Journal of Innovation and Computer Science (JICS)
Vol. 1, No. 2, April 2025
https://journal.itqanpreneurs.com/index.php/jics/index

Classification of Software Defect Prediction for Bisnissyariah.co.id Media Portal using Machine Learning Technology

Fikri Ismaya[1], Windu Gata[2], Muhammad Romadhona Kusuma[3], Dedi Dwi Saputra[4], Sigit Kurniawan[5]

[1]Nusa Mandiri University, Indonesia. E-mail: 14207089@nusamandiri.ac.id
[2]Nusa Mandiri University, Indonesia. E-mail: windu@nusamandiri.ac.id
[3]Darunnajah University, Indonesia. E-mail: m.romadhona.kusuma@darunnajah.ac.id
[4]Indonesia Cyber University, Indonesia. E-mail: dedi.dwi.s@cyber-univ.ac.id
[5]Muhammadiyah University of Technology Jakarta, Indonesia. E-mail: sigit@utmj.ac.id

Article History:
Received: 17-02-2025
Revised: 20-04-2025
Accepted: 21-04-2025

Keywords: *Software Defect Prediction, Bisnissyariah Dataset, Machine Learning Models, Random Forest.*

Abstract: *This research was conducted to develop software defect prediction using a dataset from Bisnissyariah, a forum website with up-to-date news related to Islamic business. The study employed a straightforward research design to ensure easy comprehension for the readers. Machine learning models, including Random Forest, Gradient Boosting, and Support Vector Machine, were utilized in this research—the application of these models aimed to evaluate and compare the accuracy of software defect predictions. The research findings indicated that the Random Forest model outperformed the others, achieving an accuracy rate of 96.7%. This result shows the high effectiveness of the Random Forest model in predicting software defects based on data from Bisnissyariah. In addition, these findings significantly impact the development of more reliable and high-quality software, particularly in Islamic business. This research makes a valuable contribution to enhancing our understanding of software defect prediction using data from sources like Bisnissyariah.*

Introduction

Some standard methods in software testing (Wibisono, W., & Baskoro, 2023). First, Unit Testing tests the smallest part of the software, such as a function or procedure, separately, usually done by the developer and often using automation. Second is Integration Testing, a combined test of several units to ensure the interface between units works appropriately. Third is System Testing, which tests the entire system to ensure the software meets functional and non-functional requirements. Fourth is acceptance testing, which is the final stage of testing involving users ensuring the system meets their needs, including alpha and beta testing. Fifth is Regression Testing, which is carried out after changes or updates to the software to ensure that no functions are damaged due to the modification. Sixth, Automated Testing, which is the use of automated tools and scripts to run repeated tests efficiently and consistently, especially in large projects that require periodic testing.

In an increasingly digital era, data has become integral to every aspect of life. In this context, Bisnissyariah.co.id, a national news portal founded on the 17th of Ramadan in 2014, provides a wide range of information, from Sharia economics to the developments in Muslim businesses in Indonesia. Aligned with their mission to support the rise of Muslim entrepreneurs in Indonesia, they aim to enrich the press and social discourse in the country. With the advancement of technology, companies have undergone significant changes in system design and development since March 2020. This transformation has included adapting to current technological developments, such as incorporating internet television features and more. However, as data volume increases and the need for better insights grows, effective data management has become a challenge that needs to be addressed. Defect Software is failures or errors in software systems that result in the system not functioning as expected or producing undesirable outcomes. In the context of Bisnissyariah.co.id, software defects can impact the quality and efficiency of the services they provide to users. Therefore, predicting and managing these defects before they adversely affect the system is crucial. Software defect prediction is an area that has garnered significant attention in research and industry practice. This study aims to classify software defects on the Sharia Business Media Portal using machine learning technology. This technology will enable us to predict and manage defects more effectively, enhancing service quality and efficiency.

In this research, the authors will employ various machine learning methods, including Random Forest, Support Vector Machine (SVM), and Gradient Boosting. These methods were chosen based on their ability to classify data, consider relevant features, and make accurate predictions. The expected outcome is that these results can be utilized to improve the system and prevent defects in the future. Furthermore, this research will also delve into developing and implementing this technology within the Bisnissyariah.co.id system. The authors hope these findings will provide valuable insights for developing a better and more efficient system and contribute to a deeper understanding of how machine learning can be utilized in software defect prediction. Lastly, this background also underscores the importance of this research in a broader context. By understanding how machine learning technology can be used to predict and manage software defects, we will enhance the quality and efficiency of Bisnissyariah.co.id and contribute to driving innovation and progress in the broader Sharia Business community. Therefore, this research will provide a meaningful and positive contribution to the development and advancement of Sharia businesses in Indonesia.

Bisnissyariah.co.id is one of the national news portals established on the 17th of Ramadan in 2014. It not only provides news about Sharia economics, Hajj and Umrah, Muslim tourism, Zakat, and the evolution of Muslim businesses in Indonesia but also offers various general information needed by the wider community, including health, sports, football, movies, law, celebrities, entertainment, technology, education, and international issues. Occupying a central position in the community to support the rise of Muslim entrepreneurs in Indonesia, we earnestly hope that the presence of this national news portal can contribute to expanding the perspective of the press and social control by the mandate of law number 40 of 1999, thus serving as a reference source for the public and policymakers in the country.

Starting on March 1, 2020, the Bisnis Syariah.co.id website underwent a design transformation and system renewal under the management of PT Dahlia Multi Usaha, a newspaper publisher with company registration number 9120200412637. This transformation included transitioning from the old WordPress platform to a more modern one to adapt to current technological advancements. This transformation included adding features like Bisnis Syariah.tv, an internet television platform, and others.

Machine learning is a branch of artificial intelligence (AI) that focuses on developing and applying algorithms that enable computers to learn from and make decisions or predictions based on data. In other words, machine learning is the process through which computer systems sharpen their abilities through experience without explicit programming. Machine learning can identify patterns in input data and create predictive models. Through a process known as "training," computers can take data, learn from it, and then apply the knowledge gained to make decisions or predictions.

There are various types of machine learning, including supervised learning, where models are built based on labeled training data; unsupervised learning, where models are built based on unlabeled training data; and semi-supervised learning, which combines elements of both. A related concept is "deep learning," where artificial neural networks with multiple "deep" layers are used to model and understand complex data. Therefore, machine learning is an important technology that allows computers to learn from experience, identify patterns in data, and use this knowledge to make decisions or predictions about new data or previously unseen situations (Yang & Shami, 2020 and Dogan & Birant, 2021).

Software engineering is a field of expertise that focuses on designing and developing software systems and computer applications. It encompasses the entire software lifecycle process, from the initial conception to design, coding, testing, maintenance, and quality management stages. This discipline utilizes engineering principles to achieve reliable, efficient, and effective software. Software engineering views software not only as a final product but also as a process that involves systematic and organized stages. One of the key concepts in software engineering is "requirements engineering," which involves identifying user needs and translating them into technical specifications.

The field also involves "software design," where software architecture is built and defined. "Programming" is the stage for converting technical specifications into software code. "Testing" is the stage where the quality and reliability of the software are ensured. Finally, "maintenance" is when software is updated and improved to meet changing needs. Therefore, software engineering is a holistic and systematic discipline that focuses not only on software creation but also on the processes involved and how the software is integrated and managed within a larger system (Serban et al., 2020; Davoudian & Liu, 2020; Amershi et al., 2019).

Random Forest is a machine-learning method that involves an ensemble of decision trees. In other words, it is an ensemble method that combines predictions from multiple small models to produce more accurate and stable predictions. Each tree in the Random Forest is generated from randomly selected data samples with replacement (bootstrapping), and each tree makes independent predictions. Ultimately, the predictions from all the trees are averaged (for regression) or chosen through voting (for classification) to produce the final prediction.

One of the key advantages of Random Forest is that it mitigates overfitting, which is a common issue with individual decision trees. A Random Forest can achieve better and more stable performance by using many trees and averaging their predictions. Furthermore, Random Forest has other important features, such as handling both categorical and numerical variables, dealing with missing data, and providing measures of feature importance that can be used for feature selection. Therefore, Random Forest is a powerful machine-learning technique that combines predictions from many decision trees to generate more accurate and stable (Speiser et al., 2019; Khammas, 2020; Adetunji et al., 2022).

Support Vector Machine, abbreviated as SVM, is a machine learning technique for classification and regression. The main idea of this method is to find the best hyperplane or "path" that separates data into different classes. SVM calculates the optimal distance between the separating line (or hyperplane) and the nearest data points from each class, known as support vectors. The selected line or hyperplane has the farthest distance or the most significant margin to these support vectors.

SVM has advantages in handling high-dimensional data and offers flexibility through kernel functions, allowing for more complex separation than just a straight line or flat plane. This can be very useful in many applications where data may not be separable linearly. However, SVM may be more challenging to understand and interpret than other methods and can also be slower and require more memory for large-sized data (Cervantes et al., 2020; Tao et al., 2020; Hamid et al., 2021).

Feature selection is choosing a subset of the most relevant features or attributes from a dataset for machine learning model creation. The primary goal of this process is to enhance the model's efficiency and reduce overfitting, where the model becomes too specific to the training data and performs poorly on unseen data. In principle, not all features in a dataset will be beneficial for prediction. Some features may be irrelevant or disrupt the model's performance. A simpler and more efficient model can be created by reducing the data's dimensionality (the number of features) without losing too much information.

Various techniques areVarious techniques are used in feature selection, including filter, wrapper, and embedded methods. Filter methods evaluate features based on descriptive statistics and select the best features before the modeling process, while wrapper methods evaluate features based on predictive model performance. Embedded methods, on the other hand, perform feature selection as part of the learning algorithm (Rostami et al., 2021; Nguyen et al., 2020; Maleki et al., 2021).

Information Gain is a metric used in data science and machine learning to determine which features are most valuable for making predictions based on a dataset. This concept is derived from information theory and is frequently used in algorithms like decision trees. Information Gain is measured by the reduction in entropy, which measures data uncertainty or randomness. If a feature reduces the entropy, it is considered to have high information gain. To calculate information gain, we subtract the entropy after a split from the entropy before the split. The result measures how much uncertainty has been reduced by that feature (Omuya et al., 2021; Thanoon et al., 2019; Arnavaz et al., 2020).

Cross-validation is one of the crucial techniques in assessing model performance in machine learning. It measures how well a model can generalize patterns learned from the training data to unseen data. The fundamental concept of cross-validation involves splitting the dataset into two main subsets: the training and testing datasets. The data is further divided into folds, typically called "Folds." The model is then trained on a portion of the data, for example, 80%, and its performance is tested on the 20% of data that has never been seen before. This process is repeated several times, with different folds used as the testing data, and the results are averaged to provide a more stable estimate of how well the model generalizes. Cross-validation helps avoid overfitting and ensures the resulting model is more reliable. There are several methods of cross-validation, one of which is k-fold cross-validation, where the data is divided into k folds. The model is trained and tested k times, with each fold used as the testing data one by one (Liu et al., 2020; Nti et al., 2021; Kerbaa et al., 2019).

The previous research by Ran Li, Lijuan Zhou, Shudong Zhang, and others, titled 'Software Defect Prediction Based on Ensemble Learning,' utilized the NASA MDP dataset to make predictions using algorithm methods like C4.5, AdaBoost, Bagging (Bootstrap Aggregating), RSM (Random Subspace Method), RF (Random Forest), and Vote. The best performance was achieved by Random Forest, with an accuracy of 0.987 (98.7%) (Li et al., 2019). The previous research by Lakshmi Prabha and Shivakumar, titled 'Software Defect Prediction Using Machine Learning Technique,' utilized the NASA MDP dataset to make predictions using algorithm methods like Support Vector Classifier, Random Forest, and Naive Bayes. Support Vector Classifier and Random Forest achieved the best performance with an accuracy of 0.984 (98.4%) (Prabha & Shivakumar, 2020).

The previous research by Ruchika Malhotra and Kishwar, titled 'A Study of Software Defect Prediction Using Feature Extraction Techniques,' utilized the PROMISE dataset to make predictions using the Support Vector Machine algorithm. The Support Vector Machine achieved the best performance with an accuracy of 0.987 (98.7%) (Malhotra & Khan, 2020). The review of previous research shows that several machine-learning methods are commonly used for prediction. The novelty generated from this research is the prediction of the software defects in Bisnis Syariah.co.id and the extent of the scores obtained by utilizing machine learning methods and comparing the results with previous research. The findings of this research can serve as a reference in the decision-making process for the editorial leadership.

Method

The research method is a scientific process applied to obtain information to achieve research objectives. This means that this activity involves a structured study of a particular phenomenon. Research design illustrates the steps in researching as outlined in Figure 1.

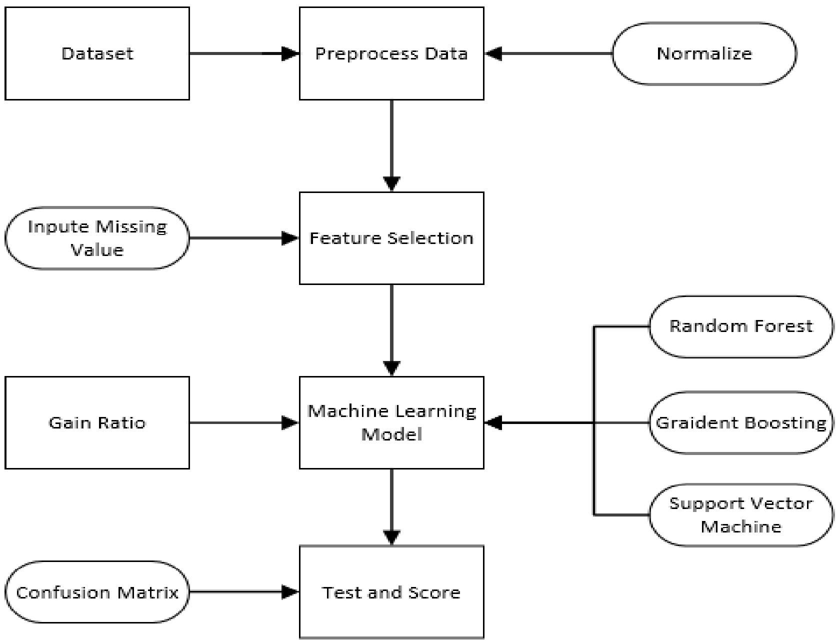


Figure 1. Research Design

Here is the description of the research design. Dataset: The initial stage involves collecting relevant data for the study. The gathered data can encompass various aspects of software, such as the number of lines of code, cyclomatic complexity, and others. These aspects may have an impact on defects. Preprocess Data: After collecting the data, the next step is to process and clean the data. This stage involves filling in missing values (Impute Missing Values) and normalizing features. The primary goal is to prepare the data for processing by a machine learning model.

In this stage, feature selection uses the Gain Ratio on the relevant dataset. The steps involve calculating the Gain Ratio for each feature, selecting a subset of features with the highest Gain Ratio values, and evaluating model performance. Modeling Machine Learning: The next step involves using machine learning models. Techniques such as SVM, Random Forest, and Gradient Boosting can be applied in this stage. The selected model is trained using the previously processed data. Test and Score: After training the model, its performance and accuracy are tested using techniques like the confusion matrix. The results of this stage can provide insights into how well the model can predict software defects.

Result and Discussion

In the results and discussion section, the research findings and discussions conducted by the researcher during the research process will be presented, with the title 'Classification of Software Defect Prediction in the Islamic Business Media Portal Using Machine Learning Technology. This study utilizes the primary dataset from private Bisnissyariah.co.id and also incorporates other datasets such as NASA. Pre-processing steps are conducted while considering several initial criteria to be observed, including the number of attributes, the number of modules, and the number of defects in each dataset. The description of the datasets used in this research is as follows:

Table 1. Specifications of Dataset

Dataset	Number of Data (Instance)	Number of Attribute (Feature)
Dataset_Fix_BISNISSYARIAH.csv	61	44

Table 1 Explanation of Attributes in the Dataset_Fix_BISNISSYARIAH.csv

No	Attribute (*Feature*)	Description
1.	Name	The name of the entity
2.	No	Unique identification number
3.	Methods	The number of methods or functions in a class
4.	nbMethodsIncludingGettersSetters	The total number of methods, including getter and setter methods
5.	nbMethods	Total number of methods
6.	nbMethodsPrivate	Number of private methods
7.	nbMethodsPublic	Number of methods that can be accessed publicly
8.	nbMethodsGetter	Number of 'getter' methods used to retrieve variable values
9.	nbMethodsSetters	Number of 'setter' methods used to set variable values
10.	Wmc	Weighted method complexity metric measuring the complexity of a class
11.	Ccn	Cyclomatic Complexity Number, measuring the complexity of logic in the code

No	Attribute (*Feature*)	Description
12.	ccnMethodMax	The highest cyclomatic complexity value of all methods in a class
13.	Externals	Number of external methods or classes accessed by this class
14.	Parents	Number of parent classes
15.	Implements	Number of interfaces implemented by this class
16.	Icom	Inter-class communication
17.	Length	Length of program code in lines
18.	Vocabulary	Number of unique words in the code
19.	Volume	Total size of the code in bytes
20.	Difficulty	Difficulty level in understanding and modifying the code
21.	Effort	Effort required to understand and modify the code
22.	Level	Code readability or comprehensibility level
23.	Time	Time required to understand and modify the code
24.	intelligentContent	Amount of information contained in the code, based on the number of unique operators and operands
25.	Number_operators	Number of operators in the code
26.	Number_operands	Number of operands in the code
27.	Number_operators_unique	Number of unique operators in the code
28.	Number_operands_unique	Number of unique operands in the code
29.	Cloc	Number of code lines containing comments
30.	Loc	Total number of code lines
31.	lloc	Number of code lines containing instructions
32.	Mi	Code readability metric
33.	mlwoC	Code readability metric without considering comments
34.	commentWeight	Weight of comments in the code
35.	relativeStructuralComplexity	Relative structural complexity of this class
36.	relativeDataComplexity	Relative data complexity of this class
37.	relativeSystemComplexity	Relative system complexity of this class
38.	totalStructuralComplexity	Total structural complexity of this class
39.	totalDataComplexity	Total data complexity of this class
40.	totalSystemComplexity	Total system complexity of this class
41.	pageRank	A metric indicating the importance of this class in the overall system
42.	afferentCoupling	Number of other classes depending on this class
43.	efferentCoupling	Number of other classes that this class depends on
44.	defect	The number of complex defects or non-defects

Preprocessing Data

The next phase is to test the dataset with several classification methods including Random Forest, Gradient Boosting, and Support Vector Machine. Furthermore, the results of the three types of proposed algorithms will be assessed using Classification Accuracy (CA), F1-Score, Precision, Recall, Confusion Matrix. The tool used in testing the algorithm is the Orange Data Mining application. Orange Data Mining is a prediction application for machine learning and various data mining needs.

The steps in carrying out data preprocessing in Orange Data Mining are as follows. Firstly, Activate and start the Orange Data Mining application.

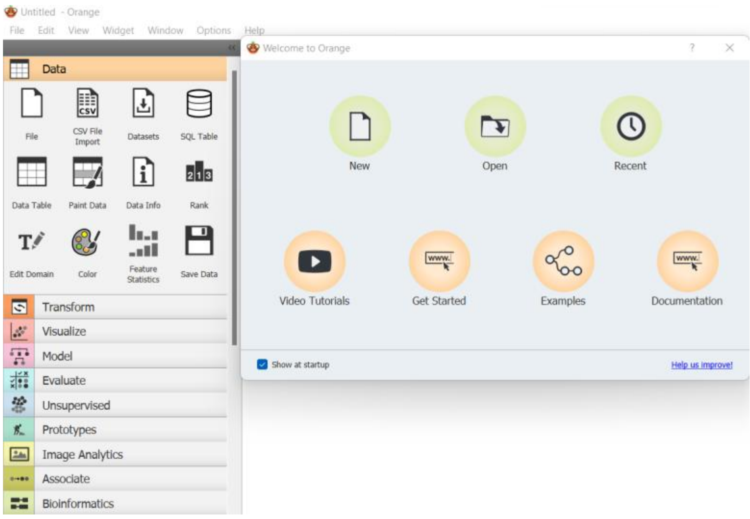


Figure 2. Orange Data Mining View

Figure 2 above shows the initial screen of the Orange Data Mining application which is used as a tool to test the Bisnissyariah.co.id private dataset, the author community web dataset, and the university web dataset using random sampling.

Secondly, click the open button on the Orange Data Mining application and select the file that has been saved with the .ows extension, then the following workflow will appear.

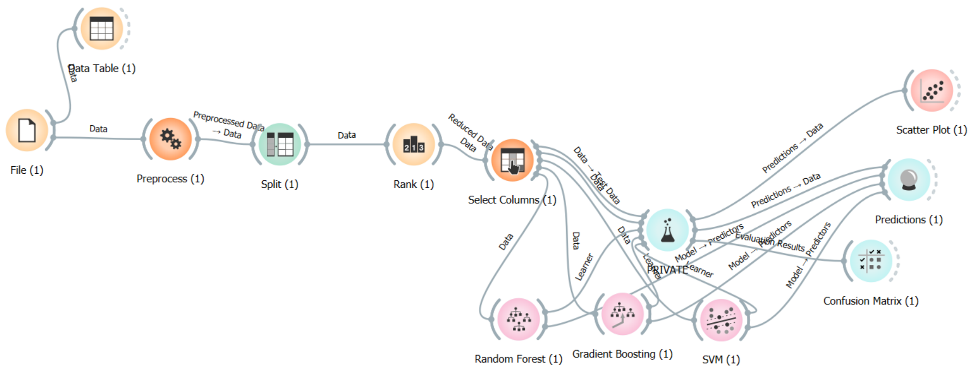


Figure 3. Orange Data Mining Workflow

In Figure 3, there are 3 datasets with the word “file”. Furthermore, each dataset will enter data preprocessing to clean up missing values, Normalize Feature to normalize features, then split to divide Defect and Non-Defect data, and Rank to do the best feature performance level from 1 to 20. Next, column selection is carried out to determine features and targets so that modeling is carried out with various machine learning. Furthermore, after modeling, the accuracy will appear.

Thirdly, Click on the “*Preprocess*” section.

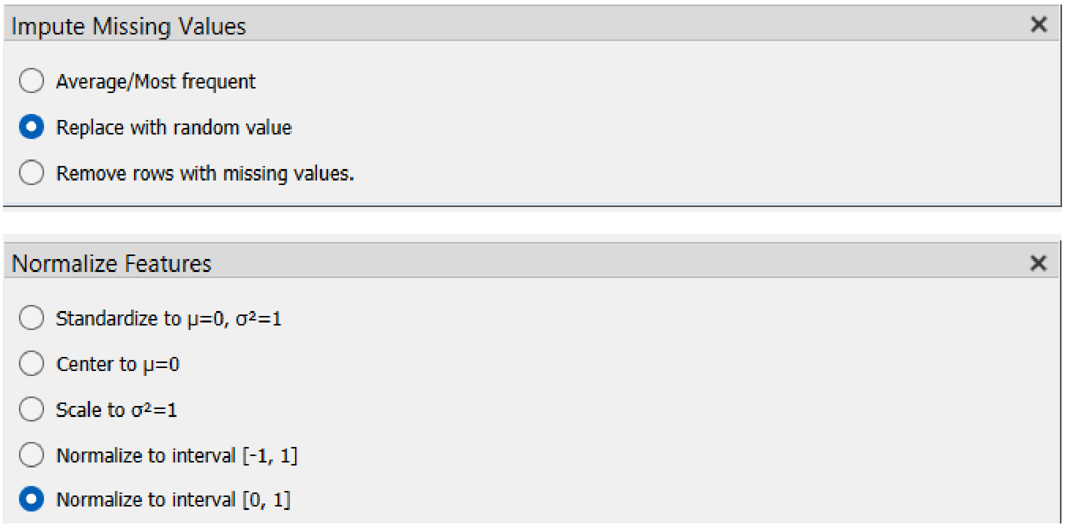


Figure 4. Preprocess example

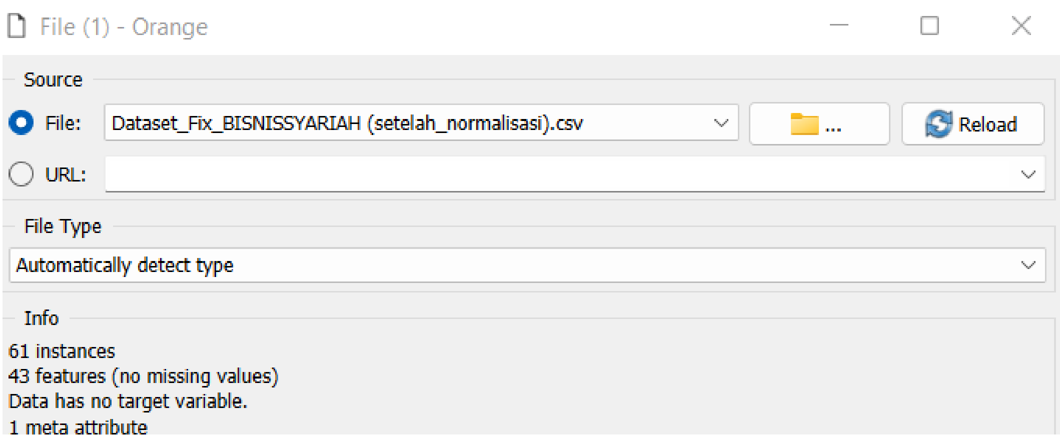
In Figure 4 above is an example of preprocessing done to clean the data, and normalizing the standard features.

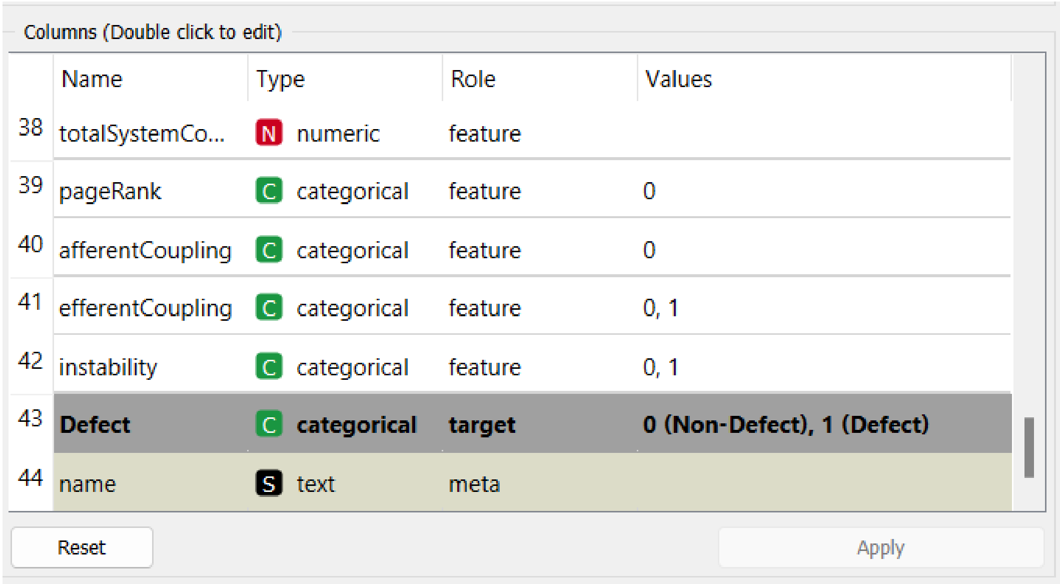
Experiment

In this section, several experiments will be conducted with the dataset provided. The following are the results of the experiments conducted

Experiment with Bisnissyariah.co.id Dataset

1. First, click File to open the Bisnissyariah.co.id dataset with the .csv extension.



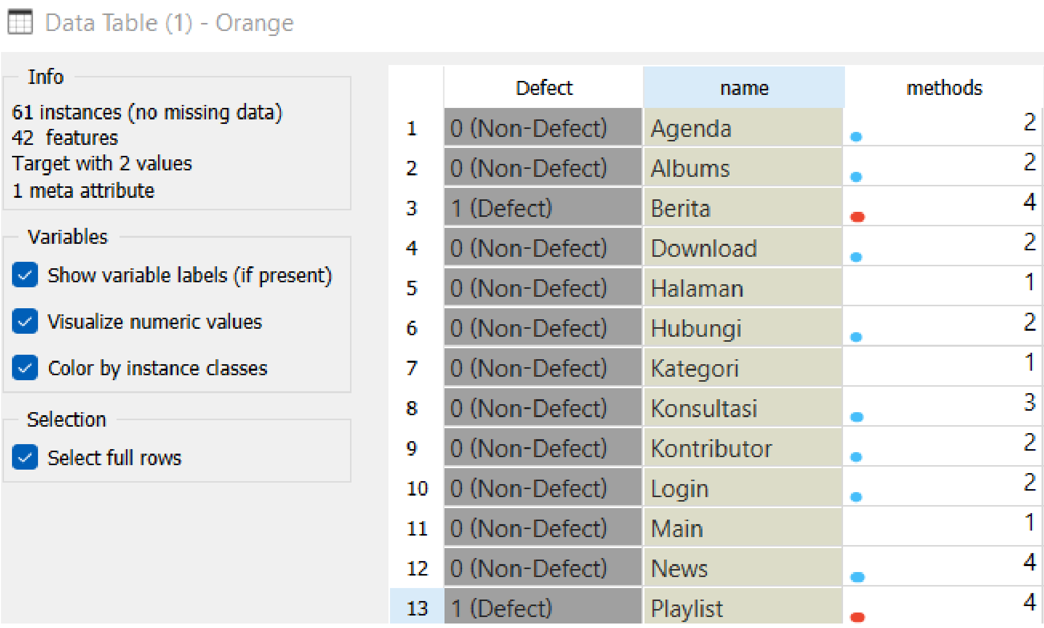


	Name	Type	Role	Values
38	totalSystemCo...	N numeric	feature	
39	pageRank	C categorical	feature	0
40	afferentCoupling	C categorical	feature	0
41	efferentCoupling	C categorical	feature	0, 1
42	instability	C categorical	feature	0, 1
43	**Defect**	C **categorical**	**target**	**0 (Non-Defect), 1 (Defect)**
44	name	S text	meta	

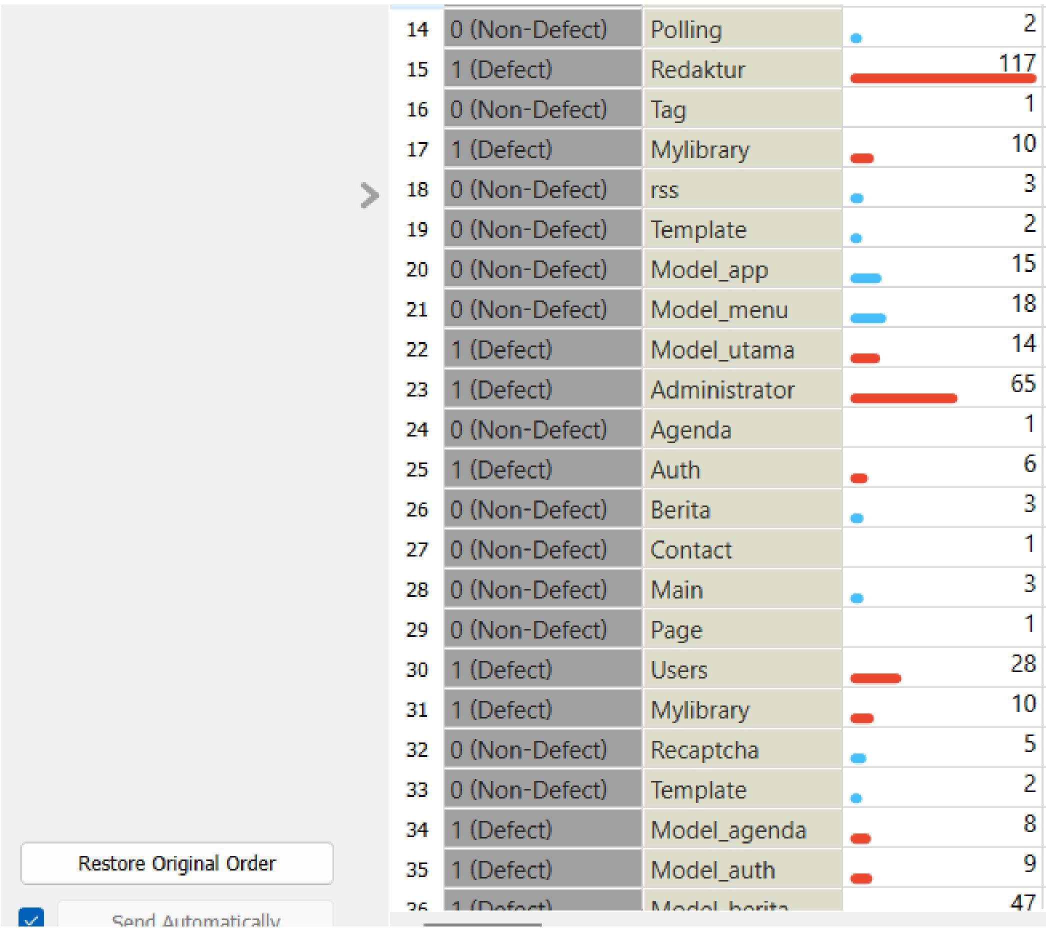
Figure 5. Dataset "File" Bisnissyariah.co.id

In Figure 5 above is a "File" filled with datasets from Bisnissyariah.co.id with file type .csv having the target "Defect" for research.

2. Click *Data Table*



	Defect	name	methods
1	0 (Non-Defect)	Agenda	2
2	0 (Non-Defect)	Albums	2
3	1 (Defect)	Berita	4
4	0 (Non-Defect)	Download	2
5	0 (Non-Defect)	Halaman	1
6	0 (Non-Defect)	Hubungi	2
7	0 (Non-Defect)	Kategori	1
8	0 (Non-Defect)	Konsultasi	3
9	0 (Non-Defect)	Kontributor	2
10	0 (Non-Defect)	Login	2
11	0 (Non-Defect)	Main	1
12	0 (Non-Defect)	News	4
13	1 (Defect)	Playlist	4



14	0 (Non-Defect)	Polling	2
15	1 (Defect)	Redaktur	117
16	0 (Non-Defect)	Tag	1
17	1 (Defect)	Mylibrary	10
18	0 (Non-Defect)	rss	3
19	0 (Non-Defect)	Template	2
20	0 (Non-Defect)	Model_app	15
21	0 (Non-Defect)	Model_menu	18
22	1 (Defect)	Model_utama	14
23	1 (Defect)	Administrator	65
24	0 (Non-Defect)	Agenda	1
25	1 (Defect)	Auth	6
26	0 (Non-Defect)	Berita	3
27	0 (Non-Defect)	Contact	1
28	0 (Non-Defect)	Main	3
29	0 (Non-Defect)	Page	1
30	1 (Defect)	Users	28
31	1 (Defect)	Mylibrary	10
32	0 (Non-Defect)	Recaptcha	5
33	0 (Non-Defect)	Template	2
34	1 (Defect)	Model_agenda	8
35	1 (Defect)	Model_auth	9
36	1 (Defect)	Model_berita	47

Figure 6. Dataset view of Bisnissyariah.co.id before Preprocess (1)

methods	nodsIncludingGetters	nbMethods	nbMethodsPrivate	nbMethodsPublic	nbMethodsGetter	nbMethodsSetters	wmc	ccn	ccnMethodMax
2	2	2	0	2	0	0	5	4	3
2	2	2	0	2	0	0	7	6	4
4	4	4	0	4	0	0	14	11	5
2	2	2	0	2	0	0	5	4	3
1	1	1	0	1	0	0	2	2	2
2	2	2	0	2	0	0	5	4	4
1	1	1	0	1	0	0	4	4	4
3	3	3	0	3	0	0	6	4	3
2	2	2	0	2	0	0	7	6	6
2	2	2	0	2	0	0	7	6	6
1	1	1	0	1	0	0	1	1	1
4	4	4	1	3	0	0	11	8	5
4	4	4	0	4	0	0	14	11	5
2	2	2	0	2	0	0	3	2	2
117	117	117	0	117	0	0	283	167	13
1	1	1	0	1	0	0	4	4	4
10	10	10	0	10	0	0	34	25	13
3	3	3	0	3	0	0	5	3	2
2	2	2	0	2	0	0	2	1	1
15	15	15	0	15	0	0	15	1	1
18	18	18	0	18	0	0	18	1	1
14	14	14	0	14	0	0	17	4	3

Figure 7. Dataset view of Bisnissyariah.co.id before Preprocess (2)

		ccnMethodMax	externals	parents	implements	lcom	length	vocabulary	volume	difficulty	effort	level	time	intelligentContent
1	4	3	1	1	0	1	129	45	708.45	7.62	5395.11	0.13	300	93.0
2	6	4	1	1	0	1	180	51	1021.04	9.20	9393.54	0.11	522	110.9
3	1	5	1	1	0	1	347	111	2357.66	9.05	21343.64	0.11	1186	260.4
4	4	3	1	1	0	1	102	39	539.11	6.91	3724.77	0.14	207	78.0
5	2	2	1	1	0	1	64	32	320.00	4.72	1511.11	0.21	84	67.7
6	4	4	1	1	0	1	110	66	664.88	3.77	2506.94	0.27	139	176.3
7	4	4	1	1	0	1	97	43	526.35	4.93	2597.11	0.20	144	106.6
8	4	3	1	1	0	1	147	69	897.95	3.66	3287.89	0.27	183	245.2
9	6	6	1	1	0	1	199	81	1261.63	6.56	8276.29	0.15	460	192.3
10	6	6	1	1	0	1	104	55	601.26	6.27	3770.41	0.16	209	95.8
11	1	1	1	1	0	1	15	9	47.55	1.43	67.93	0.70	4	33.2
12	8	5	1	1	0	1	153	63	914.52	8.73	7981.30	0.11	443	104.7
13	1	5	1	1	0	1	318	101	2117.31	9.35	19787.85	0.11	1099	226.5
14	2	2	1	1	0	1	85	30	417.09	6.50	2711.06	0.15	151	64.1
15	7	13	1	1	0	1	6430	540	58363.92	72.00	4201980.63	0.01	233443	810.6
16	4	4	1	1	0	1	92	39	486.26	5.15	2502.79	0.19	139	94.4
17	5	13	0	0	0	8	341	136	2416.82	13.89	33558.21	0.07	1864	174.0
18	3	2	0	0	0	1	95	35	487.28	6.52	3175.73	0.15	176	74.7
19	1	1	0	0	0	1	23	11	79.57	2.33	185.66	0.43	10	34.1
20	1	1	1	1	0	1	179	33	902.95	7.35	6636.66	0.14	369	122.8
21	1	1	1	1	0	1	179	41	959.00	4.13	3958.96	0.24	220	232.3
22	4	3	1	1	0	1	360	62	2143.51	30.63	65650.27	0.03	3647	69.9

Figure 8. Dataset view of Bisnissyariah.co.id before Preprocess (3)

	intelligentContent	number_operators	number_operands	mber_operators_unic	mber_operands_uniq	cloc	loc	lloc	mi	mIwoC	commentWeight	tiveStructuralComple	elativeDataComplexit
1	93.03	30	99	6	39	0	43	43	43.87	43.87	0.00	81	0.00
2	110.98	42	138	6	45	0	58	58	39.66	39.66	0.00	100	0.00
3	260.43	78	269	7	104	0	99	99	31.37	31.37	0.00	289	0.00
4	78.03	26	76	6	33	0	41	41	45.15	45.15	0.00	81	0.00
5	67.76	13	51	5	27	0	21	21	53.35	53.35	0.00	36	0.00
6	176.34	18	92	5	61	0	33	33	46.57	46.57	0.00	81	0.00
7	106.67	22	75	5	38	0	33	33	47.28	47.28	0.00	49	0.00
8	245.24	28	119	4	65	0	48	48	42.11	42.11	0.00	196	0.00
9	192.32	35	164	6	75	1	47	46	52.41	41.21	11.20	144	0.00
10	95.88	18	86	7	48	0	44	44	43.88	43.88	0.00	100	0.00
11	33.28	5	10	2	7	0	11	11	65.41	65.41	0.00	1	0.00
12	104.79	33	120	8	55	7	71	64	62.16	38.79	23.37	169	0.13
13	226.55	67	251	7	94	0	96	96	31.99	31.99	0.00	256	0.00
14	64.17	20	65	5	25	0	28	28	49.82	49.82	0.00	36	0.00
15	810.65	1020	5410	14	526	64	1953	1889	13.84	0.00	13.84	1681	0.05
16	94.47	22	70	5	34	0	33	33	47.52	47.52	0.00	49	0.00
17	174.06	99	242	14	122	12	152	143	47.01	25.93	21.08	1	11.00
18	74.77	32	63	6	29	4	63	58	61.34	42.31	19.03	1	1.50
19	34.10	2	21	2	9	0	15	15	60.90	60.90	0.00	4	1.33
20	122.85	32	147	3	30	0	83	83	37.31	37.31	0.00	225	1.11
21	232.30	18	161	2	39	0	84	84	37.01	37.01	0.00	36	1.91
22	69.99	76	284	11	51	0	118	118	30.94	30.94	0.00	144	1.32

Figure 9. Dataset view of Bisnissyariah.co.id before Preprocess (4)

	mi	mIwoC	commentWeight	tiveStructuralComple	elativeDataComplexit	ativeSystemComplex	talStructuralComplexi	totalDataComplexity	otalSystemComplexit	pageRank	afferentCoupling	efferentCoupling	violations
1	43.87	43.87	0.00	81	0.00	81.00	162	0.00	162.00	0	0	1	?
2	39.66	39.66	0.00	100	0.00	100.00	200	0.00	200.00	0	0	1	?
3	31.37	31.37	0.00	289	0.00	289.00	1156	0.00	1156.00	0	0	1	Probably bugg...
4	45.15	45.15	0.00	81	0.00	81.00	162	0.00	162.00	0	0	1	?
5	53.35	53.35	0.00	36	0.00	36.00	36	0.00	36.00	0	0	1	?
6	46.57	46.57	0.00	81	0.00	81.00	162	0.00	162.00	0	0	1	?
7	47.28	47.28	0.00	49	0.00	49.00	49	0.00	49.00	0	0	1	?
8	42.11	42.11	0.00	196	0.00	196.00	588	0.00	588.00	0	0	1	?
9	52.41	41.21	11.20	144	0.00	144.00	288	0.00	288.00	0	0	1	Probably bugg...
10	43.88	43.88	0.00	100	0.00	100.00	200	0.00	200.00	0	0	1	?
11	65.41	65.41	0.00	1	0.00	1.00	1	0.00	1.00	0	0	1	?
12	62.16	38.79	23.37	169	0.13	169.13	676	0.50	676.50	0	0	1	?
13	31.99	31.99	0.00	256	0.00	256.00	1024	0.00	1024.00	0	0	1	Probably bugg...
14	49.82	49.82	0.00	36	0.00	36.00	72	0.00	72.00	0	0	1	?
15	13.84	0.00	13.84	1681	0.05	1681.05	196677	5.57	196682.57	0	0	1	Too complex cl...
16	47.52	47.52	0.00	49	0.00	49.00	49	0.00	49.00	0	0	1	?
17	47.01	25.93	21.08	1	11.00	12.00	10	110.00	120.00	0	0	0	Too complex m...
18	61.34	42.31	19.03	1	1.50	2.50	3	4.50	7.50	0	0	0	?
19	60.90	60.90	0.00	4	1.33	5.33	8	2.67	10.67	0	0	0	?
20	37.31	37.31	0.00	225	1.11	226.11	3375	16.63	3391.63	0	0	1	?
21	37.01	37.01	0.00	36	1.91	37.91	648	34.43	682.43	0	0	1	?
22	30.94	30.94	0.00	144	1.32	145.32	2016	18.46	2034.46	0	0	1	Probably bugg...

Figure 10. Dataset view of Bisnissyariah.co.id before Preprocess (5)

Figures 7 through 10 above are the contents of the Bisnissyariah.co.id dataset which has 61 instances and 44 attributes.

3. Click *Preprocess* and *Rank*

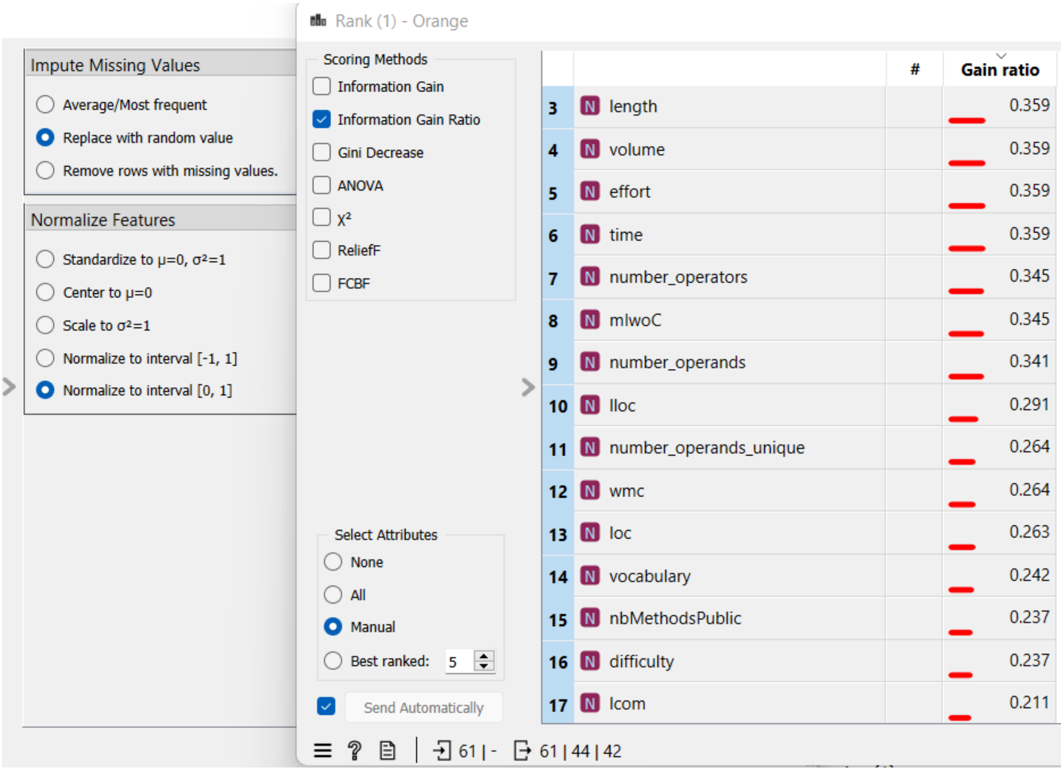
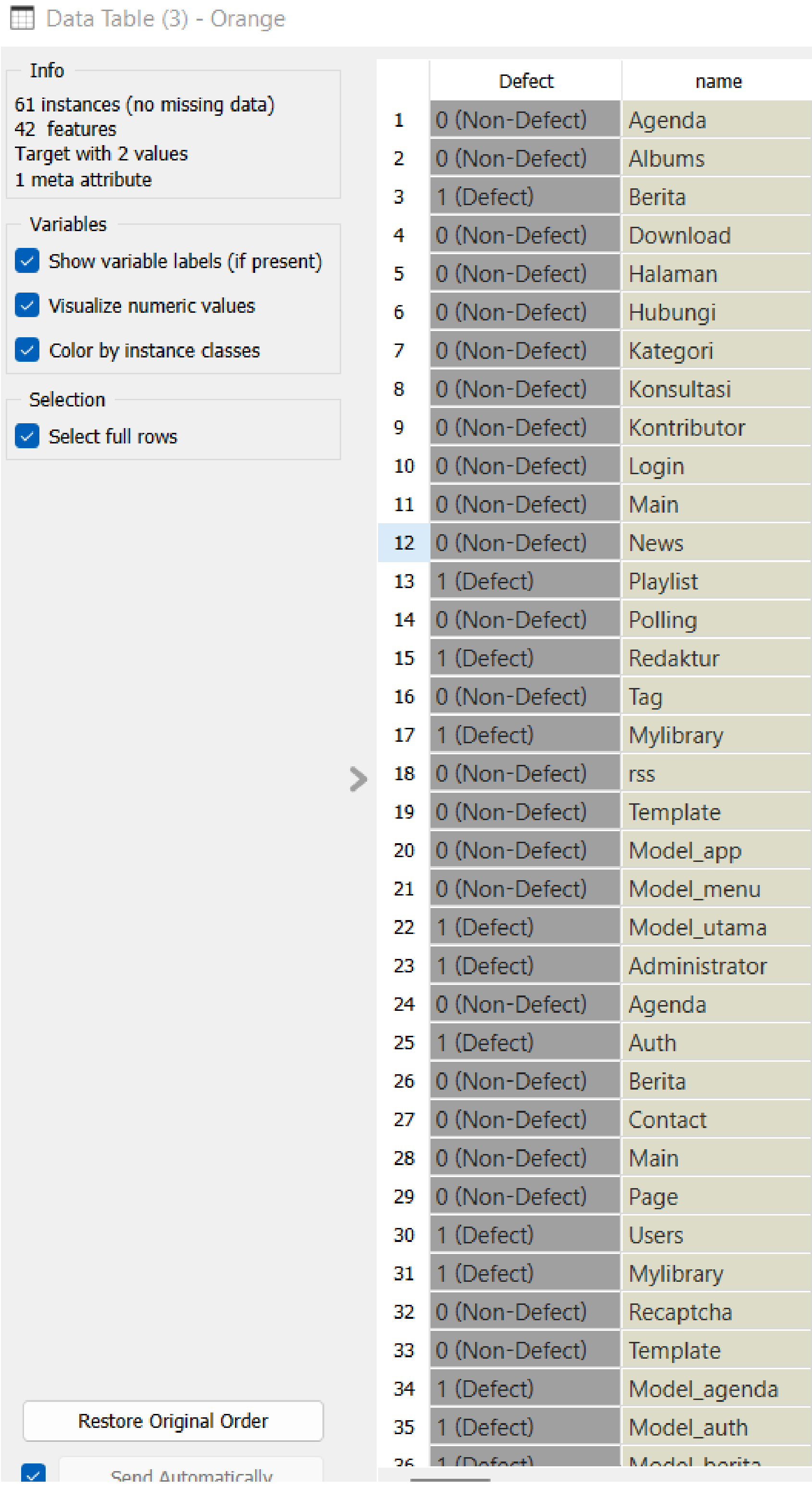


Figure 11. Preprocess and Rank Dataset of Bisnissyariah.co.id

In Figure 11 above is a Preprocess and Rank which is filled with several features from the Bisnissyariah.co.id dataset which functions to clean up datasets that have missing values, and normalize features by selecting standards. Followed by rank by using information gain to choose which features are good starting from 1 to 20.

4. *Click Data Table again to view the dataset that has been preprocessed.*



	Defect	name
1	0 (Non-Defect)	Agenda
2	0 (Non-Defect)	Albums
3	1 (Defect)	Berita
4	0 (Non-Defect)	Download
5	0 (Non-Defect)	Halaman
6	0 (Non-Defect)	Hubungi
7	0 (Non-Defect)	Kategori
8	0 (Non-Defect)	Konsultasi
9	0 (Non-Defect)	Kontributor
10	0 (Non-Defect)	Login
11	0 (Non-Defect)	Main
12	0 (Non-Defect)	News
13	1 (Defect)	Playlist
14	0 (Non-Defect)	Polling
15	1 (Defect)	Redaktur
16	0 (Non-Defect)	Tag
17	1 (Defect)	Mylibrary
18	0 (Non-Defect)	rss
19	0 (Non-Defect)	Template
20	0 (Non-Defect)	Model_app
21	0 (Non-Defect)	Model_menu
22	1 (Defect)	Model_utama
23	1 (Defect)	Administrator
24	0 (Non-Defect)	Agenda
25	1 (Defect)	Auth
26	0 (Non-Defect)	Berita
27	0 (Non-Defect)	Contact
28	0 (Non-Defect)	Main
29	0 (Non-Defect)	Page
30	1 (Defect)	Users
31	1 (Defect)	Mylibrary
32	0 (Non-Defect)	Recaptcha
33	0 (Non-Defect)	Template
34	1 (Defect)	Model_agenda
35	1 (Defect)	Model_auth

Figure 12. Preprocessed Bisnissyariah.co.id Dataset Display (1)

mIwoC	lcom	loc	lloc	length	volume
0.670693	0.0	0.016478	0.017039	0.017771	0.011333
0.606329	0.0	0.024202	0.025027	0.025721	0.0166933
0.479590	0.0	0.045314	0.046858	0.051754	0.0396134
0.690261	0.0	0.015448	0.015974	0.013562	0.00842919
0.815625	0.0	0.005149	0.005325	0.007638	0.00467193
0.711971	0.0	0.011329	0.011715	0.014809	0.0105859
0.722825	0.0	0.011329	0.011715	0.012783	0.00821039
0.643785	0.0	0.019053	0.019702	0.020577	0.0145825
0.630026	0.0	0.018538	0.018637	0.028683	0.0208189
0.670845	0.0	0.016993	0.017572	0.013874	0.00949493
1.0000	0.0	0.0000	0.0000	0.0000	0
0.593029	0.0	0.030896	0.028222	0.021512	0.0148667
0.489069	0.0	0.043769	0.045261	0.047233	0.0354919
0.761657	0.0	0.008754	0.009052	0.010912	0.00633681
0.0000	0.0	1.0000	1.0000	1.0000	1
0.726494	0.0	0.011329	0.011715	0.012003	0.00752293
0.396423	1.0	0.072606	0.070288	0.050818	0.0406279
0.646843	0.0	0.026777	0.025027	0.012471	0.00754042
0.931050	0.0	0.002060	0.002130	0.001247	0.000549074
0.570402	0.0	0.037075	0.038339	0.025565	0.0146683
0.565816	0.0	0.037590	0.038871	0.025565	0.0156294
0.473016	0.0	0.055098	0.056976	0.053780	0.0359412

Figure 13. Preprocessed Bisnissyariah.co.id Dataset Display (2)

difficulty	effort	time	number_operators	number_operands	ccn
0.087714	0.0012678	0.001268	0.027505	0.016481	0.01807
0.110103	0.00221937	0.002219	0.039293	0.023704	0.03012
0.107978	0.00506334	0.00506342	0.074656	0.047963	0.06024
0.077653	0.00087028	0.000869606	0.023576	0.012222	0.01807
0.046620	0.000343458	0.000342702	0.010806	0.007593	0.00602
0.033159	0.000580452	0.00057831	0.015717	0.015185	0.01807
0.049596	0.000601912	0.000599728	0.019646	0.012037	0.01807
0.031600	0.000766308	0.000766796	0.025540	0.020185	0.01807
0.072694	0.00195348	0.0019534	0.032417	0.028519	0.03012
0.068584	0.000881142	0.000878174	0.015717	0.014074	0.03012
0.0000	0	0	0.002947	0.0000	0.000
0.103443	0.00188328	0.00188058	0.030452	0.020370	0.04217
0.112229	0.00469308	0.00469073	0.063851	0.044630	0.06024
0.071844	0.00062903	0.000629715	0.017682	0.010185	0.00602
1.0000	1	1	1.0000	1.0000	1.000
0.052714	0.000579465	0.00057831	0.019646	0.011111	0.01807
0.176562	0.00797025	0.00796782	0.095285	0.042963	0.14458
0.072127	0.000739616	0.000736809	0.029470	0.009815	0.01205
0.012753	2.80182e-05	2.57026e-05	0.0000	0.002037	0.000
0.083888	0.00156327	0.00156358	0.029470	0.025370	0.000
0.038260	0.000926014	0.000925295	0.015717	0.027963	0.000
0.413774	0.0156077	0.0156058	0.072692	0.050741	0.01807

Figure 14. Preprocessed Bisnissyariah.co.id Dataset Display (3)

Figures 12 to 14 show the dataset that has been preprocessed to remove missing values and normalize the data.

5. Click *Select Column*

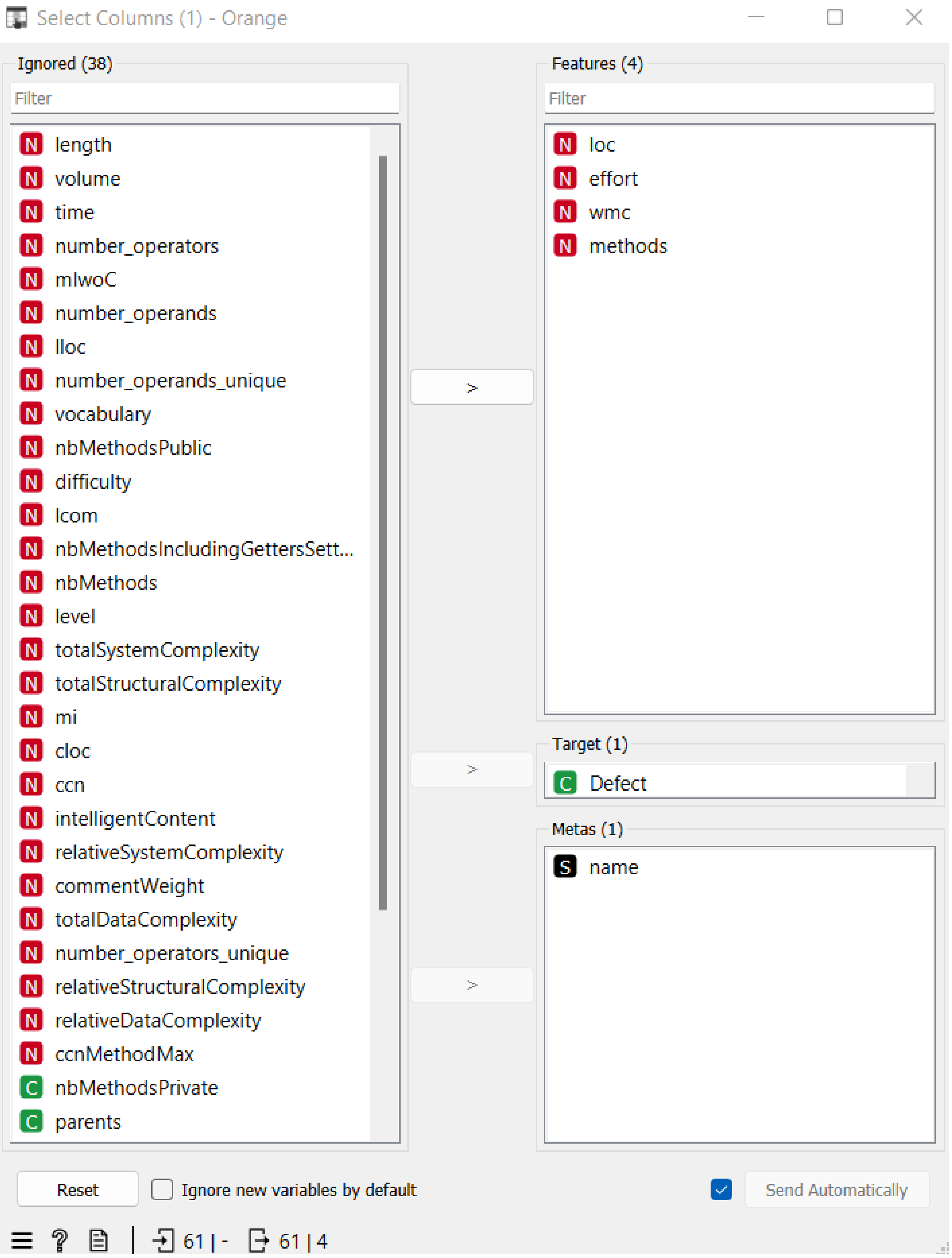


Figure 15. Select Column View on Bisnissyariah.co.id Dataset

In Figure 15 above is the Select Column display which functions to select which features have good accuracy from 1 to 38. The selected feature is only 4 because it is very relevant to the experiment.

6. Model *Machine Learning*

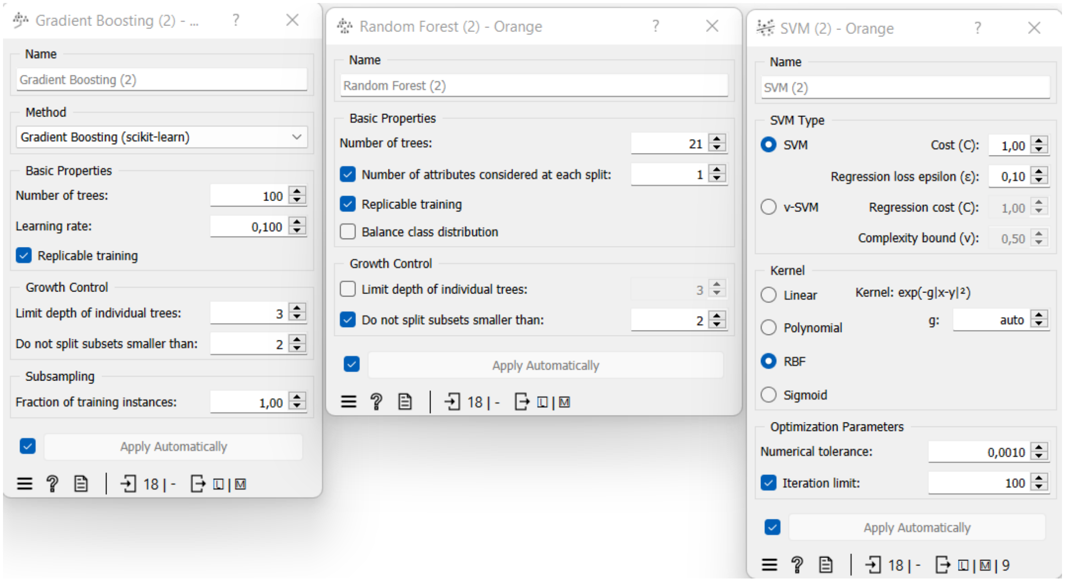
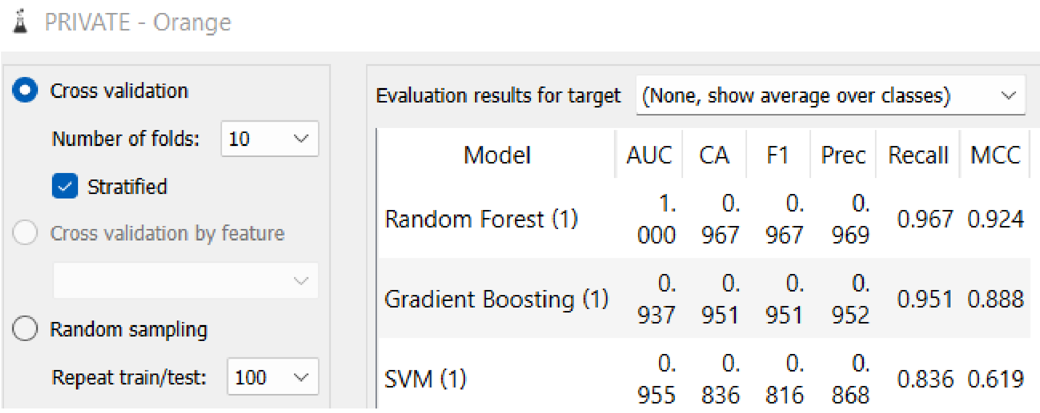


Figure 16. Machine Learning Model of Bisnissyariah.co.id Dataset

In Figure 16 above are various machine learning models. These machine learning models include Gradient Boosting, Random Forest, and Support Vector Machine (SVM). Because to do predictions with different models with default settings.

7. Click *Test and Score*



Model	AUC	CA	F1	Prec	Recall	MCC
Random Forest (1)	1.000	0.967	0.967	0.969	0.967	0.924
Gradient Boosting (1)	0.937	0.951	0.951	0.952	0.951	0.888
SVM (1)	0.955	0.836	0.816	0.868	0.836	0.619

Figure 17. Accuracy Results of Various Machine Learning Models for Bisnissyariah.co.id Dataset

In Figure 17 above are the accuracy results of various Machine Learning models. Using K-fold Cross Validation with a number of 10 folds is superior to the accuracy of Random Forest. Because the number of folds in K-Fold is 10, the predictions issued are more accurate in accordance with the data on Bisnissyariah.co.id

8. Click *Confusion Matrix*

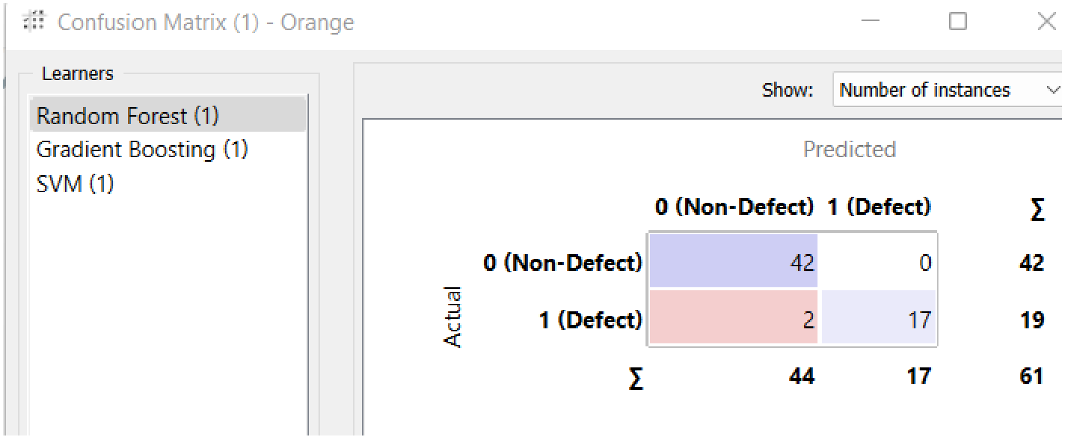


Figure 18. Confusion Matrix with Random Forest Dataset Bisnissyariah.co.id

In Figure 18 above is the number of confusion matrix using random forest.

True Negative (TN): 42, False Positive (FP): 0, False Negative (FN): 2, and True Positive (TP): 17. The next explanation in the actual section obtained for defect (1) is obtained with the number 19, and for no defect (0) 42. This means that in fact there are only 19 defects. On the other hand, according to the prediction using random forest for defect (1), the number 17 is obtained, and for no defect (0), the number 44 is obtained. This means that using random forest there are only 17 features with defects.

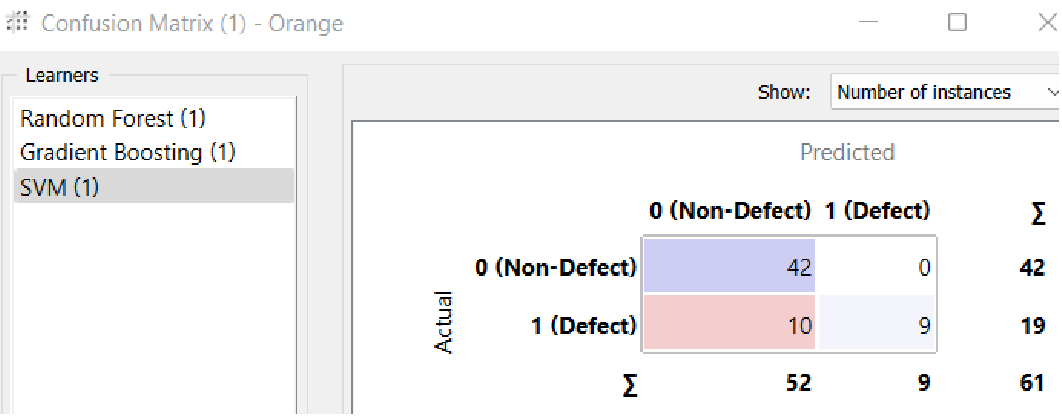


Figure 19. Confusion Matrix with Support Vector Machine (SVM) Dataset Bisnissyariah.co.id

In Figure 19 above is the number of confusion matrix using Support Vector Machine (SVM)

True Negative (TN): 42, False Positive (FP): 0, False Negative (FN): 10, and True

Positive (TP): 9. The next explanation in the actual section obtained for defect (1) is obtained with the number 19, and for no defect (0) 42. This means that in fact there are only 19 defects. On the other hand, according to predictions using Support Vector Machine for defect (1) is obtained with the number 9, and no defect (0) is obtained with the number 52. This means that using Support Vector Machine there are 9 defects

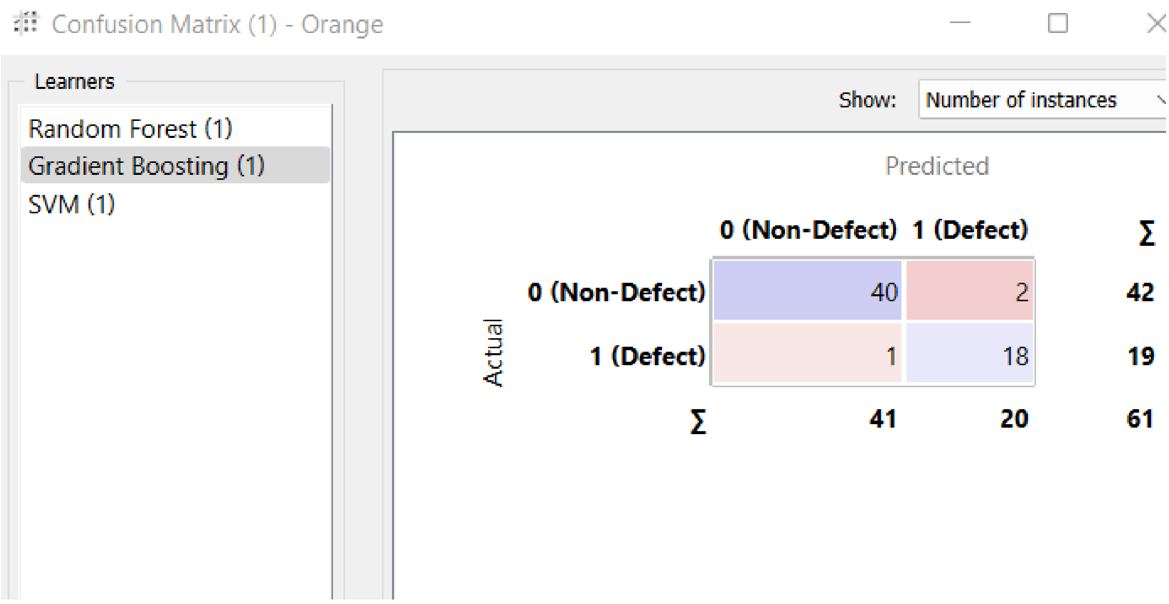


Figure 20. Confusion Matrix with Gradient Boosting Dataset Bisnissyariah.co.id

In Figure 20 above is the number of confusion matrix using Gradient Boosting.

True Negative (TN): 40, False Positive (FP): 2, False Negative (FN): 1, and True Positive (TP): 18. The next explanation in the actual section obtained for defect (1) is obtained with the number 19, and for no defect (0) 42. This means that in fact there are only 19 defects. On the other hand, according to the prediction using Gradient Boosting for defect (1) is obtained with the number 20, and no defect (0) is obtained with the number 41. This means that using Gradient Boosting there are 20 defect features.

Dataset Evaluation

In this section is to display the model accuracy comparison of each dataset used as follows:

Accuracy Evaluation of Test And Score Dataset K-Fold Cross Validation

There is a table for performing cross validation using K-Fold Cross Validation. Why the selection of data validation folds is different, because each data is different and also for data validation folds it depends on the value of the available data class, so to find out the better accuracy value, you must follow the number of folds recommended by Number Of Folds.

The results from the machine learning models using the Bisnissyariah dataset are as follows.

Table 3. Evaluation of K-Fold Cross Validation Dataset Accuracy

Dataset	Number Of Fold	Model	Accuracy	Precision	*Recall*	*F1-Score*
Bisnissyariah.co.id	10	**Random Forest**	**0.967**	**0.969**	**0.967**	**0.967**
		Gradient Boosting	0.951	0.952	0.951	0.951
		Support Vector Machine	0.836	0.868	0.836	0.816

Among the various machine learning models employed with the Bisnissyariah dataset, the Random Forest model achieved the highest accuracy score of 0.967, equivalent to 96.7%.

Conclusion

The testing and evaluating the dataset using machine learning techniques and cross-validation with the K-Fold method revealed remarkable performance. In this analysis, Random Forest demonstrated optimal results on the Bisnissyariah.co.id dataset, achieving an accuracy rate of 96.7%. The performance of various machine learning models tested and validated through the K-Fold cross-validation process showed that Random Forest emerged as the most superior model in this study, delivering the highest accuracy rate of 96.7% on the Bisnissyariah.co.id dataset. This research is expected to contribute significantly to stakeholders, particularly Bisnissyariah.co.id, a pioneering national news portal covering Islamic economy, Hajj and Umrah, Muslim tourism, and other topics. It also compares with datasets from writers' community websites and university websites, helping Bisnissyariah.co.id better understand user trends and preferences across various digital platforms. Future research could explore several recommendations to enhance this study's impact: (1) improving software on Bisnissyariah.co.id by using more advanced feature selection techniques beyond Information Gain Ratio, enabling the selection of more relevant features and increasing prediction accuracy for software defect detection; (2) applying alternative machine learning models to achieve the most accurate predictions for different types of datasets; (3) leveraging advanced machine learning models such as deep learning, which can handle data complexity and deeper correlations, thereby enhancing predictive capabilities for software defects.

Additionally, implementing solutions for software defect detection should involve collaboration between software development teams and end-users to determine the most relevant features and attributes. Direct input from experts in the field will ensure that the selected features are highly relevant and reflect characteristics influencing software performance. Such collaboration will improve quality and ensure the accuracy of prediction results.

Acknowledgements

We extend our gratitude to our families and everyone who contributed to this research, especially the respondents who generously shared their valuable insights. The support and assistance from various individuals and groups have been instrumental in the successful completion of this study.

References

Adetunji, A. B., Akande, O. N., Ajala, F. A., Oyewo, O., Akande, Y. F., & Oluwadara, G. (2022). House Price Prediction using Random Forest Machine Learning Technique. *Procedia Computer Science*, *199*, 806–813. https://doi.org/https://doi.org/10.1016/j.procs.2022.01.100

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., & Zimmermann, T. (2019). Software Engineering for Machine Learning: A Case Study. *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*, 291–300. https://doi.org/10.1109/ICSE-SEIP.2019.00042

Arnavaz, K., Feragen, A., Krause, O., & Loog, M. (2020). Bayesian active learning for maximal information gain on model parameters. *Proceedings - International Conference on Pattern Recognition*, 10524–10531. https://doi.org/10.1109/ICPR48806.2021.9411962

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215. https://doi.org/10.1016/j.neucom.2019.10.118

Davoudian, A., & Liu, M. (2020). Big Data Systems: A Software Engineering Perspective. *ACM Computing Surveys*, *53*(5). https://doi.org/10.1145/3408314

Dogan, A., & Birant, D. (2021). Machine learning and data mining in manufacturing. In *Expert Systems with Applications* (Vol. 166). Elsevier Ltd. https://doi.org/10.1016/j.eswa.2020.114060

Hamid, T. M. T. A., Sallehuddin, R., Yunos, Z. M., & Ali, A. (2021). Ensemble Based Filter Feature Selection with Harmonize Particle Swarm Optimization and Support Vector Machine for Optimal Cancer Classification. *Machine Learning with Applications*, *5*, 100054. https://doi.org/10.1016/j.mlwa.2021.100054

Kerbaa, T. H., Mezache, A., & Oudira, H. (2019). Model Selection of Sea Clutter Using Cross Validation Method. *Procedia Computer Science*, *158*, 394–400. https://doi.org/10.1016/j.procs.2019.09.067

Khammas, B. M. (2020). Ransomware Detection using Random Forest Technique. *ICT Express*, *6*(4), 325–331. https://doi.org/10.1016/j.icte.2020.11.001

Li, R., Zhou, L., Zhang, S., Liu, H., Huang, X., & Sun, Z. (2019). Software Defect Prediction Based on Ensemble Learning. *ACM International Conference Proceeding Series*, 1–6. https://doi.org/10.1145/3352411.3352412

Liu, Y., Liao, S., Jiang, S., Ding, L., Lin, H., & Wang, W. (2020). Fast Cross-Validation for Kernel-Based Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(5), 1083–1096. https://doi.org/10.1109/TPAMI.2019.2892371

Maleki, N., Zeinali, Y., & Niaki, S. T. A. (2021). A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection. *Expert Systems with Applications*, *164*. https://doi.org/10.1016/j.eswa.2020.113981

Malhotra, R., & Khan, K. (2020). A Study on Software Defect Prediction using Feature Extraction Techniques. *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 1139–1144. https://doi.org/10.1109/ICRITO48877.2020.9197999

Nguyen, B. H., Xue, B., & Zhang, M. (2020). A survey on swarm intelligence approaches to feature selection in data mining. *Swarm and Evolutionary Computation*, *54*. https://doi.org/10.1016/j.swevo.2020.100663

Nti, I. K., Nyarko-Boateng, O., & Aning, J. (2021). Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation. *International Journal of Information Technology and Computer Science*, *13*(6), 61–71. https://doi.org/10.5815/ijitcs.2021.06.05

Odhiambo Omuya, E., Onyango Okeyo, G., & Waema Kimwele, M. (2021). Feature Selection for Classification using Principal Component Analysis and Information Gain. *Expert Systems with Applications*, *174*. https://doi.org/10.1016/j.eswa.2021.114765

Prabha, C. L., & Shivakumar, N. (2020). Software Defect Prediction Using Machine Learning Techniques. *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 728–733. https://doi.org/10.1109/ICOEI48184.2020.9142909

Rostami, M., Berahmand, K., Nasiri, E., & Forouzande, S. (2021). Review of swarm intelligence-based feature selection methods. In *Engineering Applications of Artificial Intelligence* (Vol. 100). Elsevier Ltd. https://doi.org/10.1016/j.engappai.2021.104210

Serban, A., Van Der Blom, K., Hoos, H., & Visser, J. (2020, October). Adoption and effects of software engineering best practices in machine learning. *International Symposium on Empirical Software Engineering and Measurement*. https://doi.org/10.1145/3382494.3410681

Speiser, J. L., Miller, M. E., Tooze, J., & Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. In *Expert Systems with Applications* (Vol. 134, pp. 93–101). Elsevier Ltd. https://doi.org/10.1016/j.eswa.2019.05.028

Tao, X., Li, Q., Ren, C., Guo, W., He, Q., Liu, R., & Zou, J. (2020). Affinity and class probability-based fuzzy support vector machine for imbalanced data sets. *Neural Networks*, *122*, 289–307. https://doi.org/10.1016/j.neunet.2019.10.016

Thanoon, M. A., Zedan, M. J. M., & Hameed, A. N. (2019). Feature Selection Based on Wrapper and Information Gain. *NICST 2019 - 1st Al-Noor International Conference for Science and Technology*, 32–37. https://doi.org/10.1109/NICST49484.2019.9043805

Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T., & Rellermeyer, J. S. (2020). A Survey on Distributed Machine Learning. In *ACM Computing Surveys* (Vol. 53, Issue 2). Association for Computing Machinery. https://doi.org/10.1145/3377454

Wibisono, W., & Baskoro, F. (2023). Pengujian Blackbox Metode Equivalent

Partitions pada Aplikasi Karyawan Website Oby Komputer. *Jurnal Sistem Informasi (Teknofile)*, *1*(2). https://jurnal.nawansa.com/index.php/teknofile/article/view/16

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061