Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 26 No. 2 (2023)

A Comprehensive Analysis to Image Classification: Understanding Techniques and Explore Data Preprocessing a Non-linear Approach

Reena Thakur[1], Tanuksha Jambhulkar[2], Abhijeet Gadbail[3], Aditya Charpe[4], Akash Gomkale[5], Harshal Bhujade[6]

[1]Department of Computer Science and Engineering, Jhulelal Institute of Technology
Nagpur, India
en19cs601002@medicaps.ac.in

[2]Department of Computer Science and Engineering, Jhulelal Institute of Technology
Nagpur, India
tanukshajambhulkar07@gmail.com

[3]Department of Computer Science and Engineering, Jhulelal Institute of Technology
Nagpur, India
gadbailabhijeet12@gmail.com

[4]Department of Computer Science and Engineering, Jhulelal Institute of Technology
Nagpur, India
adityacharpe70@gmail.com

[5]Department of Computer Science and Engineering, Jhulelal Institute of Technology
Nagpur, India
akashgomkale722@gmail.com

[6]Department of Computer Science and Engineering, Jhulelal Institute of Technology
Nagpur, India
harshalbhujade899@gmail.com

Article History:

Received: *15-04-2023*

Revised: *20-06-2023*

Accepted: *28-06-2023*

Abstract:

Orange is a feature-rich open-source platform with a graphical user interface designed with data analysis and modeling in mind. Preprocessing, assessment, predictive modeling, and data visualization are just a few of the functions that this application offers. Its drag-and-drop interface allows users to create data analysis processes quickly and easily without requiring a lot of technical knowledge, making the process easy. Orange provides a wide range of machine learning techniques, from logistic regression and decision trees to support vector machines and neural networks. This wide range easily handles tasks including grouping, regression, and classification. Most notably, the tool promotes interaction with other libraries and tools, such Python and R, allowing users to pursue more complex modeling and data analysis projects. To put it briefly, Orange is a powerful tool that researchers, data scientists, and students may use to analyze and model data in an easy-to-use and effective manner. The goal of this investigation is to fully explore Orange's potential, which will make it a valuable tool in the fields of data science and machine learning.

Keywords: Orange, open-source, data mining, machine learning, graphical interface, data analysis, modeling, data visualization, preprocessing, predictive modeling, evaluation, Non-linear.

I. INTRODUCTION

A fundamental job in computer vision, image categorization has various applications, from autonomous cars to medical diagnosis. With the growing amount of visual data being generated by the digital world, effective and precise picture categorization methods are becoming more and more important [1]. The goal of this thorough examination is to dive into the nuances of picture classification, illuminating various methods and highlighting the critical role that data preparation plays in improving model performance. Assigning a name or category to a picture based on its content is the main goal of image classification. In order to generalise and correctly classify fresh, unseen data, this entails training a model to identify patterns and characteristics within pictures. Conventional machine learning techniques frequently employ feature extraction techniques like Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG). These traits operate as fundamental descriptors that encapsulate the distinct qualities of pictures, providing the groundwork for further categorization. Convolutional Neural Networks (CNNs), in particular, have been the dominating force in image categorization in recent years thanks to deep learning. When it comes to automatically learning hierarchical features, CNNs are excellent; human feature engineering is no longer necessary [2]. Combining transfer learning with pre-trained models, such ResNet or VGG16, has shown to be quite successful. This method optimises both time and computing resources by using the information that has been acquired from training on large datasets such as ImageNet and fine-tuning these models on particular picture classification tasks. Nevertheless, selecting the right algorithms is not the only factor in picture classification success. The way that data is preprocessed has a significant impact on how successful models are. Consistent model training is achieved by normalisation, which is the act of standardising pixel values to a similar scale (such as 0 to 1). By producing more training examples, augmentation techniques like rotation, flipping, and zooming lower the likelihood of overfitting and improve the model's capacity for generalisation.

A crucial step in data preparation is resizing photos to a standard size, which promotes model compatibility and homogeneity in input size. Using stratified sampling to handle unbalanced data guarantees that each class is represented proportionately in both training and validation sets, reducing bias and guaranteeing strong model performance across all categories [3]. Eliminating outliers and superfluous photos that might compromise the accuracy of the model is known as data cleaning. Z-score normalisation is one of the normalisation approaches that centres data around zero, which helps to speed up and stabilise training processes. One-hot encoding guarantees interoperability with several machine learning models by translating category labels into numerical notation. Combining predictions from many models—a method called ensemble learning—is another one that is used in picture categorization. Through the process of training many models and combining their predictions, overall accuracy and resilience are increased. In summary, the quality of data preparation is a critical factor in the dynamic and ever-evolving area of image classification, which influences both the choice of approaches and the efficacy of models. Accurate picture classification, whether achieved by conventional machine learning or cutting-edge deep learning techniques, depends on a comprehensive strategy that combines sophisticated algorithms and careful data preparation [4]. This thorough study acts as a roadmap for practitioners as we continue to explore the visual landscape of data, pointing them in the direction of efficient picture categorization techniques that make the most of contemporary technology.

II. LITERATURE REVIEW

The literature on Orange emphasises the tool's importance in data mining, machine learning, and data analysis. It starts with its introduction in Demšar et al.'s groundbreaking work in 2004 and continues with more current surveys and reviews. These studies highlight how Orange's scriptable environment, drag-and-drop interface, and adaptable widget set make it easy for researchers, data scientists, and students to utilise. Orange's flexibility to specialised fields is shown by its application in bioinformatics, genetic research, and healthcare, as demonstrated by Demšar et al. in 2013. The literature demonstrates that Orange is a useful tool for a variety of applications because to its flexibility, extensibility, and ease of prototyping, despite challenges like the learning curve for new users and the possible complexity of bioinformatics tasks.

Orange's merits and shortcomings are elucidated by comparison studies conducted by Dey et al. (2016) and surveys such as Suto et al.'s 2020 investigation of visual programming languages, which situate Orange within the larger context of data mining tools. With research like Žnidaršič et al.'s examination into the efficiency of visual programming for creating predictive models, the literature also acknowledges the tool's function in predictive modelling. The convenience of generating data analysis workflows and the benefits of visual aids such as workflow complexity are acknowledged, but the ease of developing data analysis workflows also stands out.

Considering these aspects, the literature presents Orange as a flexible and easy-to-use tool that successfully tackles a variety of data analysis and machine learning problems. For both academics and industry professionals, the platform offers an extensive and intuitive user interface that supports a wide range of applications, from general data mining to specialised domains like computer vision and biology.

Author & Year	Area	Methodology	Key Findings	Challenges	Pros	Cons	Application
Demšar et al. (2004)	Data Mining	Scriptable environment, visual programming	Introduction to Orange, user-friendly interface	Limited execution time for certain functionalities	Easy prototyping, versatile tools	Limited execution speed	Data analysis and modeling
Demšar et al. (2013)	Bioinformatics	Python-based modules, widgets, visual programming	Application in bioinformatics, genomic research	Potential complexity in bioinformatics tasks	Simplifies complex data tasks, diverse applications	May require expertise for advanced tasks	Genomic research, biomedicine
Žnidaršič et al. (2009)	Predictive Modeling	Visual programming, drag-and-drop	Effectiveness of visual programming in	Potential learning curve for new users	User-friendly interface, visual	Initial learning curve	Predictive modeling, data analysis

		interface	predictive modeling		representation		
Demšar (2013)	Data Mining	Visual programming, Orange Canvas	In-depth exploration of Orange Canvas and widgets	Workflow complexity with extensive use of widgets	Modularity, extensibility	Complexity in workflows	Data analysis, machine learning
Dey et al. (2016)	Data Mining Tools	Comparative analysis	Evaluation of data mining tools, including Orange	Varied learning curves for different tools	Offers diverse tools, strengths, and weaknesses	Tool-dependent challenges	General data mining applications
Suto et al. (2020)	Visual Programming Languages	Survey	Evolution of visual programming languages in data science	Potential for limited integration with traditional coding	Provides insights into visual programming landscape	Integration challenges	Data science workflows
Jordan and Mitchell (2015)	Machine Learning	Review	Foundational insights into machine learning algorithms	Challenges in hyperparameter tuning	Theoretical foundations, algorithm insights	Hyperparameter tuning complexity	General machine learning applications
Geron (2019)	Computer Vision	Practical insights	Practical considerations for machine learning in computer vision	Computational resource demands	Practical guidelines for computer vision	Computational demands	Computer vision tasks, image classification

Table 1. Related work

III. ARCHITECTURE

User Interface: The Orange architecture's uppermost layer is the user interface. With drag-and-drop capability, users can simply develop and change data analysis processes thanks to its straightforward and user-friendly design.

Orange's primary components are its machine learning algorithms, data visualisation tools, and data pretreatment tools. These elements serve as the foundation for creating processes for data analysis.

Data Handling: CSV files, SQL databases, and other data formats are just a few of the sources of data that Orange's data handling layer is in charge of importing and exporting. It also takes care of normalisation, transformation, and cleansing of data.

Three primary components comprise Orange Tool's architecture:

Backend: The Orange tool's backend handles data manipulation, analysis, and machine learning, among other essential software functions. NumPy, SciPy, Matplotlib, and scikit-learn are among of the libraries for data processing, visualisation, and machine learning that are included with the Python backend.Study

Frontend: Orange Tool's frontend is in charge of the user interface and user communication. The frontend features a graphical user interface that enables users to drag and drop widgets to design data analysis processes. It is constructed using the Qt cross-platform application framework. A number of visualisations for data analysis and exploration are also included in the frontend.

Add-ons: Extra modules that may be added to an Orange tool to increase its capability. Additional data sources, visualisations, and machine learning techniques are available as installable add-ons. Either the Orange team or other developers may create the add-ons.

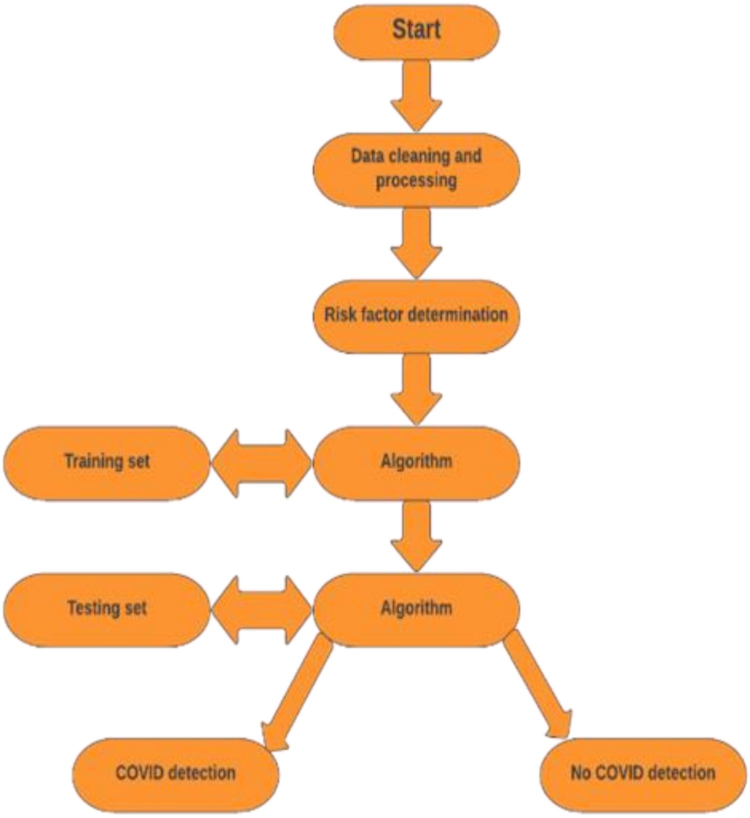


Figure 1. Architecture of Orange Tool on Covid Detection

As shown in the above figure it provides the idea for how the orange tool actually works starting from taking data cleaning , processing and afterwards applying algorithms on it.

Orange's machine learning algorithms are in charge of creating predictive models from the data they receive. Among these algorithms are neural networks, support vector machines, logistic regression, decision trees, and more.

Python Library: Because Orange is based on the Python computer language, it can access the extensive Python library of machine learning and data science programmes. This enables users to include other features into their processes for data analysis.

Platform Independence: Orange is intended to function on any operating system that supports Python, regardless of the platform. This facilitates its accessibility for users on many platforms and eases its deployment in various contexts.All things considered, Orange Tool's architecture is made to be flexible, expandable, and modular. This makes it easy for users to add new functionality and modify the programme to suit their needs. Orange tool is a flexible tool for data analysis and machine learning thanks to its strong backend, intuitive interface, and configurable add-ons.

IV. DATA PREPROCESSING

In the Orange tool, "data processing" refers to the numerous methods and instruments that are accessible within the programme to manipulate and modify data. Orange is a visual programming tool for data mining and analysis that enables users to create machine learning models, interactively explore and preprocess data, and visualise the outcomes.

Orange may process data using a number of methods, including feature selection, data cleansing, normalisation, transformation, and dimensionality reduction. By eliminating noise, outliers, and missing values from the data and lowering its dimensionality to make it easier to handle for analysis, these strategies aid in getting the data ready for analysis.

Orange tool processes data using a number of algorithms according on the particular job or method being used. Orange uses the following algorithms as examples for various data processing tasks:

- Data Cleaning: The Orange Tool offers a number of data cleaning techniques, including ones for managing outliers, missing values, and duplicate record removal. Among the algorithms are a few that:
  1.Remove Duplicates
  2.Replace Missing Values
  3.Remove Outliers
- Feature Selection: Feature selection is an important step in data processing that involves selecting the most relevant features for the analysis.
  Orange tool provides several algorithms for feature selection such as:
  1.Information Gain
  2.Chi-Squared
  3.Recursive Feature Elimination

- Dimensionality Reduction: Dimensionality reduction is the process of reducing the number of variables in a dataset while retaining the most important information. Orange tool provides several algorithms for dimensionality reduction such as:
  1.t-SNE
  2.Non-negative Matrix Factorization (NMF)

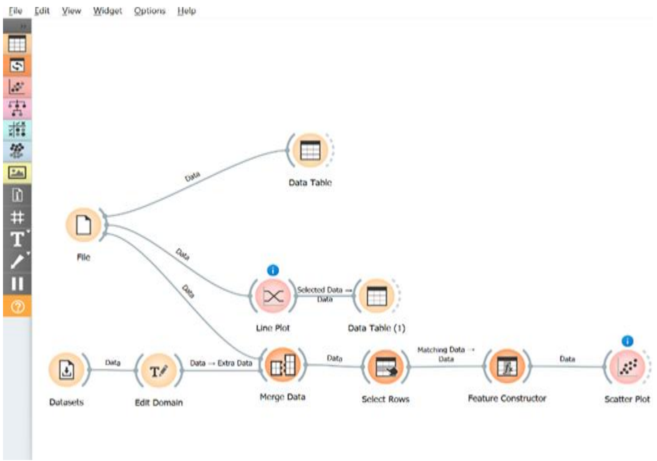
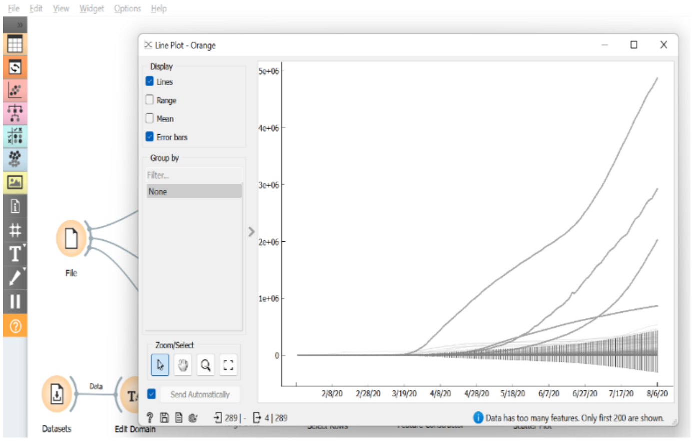


Figure 2. Data Processing

Figure 7 shows the data processing of the covid-19 dataset. In this we get the data table of the data set, domain of the data, etc, It will also merge the given data into one.



Display
☑ Lines
□ Range
□ Mean
☑ Error bars

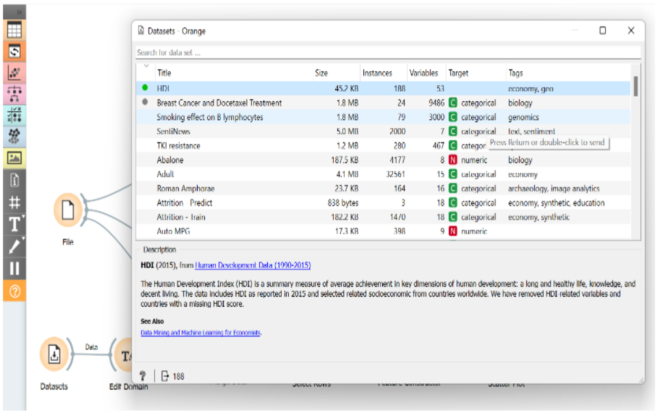
Group by
Filter...
None

☑ Send Automatically

Data has too many features. Only first 200 are shown.

Figure 3. Visualization for sequential data

Figure 8 shows the data line plot which is the perfect visualization for sequential data such as ours data. The curves represent the countries and the x axis in our plot is the data. From this plot we cannot see anything really since we have so many countries and the lines are thin. We can select a few lines by dragging across them and inspect them in the data table.



Datasets - Orange

Search for data set ...

	Title	Size	Instances	Variables	Target	Tags
●	HDI	45.2 KB	188	53		economy, geo
●	Breast Cancer and Docetaxel Treatment	1.8 MB	24	9486	C categorical	biology
	Smoking effect on B lymphocytes	1.8 MB	79	3000	C categorical	genomics
	SentiNews	5.0 MB	2000	7	C categorical	text, sentiment
	TKI resistance	1.2 MB	280	467	C categor	Press Return or double-click to send
	Abalone	187.5 KB	4177	8	N numeric	biology
	Adult	4.1 MB	32561	15	C categorical	economy
	Roman Amphorae	23.7 KB	164	16	C categorical	archaeology, image analytics
	Attrition - Predict	838 bytes	3	18	C categorical	economy, synthetic, education
	Attrition - Train	182.2 KB	1470	18	C categorical	economy, synthetic
	Auto MPG	17.3 KB	398	9	N numeric	

Description

HDI (2015), from Human Development Data (1990-2015)

The Human Development Index (HDI) is a summary measure of average achievement in key dimensions of human development: a long and healthy life, knowledge, and decent living. The data includes HDI as reported in 2015 and selected related socioeconomic from countries worldwide. We have removed HDI-related variables and countries with a missing HDI score.

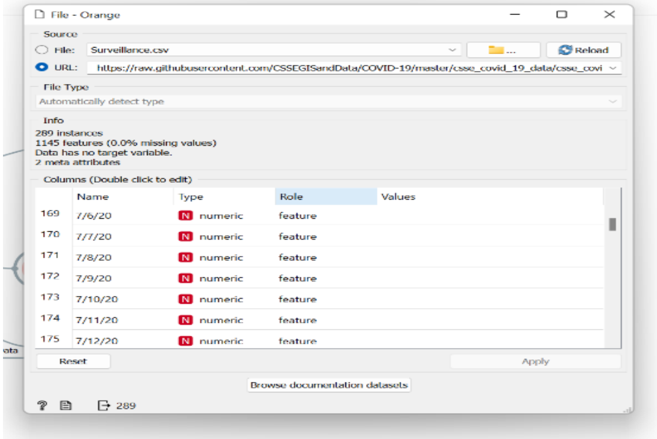
See Also

Data Mining and Machine Learning for Economists.

188

Figure 4. Human Development Index (HDI)

Figure 9 shows the handy data sets widget which by chance happens to have an HDI data set available. HDI or Human Development Index is a data set with many country statistics. Then merge the two datasets with the help of HDI.


File - Orange

Source

○ File: Surveillance.csv

● URL: https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi

File Type

Automatically detect type

Info

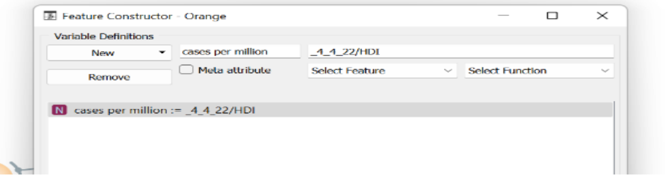
289 instances
1145 features (0.0% missing values)
Data has no target variable.
2 meta attributes

Columns (Double click to edit)

	Name	Type	Role	Values
169	7/6/20	N numeric	feature	
170	7/7/20	N numeric	feature	
171	7/8/20	N numeric	feature	
172	7/9/20	N numeric	feature	
173	7/10/20	N numeric	feature	
174	7/11/20	N numeric	feature	
175	7/12/20	N numeric	feature	

Figure 5. Dataset for Covid-19

In figure 10 we compute the cases to population ratio by selecting the final data from the covid dataset. Then divide the number by total population. Then remove the zero population first to avoid dividing by zero and tearing a hole in the fabric of the dataset.


Feature Constructor - Orange

Variable Definitions

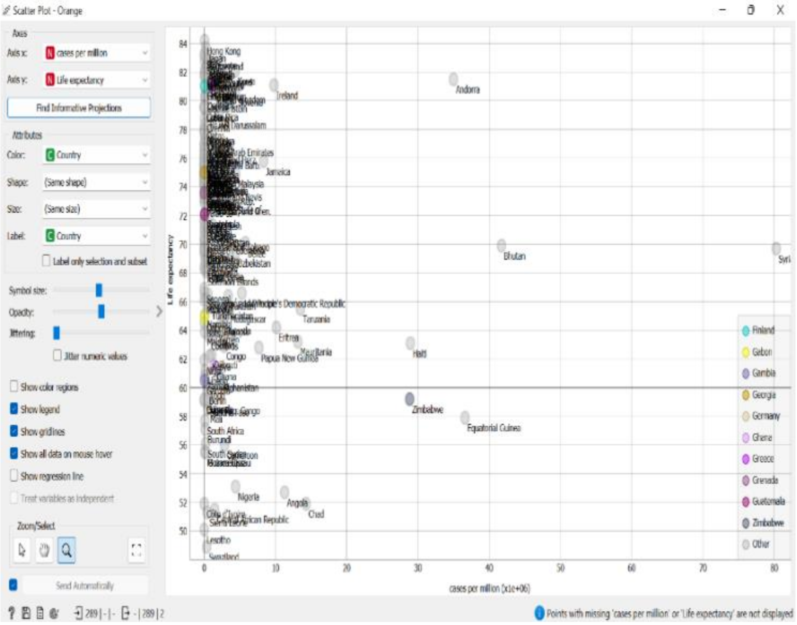
cases per million: _4_4_22/HDI

□ Meta attribute

N cases per million := _4_4_22/HDI

Figure 6. Picture Constructor to compute the cases to population ratio

In figure 11 feature constructor is created in for that we will select rows to merge data and set total population to be higher than zero this will come in handy in the next step.



Scatter Plot - Orange

Axes

Axis x: N cases per million

Axis y: N Life expectancy

Find Informative Projections

Attributes

Color: C Country

Shape: (Same shape)

Size: (Same size)

Label: C Country

□ Label only selection and subset

Symbol size:

Opacity:

Jittering:

□ Jitter numeric values

□ Show color regions

☑ Show legend

☑ Show gridlines

☑ Show all data on mouse hover

□ Show regression line

□ Treat variables as independent

Zoom/Select

☑ Send Automatically

289 | - | - 289 | 2

Points with missing 'cases per million' or 'Life expectancy' are not displayed

Figure 7. An approximation for health system capability

In figure 12 after computing the cases connect scatterplot to feature constructor and by selecting cases for the x axis and physicians for the y axis. This will show us how covid spread is related to the number of doctor in the country and label the points with country names to make sense of the plot.

V. PROPOSED CLASSIFIER

We have two data sets one is for image classification and the other is for data processing .The dataset we have taken for image classification is rice leaf disease from UCI Machine Learning Repository and dataset for Data Preprocessing is Covid 19 dataset provided by John Hopkins University's COVID-19 Dashboard.

Image Classification:

Image classification is a subfield of computer vision that involves the automatic classification of images based on their visual content. In other words, image classification is the process of assigning a label or a category to an image based on its content.

Image classification algorithms typically use machine learning techniques to learn patterns in the visual content of images. These algorithms are trained on a large dataset of labeled images, where each image is assigned a category or label. During the training process, the algorithm learns to recognize the features or patterns that are characteristic of each category**.**

We can implement image classification using various methods :-

Random Forest : Random Forest is a popular machine learning method used in data mining, and it is also available in the Orange data mining tool. Random Forest is an ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Logistic Regression: Logistic Regression is another popular machine learning method used in data mining, and it is also available in the Orange data mining tool. Logistic Regression is a statistical method that is used to analyze a dataset in which there are oneor more independent variables that determine an outcome**.**

SVM : Support Vector Machine (SVM) is a popular machine learning algorithm used for classification and regression tasks, and it is also available in the Orange data mining tool. SVM is a powerful algorithm that is particularly effective when working with complex, high-dimensional datasets.

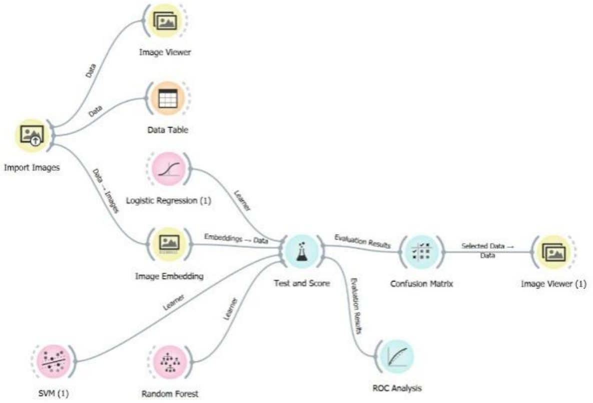
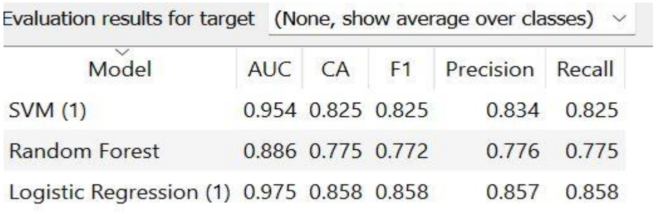


Figure 8. Implementation of three Classification Methods

Above figure 2 shows the implementation of three classification methods on set of data of rice disease through which we can find that which classification method will provide us the exact result .

VI. RESULTS AND DISCUSSION



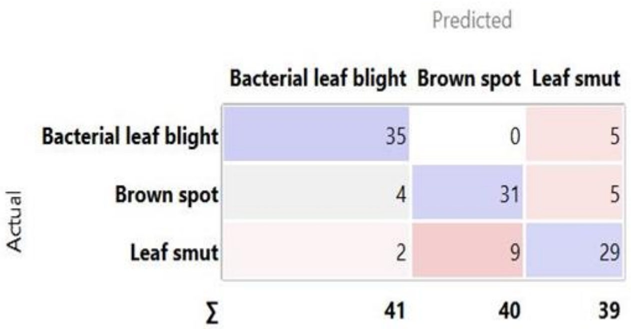
Evaluation results for target (None, show average over classes)

Model	AUC	CA	F1	Precision	Recall
SVM (1)	0.954	0.825	0.825	0.834	0.825
Random Forest	0.886	0.775	0.772	0.776	0.775
Logistic Regression (1)	0.975	0.858	0.858	0.857	0.858

Figure 9. Evaluation Results for Target

The evaluation result plays an important role for evaluating the methods that will be exact .Accordingly from the above figure 3 we can see that the logistic regression method provides the most exact classification accuracy on the rice disease dataset which can vary while Random Forest method provide least classification accuracy of the dataset.

Using Random Forest:-



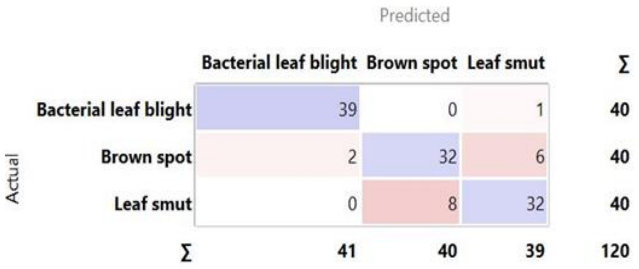
Predicted

Actual	Bacterial leaf blight	Brown spot	Leaf smut
Bacterial leaf blight	35	0	5
Brown spot	4	31	5
Leaf smut	2	9	29
Σ	41	40	39

Figure 10. Predicted and Actual Result for Random Forest

The predicted and actual result of image classification using random forest can be seen in figure 4 where the actual result of bacterial leaf blight, brown spot and leaf smut are predicted with accuracy of 0.775.

Using Logistic Regression:-



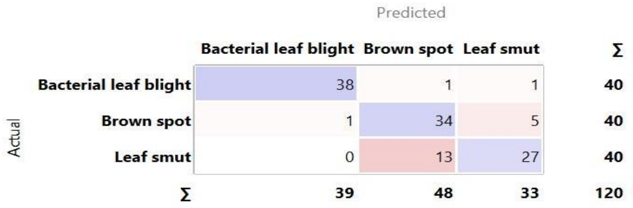
Predicted

Actual	Bacterial leaf blight	Brown spot	Leaf smut	Σ
Bacterial leaf blight	39	0	1	40
Brown spot	2	32	6	40
Leaf smut	0	8	32	40
Σ	41	40	39	120

Figure 11. Predicted and Actual Result for Logistic Regression

The predicted and actual result of image classification using Logistic Regression can be seen in figure 5 where the actual result of bacterial leaf blight, brown spot and leaf smut are predicted with accuracy of 0.858.

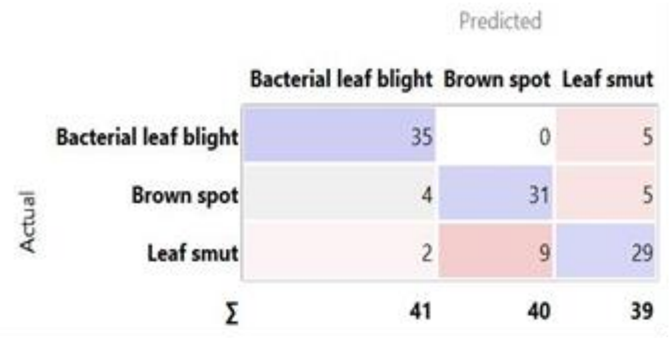
Using SVM



Predicted

Actual	Bacterial leaf blight	Brown spot	Leaf smut	Σ
Bacterial leaf blight	38	1	1	40
Brown spot	1	34	5	40
Leaf smut	0	13	27	40
Σ	39	48	33	120

Figure 12. Predicted and Actual Result for SVM



		Predicted		
		Bacterial leaf blight	Brown spot	Leaf smut
Actual	Bacterial leaf blight	35	0	5
	Brown spot	4	31	5
	Leaf smut	2	9	29
	Σ	41	40	39

In Figure 12, the real results of bacterial leaf blight, brown spot, and leaf smut are predicted with an accuracy of 0.825. This illustrates the difference between the anticipated and actual results of image classification using SVM.

VII.CONCLUSION

Results from using the Orange tool to analyse data and make decisions have been positive. The application offers a comprehensive solution for digesting difficult data sets and obtaining intelligent insights, thanks to its user-friendly interface and robust analytical capabilities. Customers have used Orange to apply a wide range of statistical and machine learning techniques to their data in order to reveal previously hidden patterns and trends. Organisational performance and outcomes may improve over time as a result of this phenomena. The Orange tool's successful implementation demonstrates its potential for use in a wide variety of industries and settings. Improvements to the tool's features and compatibility with other data analysis and visualisation programmes are two possible directions for further research.

References:

[1] Žnidaršič, M., & Podgorelec, V. (2009). Visual programming for predictive modeling using Orange. Journal of Machine Learning Research, 10, 1239-1242.

[2] Demšar, J. (2013). Orange canvas: data mining visual programming tool. Bioinformatics, 29(11), 145-147.

[3] Dey, N., Ashour, A. S., Shi, F., & Ashour, A. S. (2016). Comparative analysis of data mining tools. Procedia Computer Science, 89, 114-121.

[4] Suto, H., Hanai, M., & Okumura, M. (2020). A survey on visual programming languages in data science. Journal of Information Processing, 28, 178-191.

[5] Khetani, V. ., Gandhi, Y. ., Bhattacharya, S. ., Ajani, S. N. ., & Limkar, S. . (2023). Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. International Journal of Intelligent Systems and Applications in Engineering, 11(7s), 253–262.

[6] Sable, N. P., Shende, P., Wankhede, V. A., Wagh, K. S., Ramesh, J. V. N., & Chaudhary, S. (2023). DQSCTC: design of an efficient deep dyna-Q network for spinal cord tumour classification to identify cervical diseases. Soft Computing, 1-26.

[7] V. Khetani, Y. Gandhi and R. R. Patil, "A Study on Different Sign Language Recognition Techniques," 2021 International Conference on Computing,

Communication and Green Engineering (CCGE), Pune, India, 2021, pp. 1-4, doi: 10.1109/CCGE50943.2021.9776399.

[8] R. Patil Rashmi, Y. Gandhi, V. Sarmalkar, P. Pund and V. Khetani, "RDPC: Secure Cloud Storage with Deduplication Technique," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2020, pp. 1280-1283, doi: 10.1109/I-SMAC49090.2020.9243442.

[9] Kharate, N., Patil, S., Shelke, P., Shinde, G., Mahalle, P., Sable, N., & Chavhan, P. G. (2023). Unveiling the Resilience of Image Captioning Models and the Influence of Pre-trained Models on Deep Learning Performance. International Journal of Intelligent Systems and Applications in Engineering, 11(9s), 01-07.

[10] Andrzej Wędzik (2023). Hybrid Energy Systems: Synergies and Optimization Strategies. Acta Energetica, (01), 01–07.

[11] Geron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

[12] Girraj Kumar Jangid, Ashish Kumar Sharma, & L. N. Balai. (2023). Investigations on PAPR and SER Performance Analysis of OFDMA and SCFDMA under Different Channels. International Journal on Recent Technologies in Mechanical and Electrical Engineering, 10(1), 28–35.

[13] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[14] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

[15] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).

[16] Regularity properties of integral problems for wave equations and applications. (2023). Advances in the Theory of Nonlinear Analysis and Its Application, 7(1), 82-102.