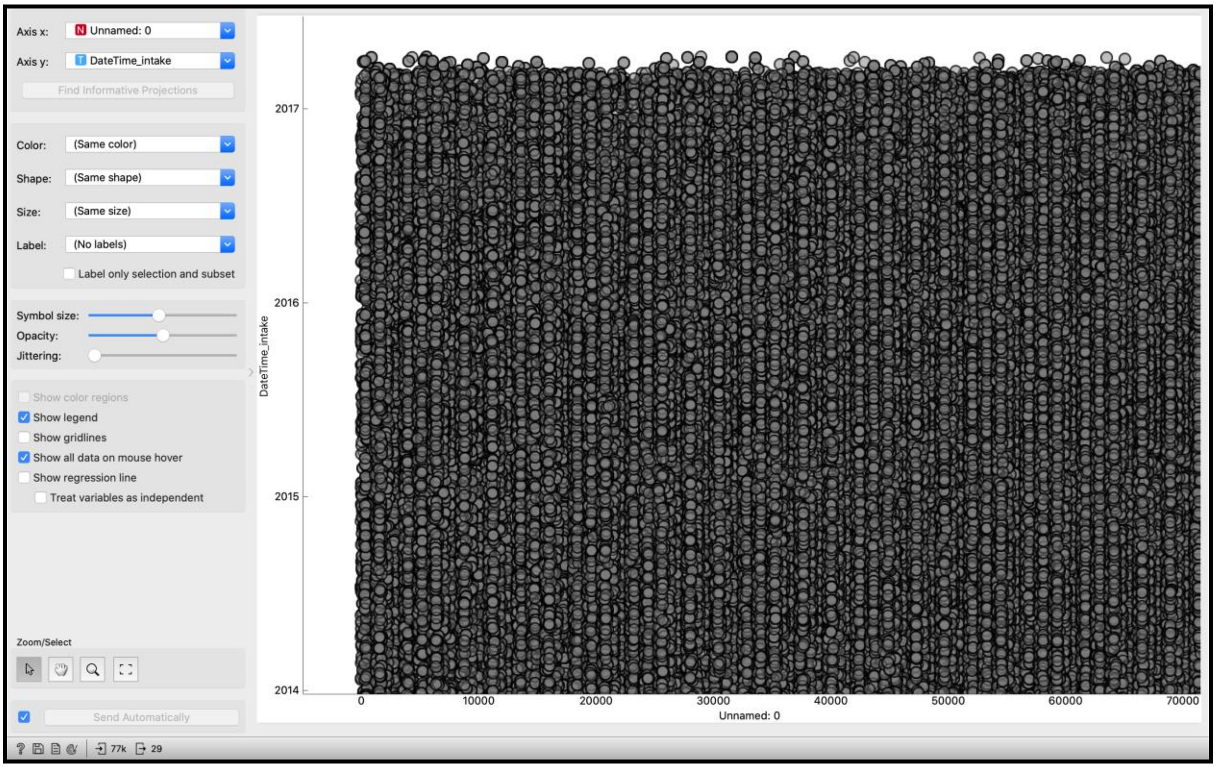


Figure 6.1: Scatter Plot.

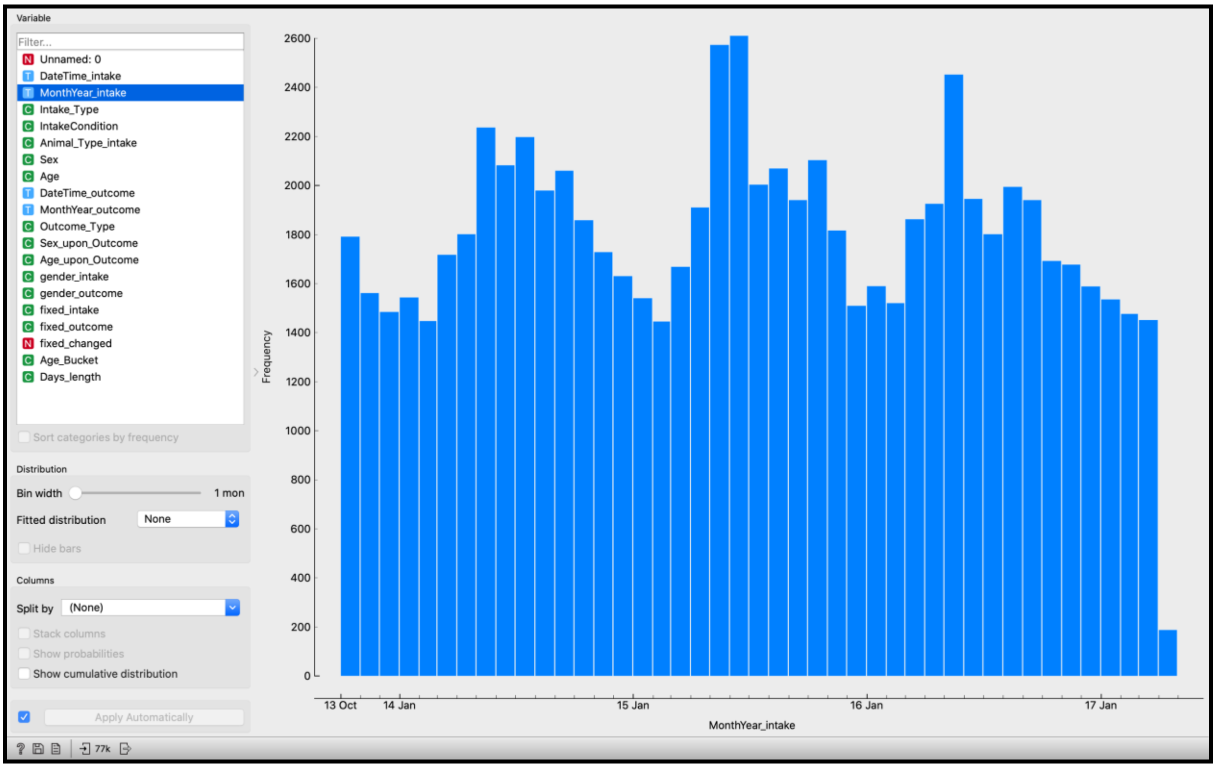


Figure 6.2: Bar Plot (Date Intake).

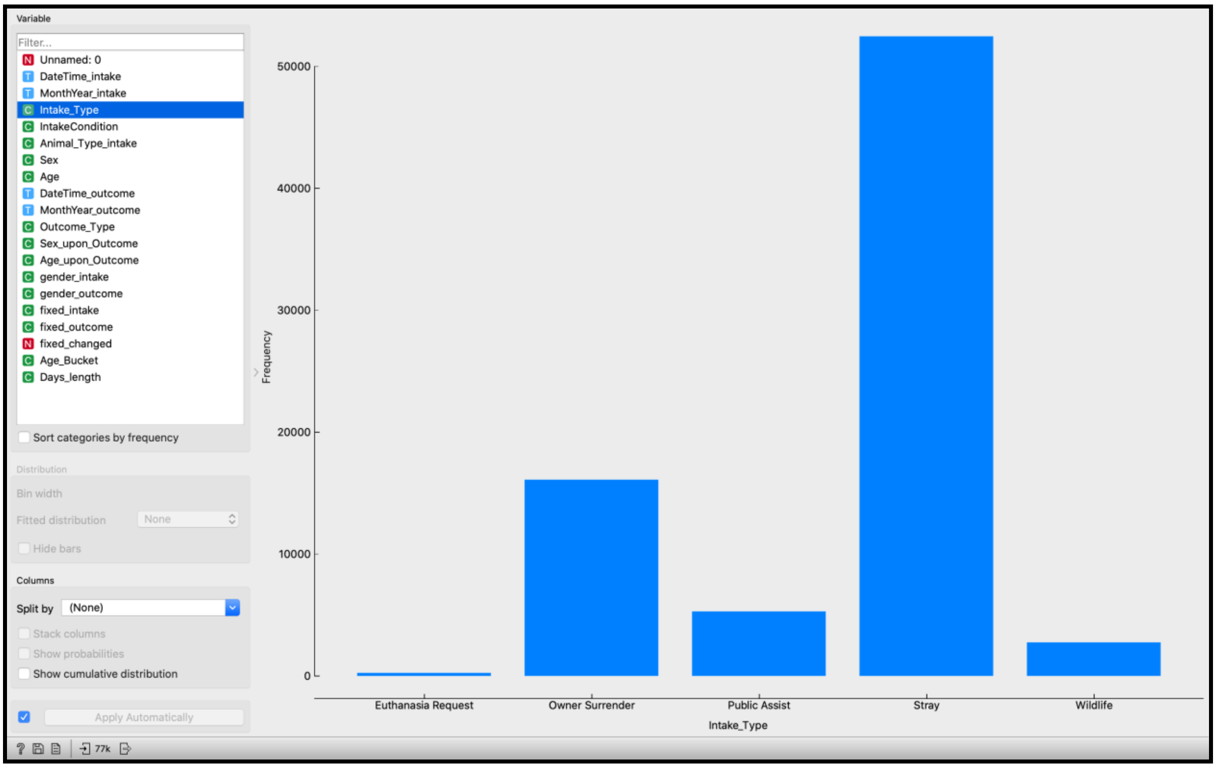


Figure 6.3: Bar Plot (Intake Type).

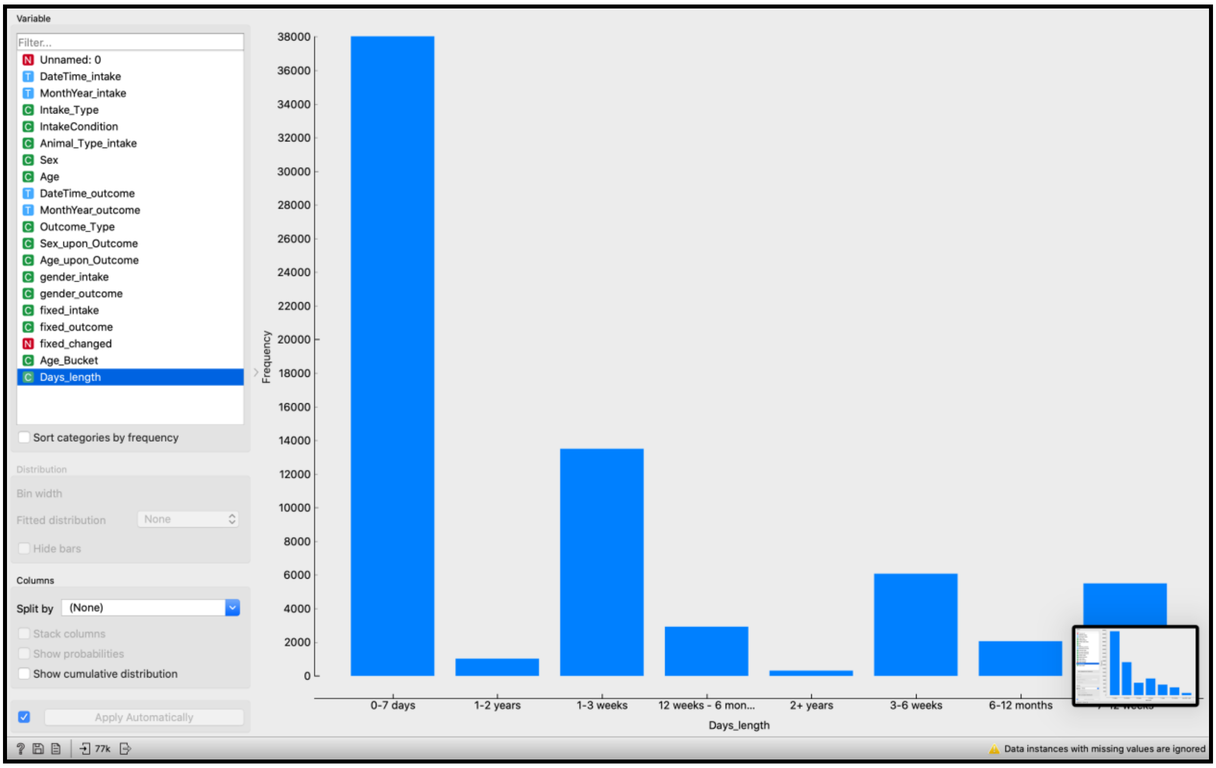


Figure 6.4: Bar Plot (Stay Length).

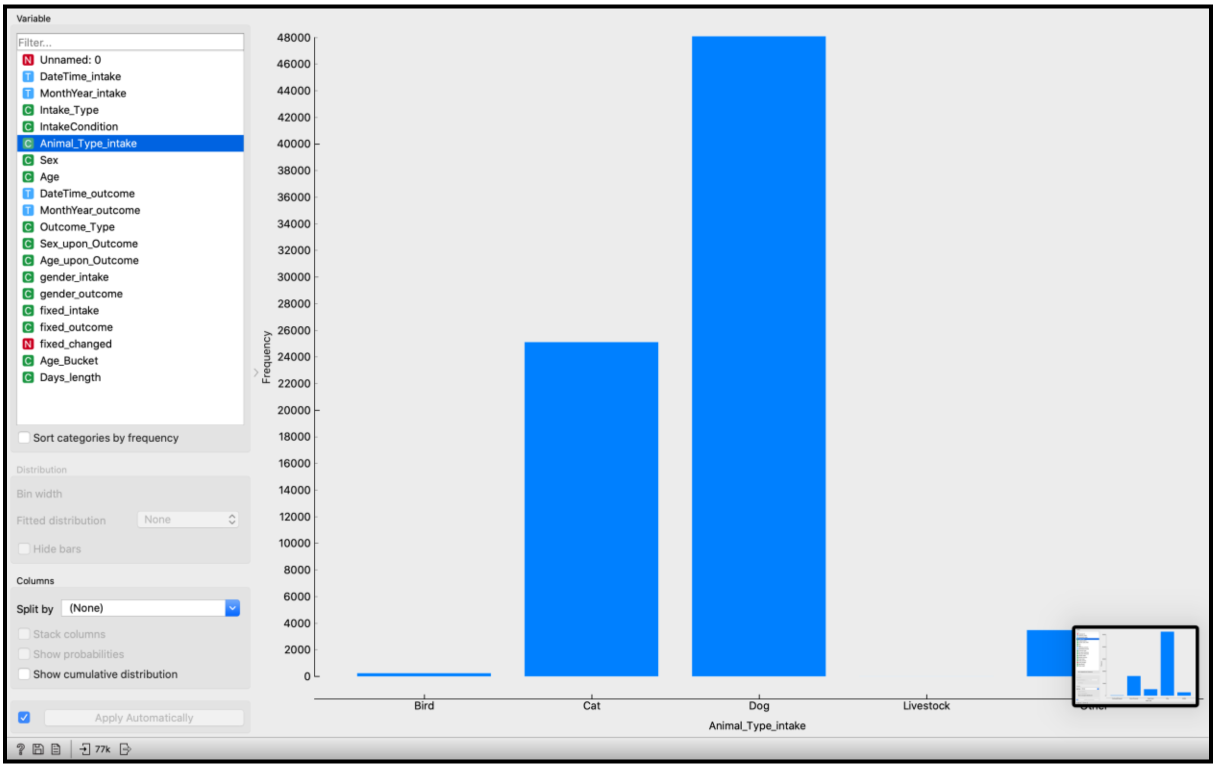


Figure 6.5: Bar Plot (Animal Type).

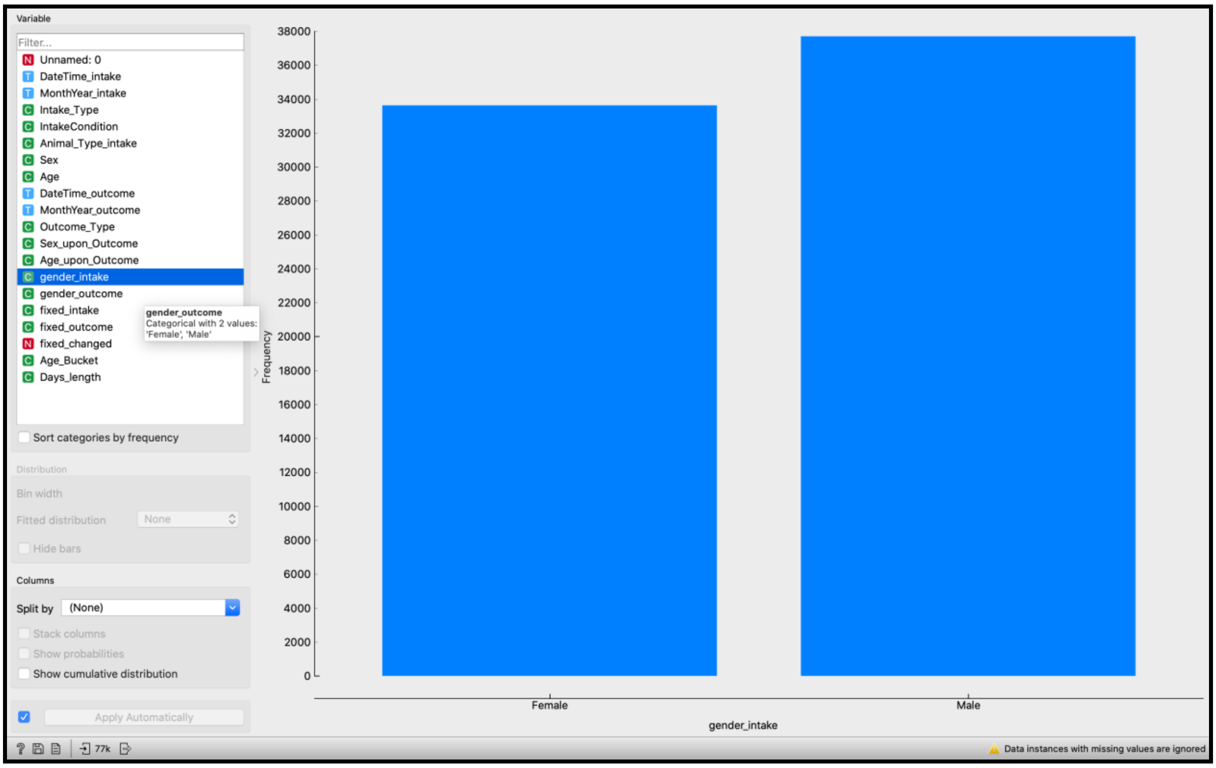


Figure 6.6: Bar Plot (Gender).

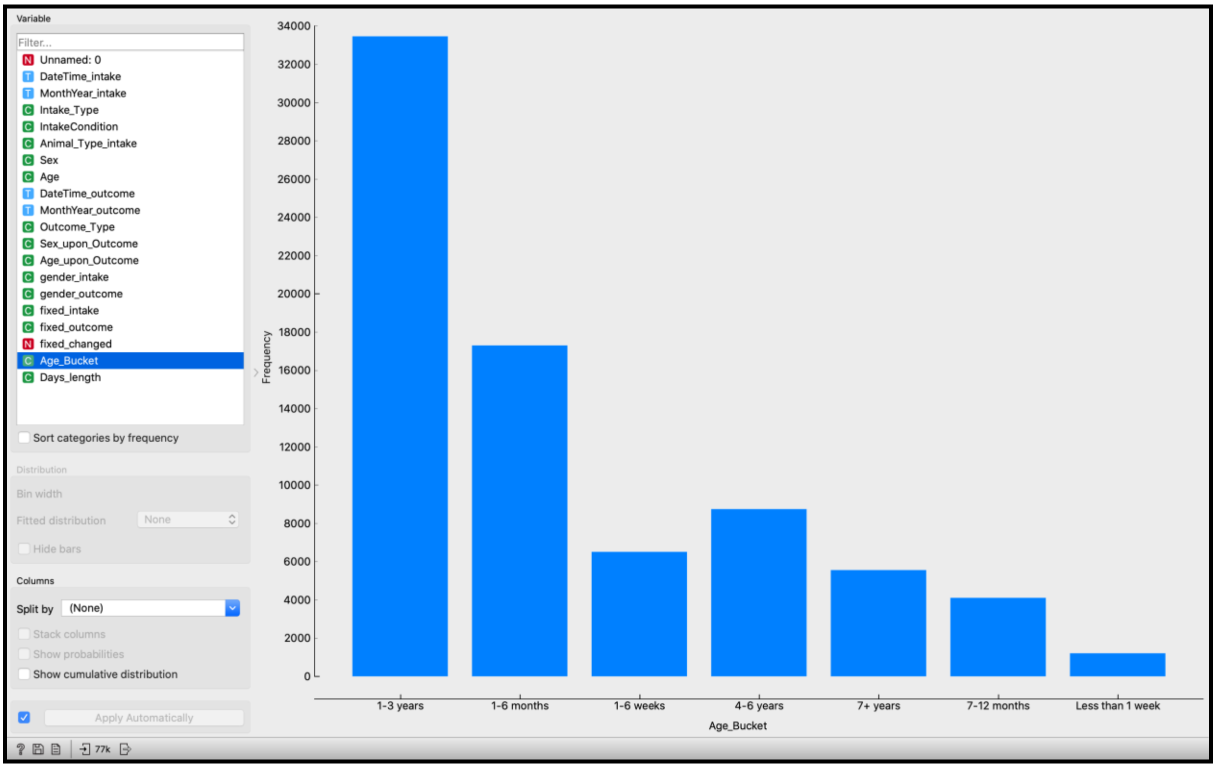


Figure 6.7: Bar Plot (Age).

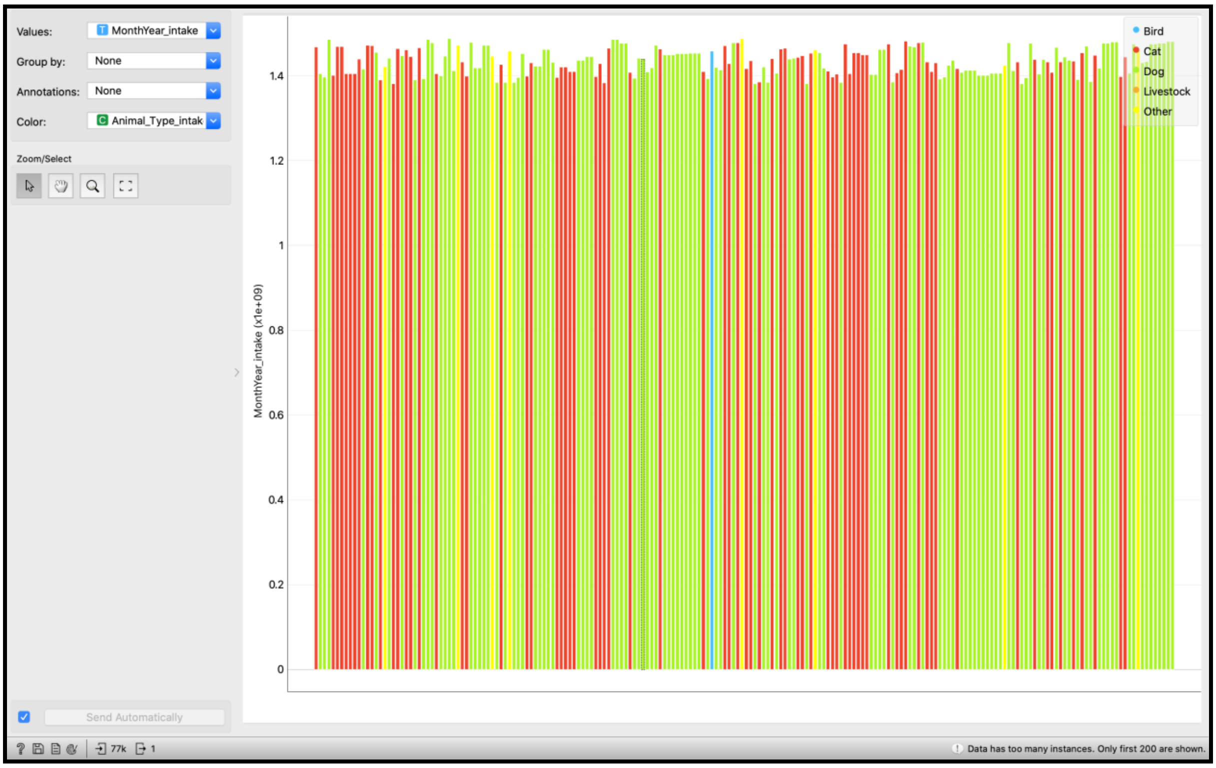


Figure 6.8: Histogram (Date Intake).

Chapter 6
Conclusion

Orange is a C++ core object and routines library that incorporates a huge variety of standard and non-standard machine learning and data mining algorithms. It is an open-source data visualization, data mining, and machine learning tool. Orange is a scriptable environment for quick prototyping of the latest algorithms and testing patterns. It is a group of python-based modules that exist in the core library. It implements some functionalities for which execution time is not essential, and that is done in Python.

It incorporates a variety of tasks such as pretty-print of decision trees, bagging and boosting, attribute subset, and many more. Orange is a set of graphical widgets that utilizes strategies from the core library and orange modules and gives a decent user interface. The widget supports digital-based communication and can be gathered together into an application by a visual programming tool called an orange canvas.

All these together make an orange an exclusive component-based algorithm for data mining and machine learning. Orange is proposed for both experienced users and analysts in data mining and machine learning who want to create and test their own algorithms while reusing as much of the code as possible, and for those simply entering the field who can either write short python contents for data analysis.

- **Strengths of using Orange as data mining tool:**
  - Open Source
  - Available on different prominent OS.
  - Python supported.
  - Python scripting inbuilt.
  - Appealing GUI.
  - Easy accessibility.
  - Various features of visualization and data processing available.
  - Support many Machine Learning Algorithms.

- **Weakness of using Orange data mining tool:**
  - No scope for developing ML programming skills. As usually we can connect objects on the canvas.
  - Not repeatable. You'll have to start again for different parameter intake.
  - Big Installation.
  - Limited Reporting capabilities.

References

[1] Manisha Chandak, Sheetal Girase, Introducing hybrid technique for optimization of book recommender system, International Conference on Advanced Computing Technologies and Applications (ICACTA-2015), Procedia Computer Science page nos- 23 – 31, ELSEVIER
[2]Semantic Scholar: https://www.semanticscholar.org/paper/Comparative-Study-of-Data-Mining-Tools-Rangra-Bansal/1597c7a98a9915dcde0f6e4aefda15cd032d6b13
[3]Clustering:https://www.tutorialspoint.com/data_mining/dm_cluster_analysis.htm#:~:text=Clustering%20is%20the%20process%20of,the%20labels%20to%20the%20groups.
[4]Orange Data Mining Library: https://orange3.readthedocs.io/projects/orange-data-mining-library/en/latest/tutorial/data.html
[5]Nvidia Blogs: https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/#:~:text=In%20a%20supervised%20learning%20model,and%20patterns%20on%20its%20own.