Available at https://journalenrichment.com/index.php/jr/

Enrichment: Journal of Multidisciplinary Research and Development

Implementation of Random Forest Using Smote and Smoteenn in Customer Churn Classification in E-Commerce

Muhammad Munzir Rizkya Mubarak[1*], Yulison Herry Chrisnanto[2], Puspita Nurul Sabrina[3]

[1]Department Informatics Engineering/Faculty of Science and Informatics/Universitas Jenderal Achmad Yani, Cimahi, West Java, Indonesia
[2] Department Informatics Engineering/Faculty of Science and Informatics/Universitas Jenderal Achmad Yani, Cimahi, West Java, Indonesia
[3] Department Informatics Engineering/Faculty of Science and Informatics/Universitas Jenderal Achmad Yani, Cimahi, West Java, Indonesia

[*]Email: email@email.com

ARTICLE INFO

Keywords:
customer churn
random forest
imbalance data
smote
smoteenn

ABSTRACT

The rapid development of the internet is one of the driving factors behind the growth of e-commerce. This has led to the emergence of many e-commerce companies, resulting in intense competition among them. Customers have the right to choose the e-commerce platforms that suit their needs and can switch to competing e-commerce platforms, a phenomenon known as customer churn. This issue can be addressed by classifying customer behavior based on existing data. This study utilizes the Random Forest Classifier method, employing the SMOTE and SMOTEENN resampling techniques to handle data imbalance. From the conducted research, the best results were achieved using the SMOTE implementation, with an accuracy of 96.3%, precision of 87.8%, recall of 87.1%, f1-score of 87.4%, and an AUC score of 93%. These results successfully strike a balance between recognizing the positive class (churn) and controlling false positives. On the other hand, the SMOTEENN implementation yields the best recall value and an increase in AUC score, but it comes with a significant decrease in precision, indicating a challenge in controlling false positives.

INTRODUCTION

One of the challenges faced by the e-commerce industry is efforts to reduce the number of customers who stop using a company's services and switch to competitors, which is often referred to as churn. The churn rate describes unsustainable customer behavior or the comparison between the number of customers who stop using a company's services in a certain period with the total number of customers in the same period. A high churn rate can hurt a company's profitability (Ashish & Kumar, 2017).

Data mining is an analytical tool in the CRM methodology that has been proven useful for an industry in calculating customer churn, namely classifying customer behavior based on previously existing data, by taking into account the number of customers who switch to competing companies, companies can create advertising that is intended to be useful. increase customer loyalty improve marketing strategies to acquire current customers and implement customer retention (Dyche, 2002).

The data collected for churn prediction is usually imbalanced, where cases of non-churn customers far outnumber cases of churned customers (Rodan et al., 2015). The classification algorithm does not consider the existence of data imbalances in the dataset which can affect prediction performance and can increase bias in the majority class. There are two methods for handling unbalanced data, namely using oversampling and undersampling. Oversampling operates by balancing the minority class by duplicating the class's entities, while undersampling works by equalizing the minority class by eliminating majority class entities until the distribution becomes balanced (Wijayanti et al., 2021).

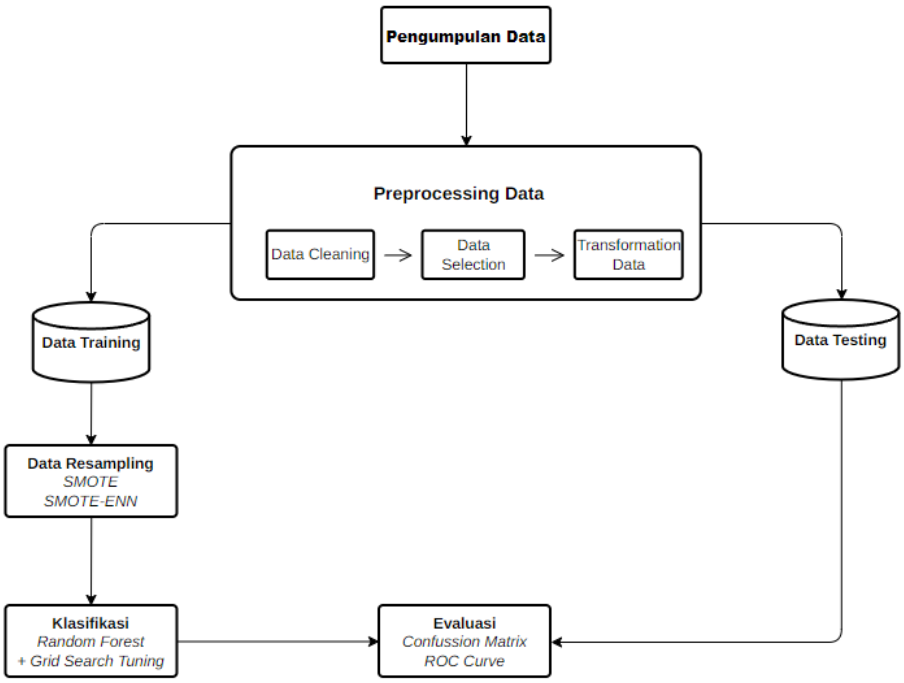
Previous research that studied customer churn was research to classify customer churn in the telecommunications industry using random forest, logistic regression, and XGBoost. This research obtained results that the random forest algorithm had slightly greater accuracy than the other two algorithms (Kavitha et al., 2020).

The next research is about the resampling approach for class imbalance problems in churn prediction. This research uses several resampling techniques to get improvements in model performance (Salunkhe & Mali, 2018). Then in other research regarding data resampling approaches to handle data imbalance problems using the SMOTE, ADASYN, RUS, and SMOTEENN resampling techniques (Sir & Soepranoto, 2022). From this research, SMOTEENN resampling had the best results. Furthermore, the research "Credit assessment classification using random forest by applying the SMOTE and Random Over-Under Sampling methods" can improve the performance of Random Forest (Syukron & Subekti, 2018).

Therefore, researchers are interested in applying the Random Forest Classifier method with SMOTE and SMOTENN resampling techniques to classify customer churn data that has unbalanced data (Indrawati, Subagyo, & Sihombing, 2020). This research aims to provide better insight into the performance of the Random Forest Classifier method in the context of informatics.

METHOD

The research methodology is in the form of a flow of stages which is a reference for classifying customer churn. The plot design in this research is shown in Figure 1:



In this research, the first step taken was a literature study to collect various information and as a reference related to the topic being searched for, namely data mining, random forests, and how to overcome data imbalances. At this stage, observing the collection of customer churn data in e-commerce. At the data preprocessing stage, data cleaning, data selection, and data transformation are carried out. Cleaning steps in the initial data to ensure there is no incomplete or null data as well as redundant data and preparing the data so that the data is ready to be processed according to needs (Suripto, Rahmanita, & Kirana, 2022). The data resampling process is a way to overcome data imbalance. When dealing with unbalanced data, this means that there are significant differences in the number of observations between different classes or categories in the target variable. The resampling techniques that will be used in this research are SMOTE and SMOTEENN (Wulandari, 2018).

At this stage, the classification process combines decisions from many randomly generated decision trees to achieve more stable and accurate predictions. Random Forest also can handle data with many features and can identify important features in the dataset. Model tuning is a step to provide parameters to the model that have the highest accuracy. The evaluation results of the tuning model were carried out with Grid search CV using the Python

library to find the best parameter combination. Once predictions are made, the final step is to evaluate the model's performance (Turban, Kelly Rainer, & Richard, 2005). Evaluation metrics commonly used in classification are accuracy, precision, recall, F1-score, and confusion matrix. This metric provides an idea of the extent to which the model can correctly predict the target class

RESULTS AND DISCUSSION

Preparation for Implementation

In this research, the software system used is web-based, the system design was created with a MySQL database connection using the Python and HTML programming languages with the Flask framework and the XAMPP control panel tool, the system theme was created with Visual Paradigm, and the media that runs the system is the Google Chrome browser. Interface design using the Figma application (Larose & T., 2006).

System Implementation

System implementation is the implementation stage of the system designed in Chapter 3, which is carried out using the Random Forest Classifier to create a customer churn prediction system in e-commerce.

Database Implementation

The database implementation is built based on the database design that has been designed in CHAPTER III analysis and design of customer churn prediction systems in e-commerce using the Random Forest Classifier method

1. Database Users

In the image below is the structure of the user's table. There are 3 columns created in this table that are used in the login process, namely ID, username, and password.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Figure 1 Database Users

2. Predictions Database

In the image below is the structure of the predictions table. There are 5 columns created in this table which are used to store prediction results, namely id, timestamp, churn, input_values, and username.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	timestamp	datetime			No	None			Change Drop More
3	churn	int(11)			No	None			Change Drop More
4	input_values	longtext	utf8mb4_bin		No	None			Change Drop More
5	username	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

Figure 2 Prediction Database

User Interface Implementation

Interface implementation is the implementation of the system design results that were defined in the previous chapter. The purpose of implementing a user interface is to make the system created available to system users. This allows users to interact with the system through the interface. The interface built in this research was built using HTML and Bootstrap. Below is the interface implementation.

1. Authentication View
   a. Login View

      This Login page consists of a form for users to log in to the system. Users must fill in their email and password. The implementation of the login page can be seen in the image below.

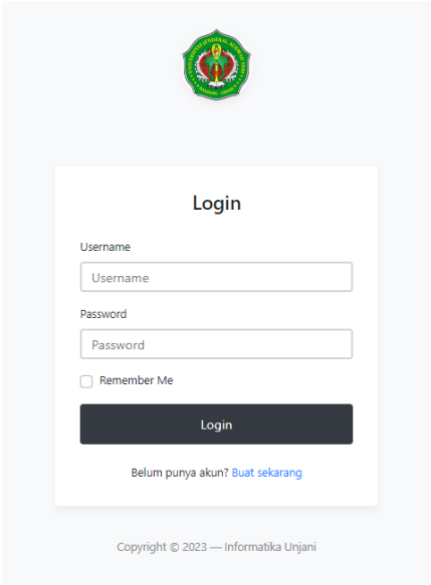


Figure 3 Login Page Implementation

b. Register Display

This Register page consists of a form for users to register an account on the system. Users must fill in their username and password. The implementation of the register page can be seen in the image below.

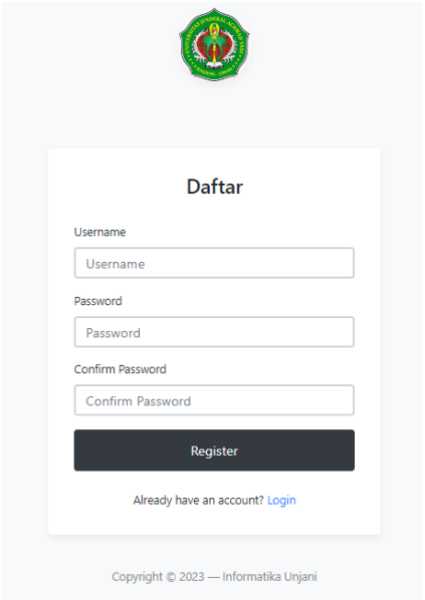


Figure 4 Implementation of the Register Page

2. Dashboard display

The dashboard interface is the first display when opening the software. This interface implementation has a system name page at the top and a system menu vertically. The implementation of the dashboard page can be seen in the image below

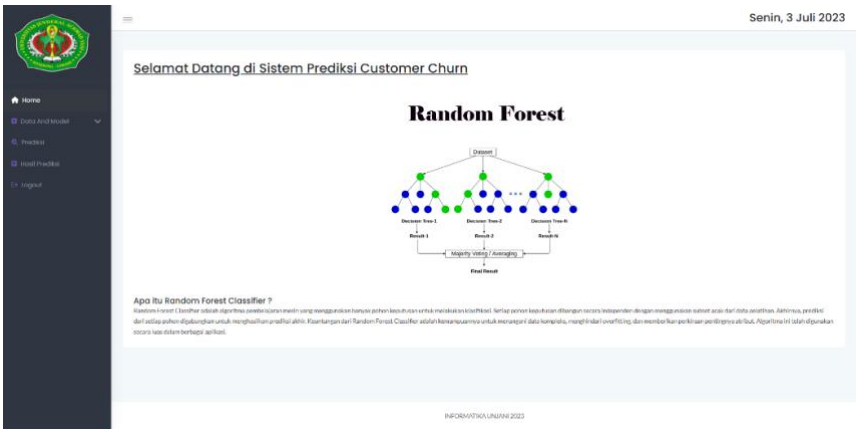


Figure 5 Dashboard Page Implementation

3. Data Preprocessing Interface

The data preprocessing interface contains a display for importing datasets and will also display the results of the data before preprocessing and after preprocessing.

a. Import Dataset

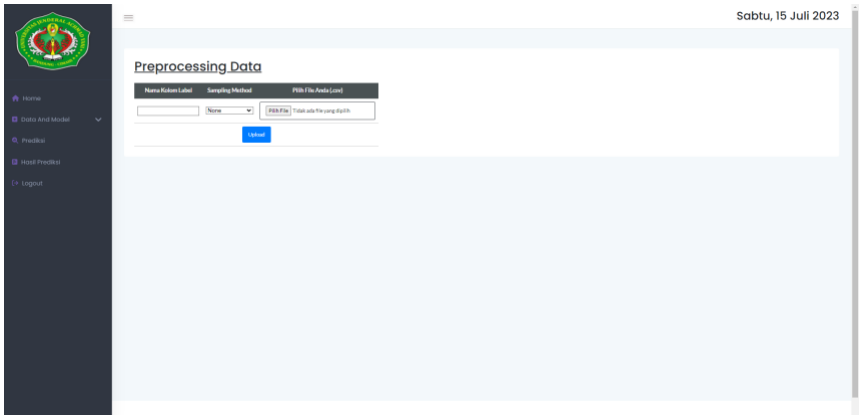


Figure 6 Implementation of the Dataset Import Page

4. Data and Model Interface

The Data and Model interface contains the import dataset display (if the user has not carried out preprocessing, otherwise if the user has carried out preprocessing the upload dataset option will not appear), then it will display a preview of the dataset, and also a model evaluation will be displayed if the user selects the process button (Presman, 2012).

a. Import Data

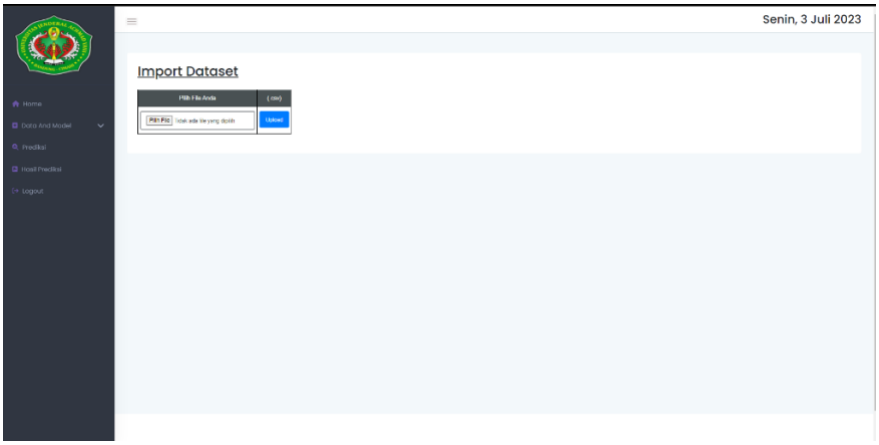


Figure 4 7 Data Import Implementation

b. Classification Process

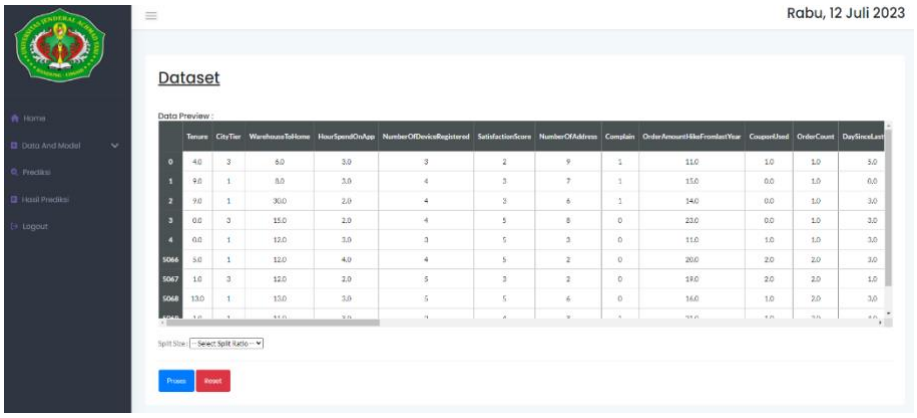


Figure 8 Implementation of the Classification Process

c. Model Evaluation

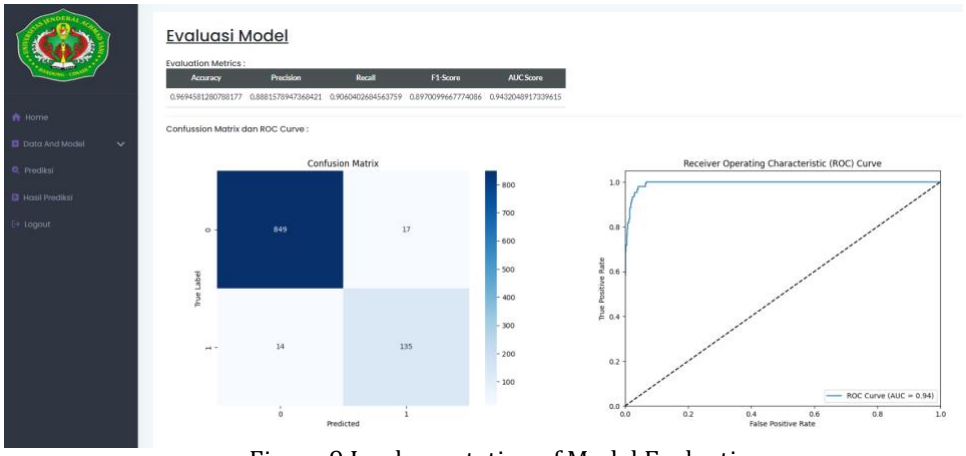


Figure 9 Implementation of Model Evaluation

5. Prediction Interface

The prediction interface is an additional feature that was created, this page will display a new data upload form and then will display the churn (1) or non-churn (0) prediction results.

a. Make a Prediction

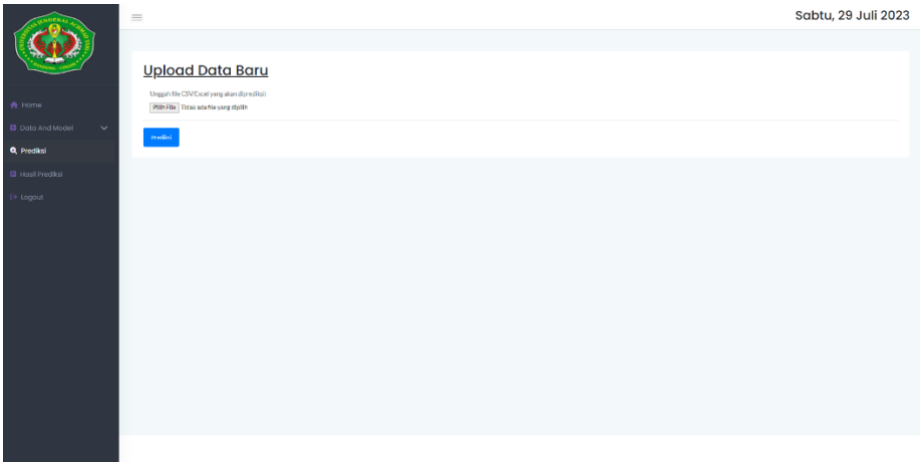


Figure 10 Implementation of Making Predictions

b. Prediction Results

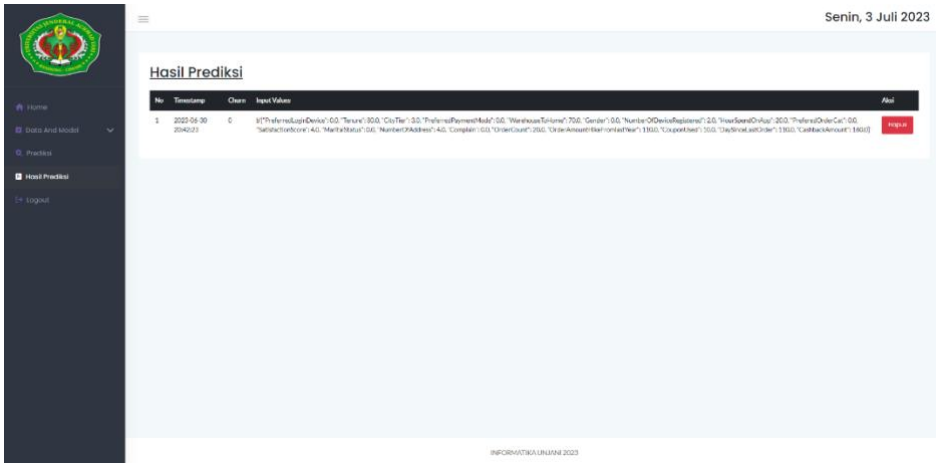


Figure 11 Implementation of Prediction Results

Model Testing Experiments

At this stage, testing will be carried out on the imbalanced data, with SMOTE resampling and with SMOTEEN resampling (Maria Dimova & Stirk, 2019). The parameters that will be used in Random Forest based on searches using Grid Search are as follows in Table 1.

Table 1. Tuning Parameters

Data	Parameter	Value
Imbalanced	n_estimators	100
	max_depth	20
	criterion	Gini
SMOTE	n_estimators	300
	max_depth	25
	criterion	entropy
SMOTEENN	n_estimators	400
	max_depth	25
	criterion	entropy

Testing on Imbalanced Data

The parameters used in the imbalance data according to the Grid Search parameter search are (n_estimators = 100, max_depth = 20, criterion = Gini). Testing is carried out by splitting test data by 30% and training data by 70% and will be evaluated using a confusion matrix (Karn et al., 2022).

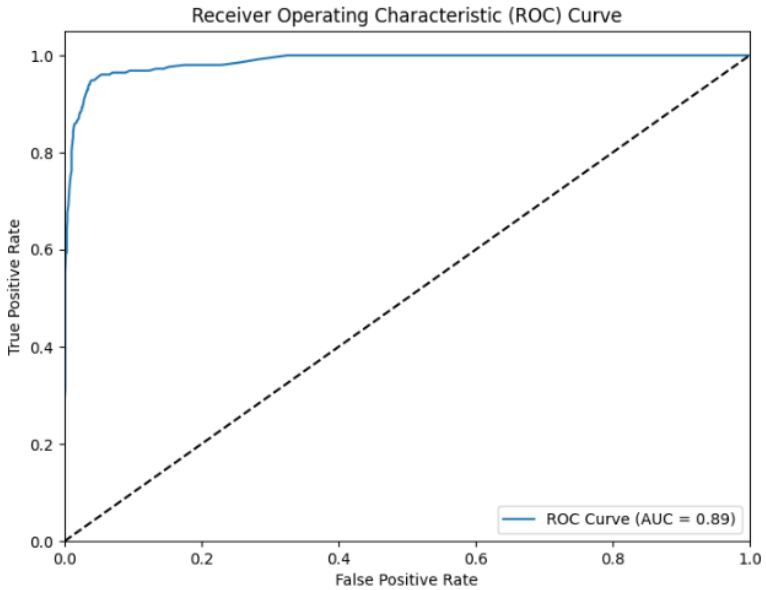


Figure 12 ROC Curve Imbalanced Data

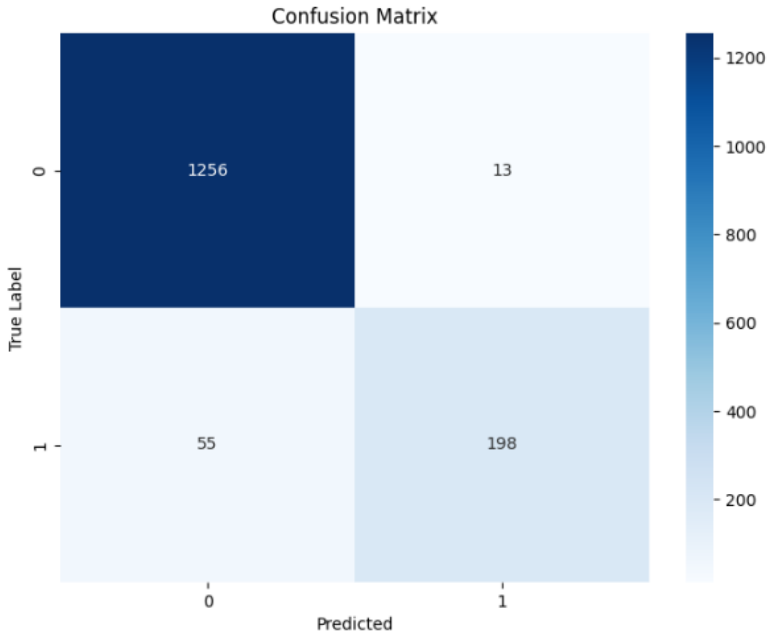


Figure 13 Confusion Matrix Imbalanced Data

The following is the confusion matrix calculation by calculating Accuracy, Precision, Recall, and F1-Score as follows:

a. Precision
   True Positive / (True Positive + False Positive)
   =198/(198+13)
   = 93.8%
b. Recall
   True Positive / (TruePositive + False Negative)
   = 198(198+55)
   = 78.2%
c. F1-Score
   2 * (Precission*Recall) / (Precission+Recall)
   = 2 *(93.8*78.2) / (93.8+78.2)
   = 2*(7,335) / (172)
   = 14.67 / 172
   = 0.0852X 100%
   = 85.2%
d. Accuracy
   (TP+TN) / (TP+TN+FP+FN)
   = (198 + 1256) / (198+ 1256 + 13 + 55)
   = 1454 / 1522
   = 0.955x 100%
   = 95.5%

Testing on SMOTE

The parameters used in smote data according to the Grid Search parameter search are (n_estimators = 400, max_depth = 25, criterion = entropy). Testing is carried out using 30% test data and 70% training data and will be evaluated using a confusion matrix.

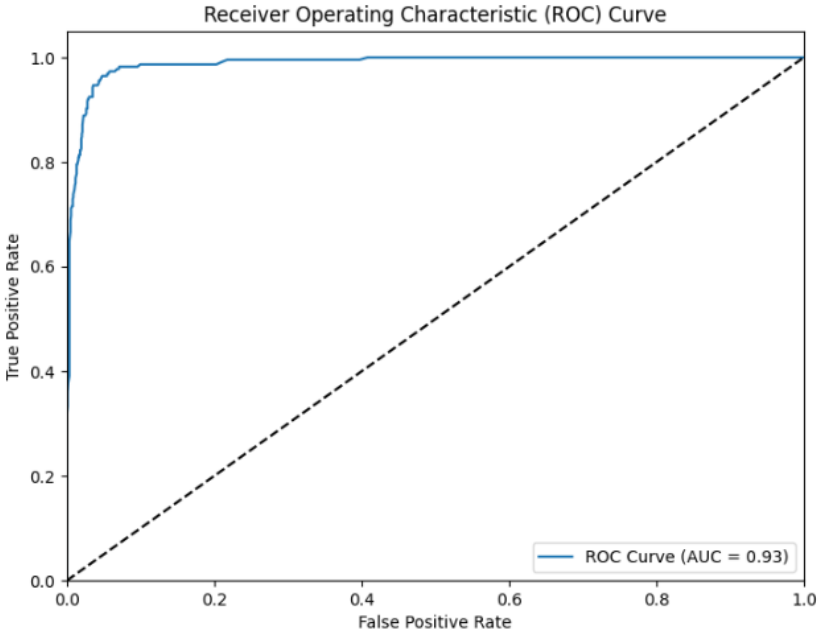


Figure 14 ROC Curve SMOTE

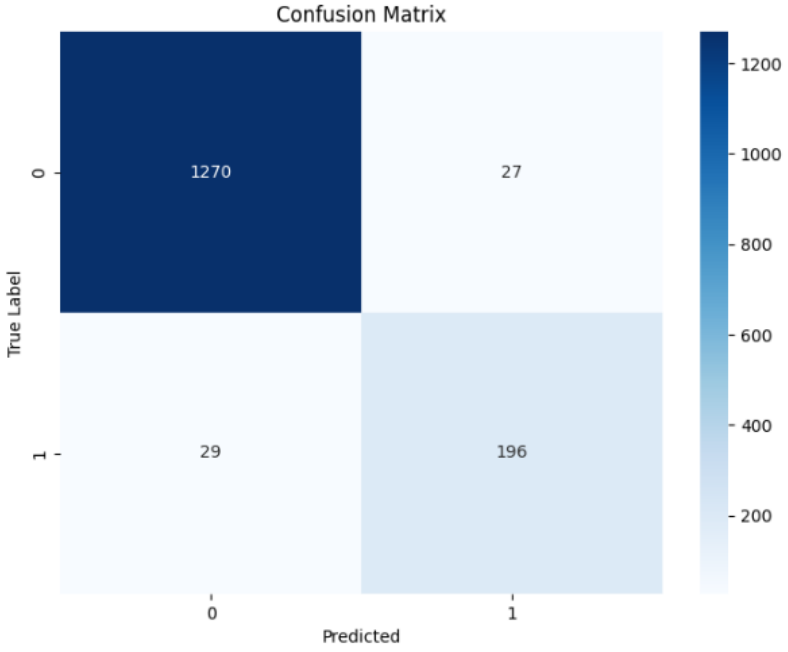


Figure 15 Confusion Matrix SMOTE

The following is the confusion matrix calculation by calculating Accuracy, Precision, Recall, and F1-Score as follows:

a. Precision
   True Positive / (True Positive + False Positive)
   = 196/(196+27)
   = 87.8%
b. Recall
   True Positive / (TruePositive + False Negative)
   = 196/(196+29)
   = 87.1%
c. F1-Score
   2 * (Precission*Recall) / (Precission+Recall)
   = 2 *(87.8*87.1) / (87.8+87.1)

= 2*(7.647) / (174.9)
= 15.294 / (174.9)
= 0.0874 X 100%
= 87.4%

d. Accuracy
(TP+TN) / (TP+TN+FP+FN)
= (196 + 1270) / (196 + 1270 + 27 + 29)
= 1466 / 1522
= 0.963 x 100%
= 96.3%

Testing on SMOTENN

The parameters used in SMOTEENN data according to the Grid Search parameter search are (n_estimators = 400, max_depth = 25, criterion = entropy). Testing is carried out using 30% test data and 70% training data and will be evaluated using a confusion matrix.

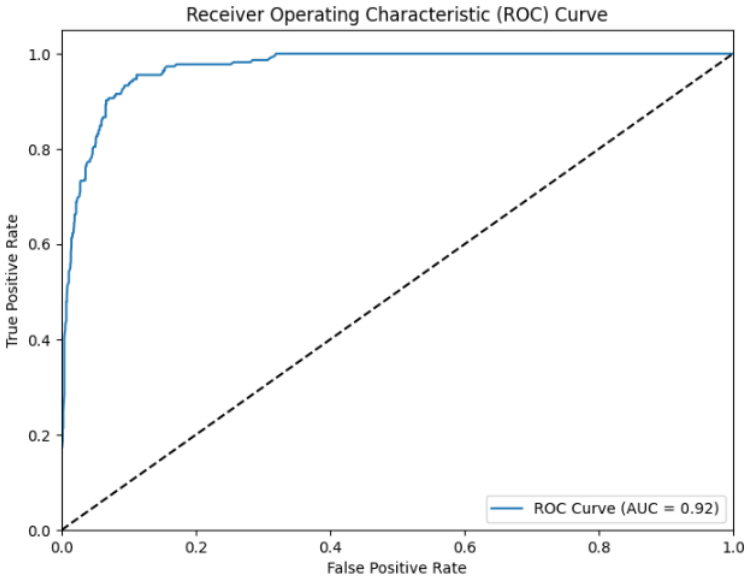


Figure 16 ROC Curve SMOTEEN

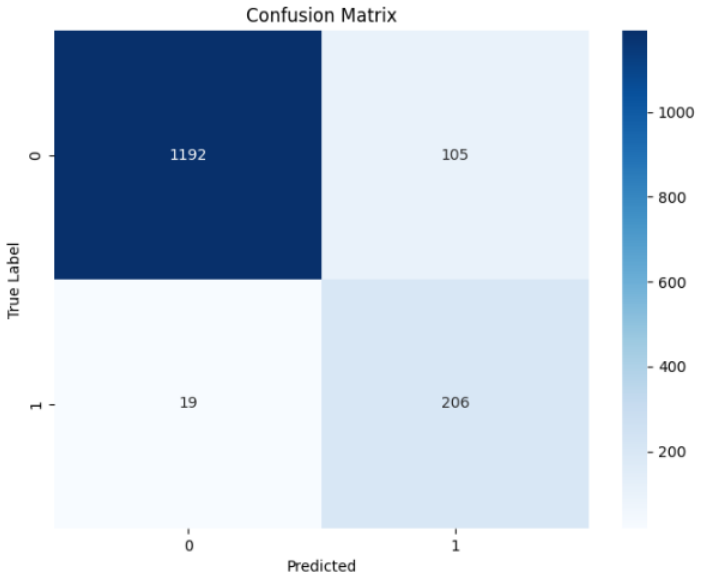


Figure 17 SMOTEENN Confusion Matrix

The following is the confusion matrix calculation by calculating Accuracy, Precision, Recall, and F1-Score as follows:

a. Precision
   True Positive / (True Positive + False Positive)
   =206/(206+105)
   = 66.2%
b. Recall
   True Positive / (TruePositive + False Negative)
   = 206(206+19)
   = 91.5%
c. F1-Score
   2 * (Precission*Recall) / (Precission+Recall)
   = 2 *(66.2*91.5) / (66.2+91.5)
   = 2*(6.057) / (157.7)
   = 12.114 / (157.7)
   = 0.0768 X 100%
   = 76.8%
d. Accuracy
   (TP+TN) / (TP+TN+FP+FN)
   = (206 + 1192) / (206 + 1192 + 105 + 19)
   = 1398 / 1522
   = 0.918 x 100%
   = 91.8%

Black Box Testing

Black Box testing is a test carried out to observe execution results through test data and check the functionality of the software that has been built (Syarif et al., 2016).

Testing Stages

The steps taken when testing this software include the activities described below:

1. Determine the objectives of quality testing
2. Determine the category of quality test results
3. Design quality testing based on use case grouping
4. Implementation of quality testing
5. Conclusion and quality testing results.

Testing Objectives

This stage discusses the objectives of quality testing of the software being built (Breiman & Cutler, 2003). The purpose of the test is described in this table

No	Use Case	Purpose
1.	Register	Perform testing on actors, to register in the system.
2.	Login	Testing the actor, to carry out the login process on the system.
3.	Import Data	Carry out tests on actors, to carry out the process of importing datasets into the system.
4.	View Data	Carry out testing on actors, to display the pre-processing process in the system
5.	Classification	Testing actors, to carry out classification using the Random Forest Classifier method
6.	Evaluation	Conduct testing on actors, to see the evaluation of the random forest classification process
7.	Make Prediction	Predictions Test the actor, to make predictions based on input from the actor
8.	Results Prediction	Results Test the actors, to see the results of the predictions that have been made.

Test Success Category

The category for determining quality testing on software is divided into two categories, namely:

1. Appropriate, if the quality and function of the software being tested are by the planning objectives and its use, then it is included in the appropriate category.
2. Not Appropriate, if the quality and function of the software being tested is not by the planning objectives and its use, then it is included in the Not Appropriate category.

Testing Scenarios

The purpose of designing quality testing of the software that has been built is to serve as a reference in carrying out quality testing of the software that has been built. The test scenario is shown in the table below

Table 3 Test Scenarios

Use Case	Feature Name	Test Code	Test Case
Authentication	Register	UC1.1	This feature functions to register into the system.
	Login	UC1.2	This feature functions to validate user identity and provide access.
Preprocessing	Overcoming Missing Values	UC2.1	This feature functions to resolve datasets if there is missing data.
	One Hot Encoding	UC2.2	This feature functions to convert categorical data into numeric
	Data Sampling	UC2.3	This feature functions to overcome data imbalance
Classification	Import Data	UC3.1	Import Classification This feature functions if the user wants to carry out the classification process directly without preprocessing the data.
	Classification Process	UC3.2	Classification Process This feature can carry out a classification process with the option of splitting training and test data according to the selected ratio
	Model Evaluation	UC3.3	This feature can display random forest model evaluation
Prediction	Make Prediction	UC4.1	This feature functions to predict labels based on feature values input by the user
	Results Prediction	UC4.2	This feature functions to predict labels based on feature values input by the user

Test Implementation

At this stage, testing the quality of the customer churn prediction system software for e-commerce has been built. Testing is carried out by looking at the reference design that has been made, and then adjusting the test results to the objectives to be achieved from the design that has been made. The test implementation is shown in the table below.

Table 4 Test Implementation

No	Test Code	System Response	Expected Results	Results
1	UC1.1 (Register)	Register then user data is stored in the database	The system stores the user's username and password	**In accordance**
	UC1.2 (Login)	Login to the system then the database validates. The system stores the user's username and password	The system can validate users registered in the database	**In accordance**

No	Test Code	System Response	Expected Results	Results
2	UC2.1 (Handling Missing Value)	Eliminate missing data in the dataset.	The system eliminates missing data in the dataset.	**In accordance**
	UC2.2 (One Hot Encoding)	Displays the encoded data	The system displays the encoded data.	**In accordance**
	UC2.3 (Oversampling Data)	Displays the amount of data sampled	The system displays the amount of data that has been oversampled	**In accordance**
3	UC3.1 (Import Data)	Import the dataset in .csv format and then save it in the static/upload folder.	The system saves the dataset in the static/upload folder	**In accordance**
	UC3.2 (Classification Process)	Carry out the classification process using the random forest method	The system processes classification according to the split data ratio.	**In accordance**
	UC3.3 (Model evaluation)	Displays evaluation results from creating a random forest model	The system displays a confusion matrix and roc curve evaluation	**In accordance**
4	UC4.1 (Prediction)	Displays the attribute value input form	The system displays the attribute value input form	**In accordance**
	UC4.2 (Results Prediction)	Displays prediction results	The system displays the prediction results	**In accordance**

Model and Black Box Testing Conclusions

After testing the model from several scenarios, it can be concluded that the results obtained for measuring model performance using the evaluation method using the confusion matrix and AUC score are shown in Table 4.5 and Table 4.6.

Table 5 Experiment Code

Coding	Modeling
A	RF + Imbalanced Data
B	RF + SMOTE
C	RF + SMOTEENN

Experiment Recall Precision F1-Score Accuracy AUC

Table 6 Experiment Results

Experiment	Recall	Precision	F1-Score	Accuracy	AUC
A	78,2%	93,8%	85,2%	95,5%	89%
B	87,1%	87,8%	87,4%	96,3%	93%
C	91,5%	66,2%	76,8%	91,8%	92%

The results of the experiments carried out can be concluded as follows:

1. Experiment A (RF + Imbalanced Data): In this experiment, the Random Forest (RF) model is applied to imbalanced data. The results show high precision (93.8%), which indicates the model's ability to accurately recognize the positive class. However, the lower recall (78.2%) reveals that the model tends to miss some true positive classes. F1-Score (85.2%) reflects the balance between precision and recall. Accuracy (95.5%) is relatively high, but note that accuracy can be biased in data imbalance. The AUC Score value (89%) indicates that the model has a good ability to differentiate between positive and negative classes.

2. Experiment B (RF + SMOTE): In this experiment, the RF model is applied after oversampling using the SMOTE (Synthetic Minority Over-sampling Technique) method to handle data imbalance. The results show higher recall (87.1%), indicating the model's ability to identify more positive classes. Although precision decreased slightly (87.8%), the F1-Score (87.4%) remained high, indicating a good balance between precision and recall. Accuracy (96.3%) remained high, and AUC Score (93%) increased, indicating improvements in the model's ability to differentiate between classes.
3. Experiment C (RF + SMOTEENN): In this experiment, the RF model is applied after carrying out a combination of oversampling and undersampling using the SMOTEENN method. The results show the highest recall (91.5%), indicating good ability in recognizing positive classes. However, the low precision (66.2%) indicates an increase in false positives. F1-Score (77.8%) reflects a compromise between high recall and lower precision. Although the accuracy (91.8%) decreased, the value of accuracy alone cannot be measured in the data imbalance problem. The AUC Score value (92%) reflects the overall quality of the model in dealing with data imbalance.

From the explanation above, Experiment B using the SMOTE technique appears to be a better approach to overcome data imbalance in this case, because it achieves a good balance between precision and recall, and has better performance in distinguishing between classes. Furthermore, the conclusion from the black box testing that has been carried out is in a table with a total of 4 functions, namely authentication, preprocessing, classification, and prediction. So the percentage of suitability of functions in the system can be calculated as follows:

Number of Test Codes = 10 Test Codes
Test Codes with Corresponding Results = 10 Test Codes
Test Codes with Inappropriate Results = 0 Test Codes
Percentage

$$Percentage = \frac{(number\ of\ test\ codes - non - matching\ test\ codes)}{(number\ of\ test\ codes)} \times 100\%$$

The results of the calculation of the system suitability function can be concluded that testing of the customer churn prediction software using black box testing has run according to the specifications that have been set with a percentage of 100%

CONCLUSION

In this research, the Random Forest method was applied with SMOTE and SMOTENN resampling. From the results of the research carried out, Random Forest classification of imbalance data obtained quite good results with an accuracy of 95.5%, recall of 78.2%, precision of 93.8%, f1-score of 85.2%, and an AUC score of 89%. After resampling the data with SMOTE and SMOTEENN, this affects the performance in recognizing positive classes (churn) which can be seen in the recall value which increases by 8.9% in SMOTE resampling and 13.3% in SMOTEENN resampling but there is a decrease in the precision value in SMOTE resampling of 6% and SMOTEENN resampling 27.6%. However, the AUC value for the model with SMOTE is 4% higher and the model with SMOTENN is 3% higher than the AUC value produced by the model without resampling. This shows an improvement in the model's ability to differentiate between classes. When considering overall performance, the SMOTE resampling technique is better at dealing with imbalances compared to the SMOTEEN resampling technique because it can achieve a good balance between recognizing positive classes and controlling false positives.

REFERENCES

Ashish, S., & Kumar, V. (2017). *Analyzing Client Profitability across Diffusion Segments for a Continuous Innovation*. *54*(6).

Breiman, L., & Cutler, A. (2003). *Manual--Setting Up, Using, and Understanding Random Forest V4.0*.

Dyche, J. (2002). *The CRM Handbook*. Addison-Wesley Professional.

Indrawati, A., Subagyo, H., & Sihombing, A. (2020). *ANALYZING THE IMPACT OF RESAMPLING METHOD FOR IMBALANCED DATA TEXT IN INDONESIAN*. *9008*(21), 133–141.

Karn, A., Romero, C., Sengan, S., Mehbodniya, A., Webber, J., Pustokhin, D., & Wende, F.-D. (2022). Fuzzy & SVM Based Classification Model to Classify Spectral Objects in Sloan Digital Sky. *IEEE Access*, *PP*, 1. https://doi.org/10.1109/ACCESS.2022.3207480

Kavitha, V., Kumar, G., Kumar, S., & Harish, M. (2020). Churn Prediction of Customer in Telecom Industry using Machine Learning Algorithms. *International Journal of Engineering Research And*, *V9*. https://doi.org/10.17577/IJERTV9IS050022

Larose, & T., D. (2006). *Discovering Knowledge in Data: An Introduction to Data Mining*. John Willey & Sons. Inc.

Maria Dimova, C., & Stirk, P. M. R. (2019). 済無No Title No Title No Title. *Analysis and Prediction of Insurance Company's Customer Loss Based on BP Neural Network. Lanzhou University*, 9–25.

Presman. (2012). Jurnal Teknik Informatika Atmaluhur. *Jurnal Teknik Informatika Atmaluhur*, *6*(1), 40.

Rodan, A., Fayyoumi, A., Faris, H., Alsakran, J., & Al-Kadi, O. (2015). Negative correlation learning for customer churn prediction: A comparison study. *Scientific World Journal*, *2015*. https://doi.org/10.1155/2015/473283

Salunkhe, U. R., & Mali, S. N. (2018). A hybrid approach for class imbalance problem in customer churn prediction: A novel extension to under-sampling. *International Journal of Intelligent Systems and Applications*, *10*(5), 71–81. https://doi.org/10.5815/ijisa.2018.05.08

Sir, Y. A., & Soepranoto, A. H. H. (2022). Pendekatan Resampling Data Untuk Menangani Masalah Ketidakseimbangan Kelas. *Jurnal Komputer Dan Informatika*, *10*(1), 31–38. https://doi.org/10.35508/jicon.v10i1.6554

Suripto, Rahmanita, R. N., & Kirana, A. S. (2022). *Teknik pre-processing dan classification dalam data science*.

Syarif, I., Prugel-Bennett, A., & Wills, G. (2016). SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *14*(4), 1502. https://doi.org/10.12928/telkomnika.v14i4.3956

Syukron, A., & Subekti, A. (2018). Penerapan Metode Random Over-Under Sampling dan Random Forest Untuk Klasifikasi Penilaian Kredit. *Jurnal Informatika*, *5*, 175–185. https://doi.org/10.31294/ji.v5i2.4158

Turban, E., Kelly Rainer, R., & Richard, E. P. (2005). *Introduction to Information Technology* (3rd ed.). John Willey & Sons, Inc.

WIJAYANTI, N. P. Y. T., N. KENCANA, E., & SUMARJAYA, I. W. (2021). Smote: Potensi Dan Kekurangannya Pada Survei. *E-Jurnal Matematika*, *10*(4), 235. https://doi.org/10.24843/mtk.2021.v10.i04.p348

Wulandari, R. Y. (2018). Analisa Data Mining Dengan Metode Klasifikasi Untuk Produk Cacat Pada Pt. Shuangying International Indonesia. *Program Studi Teknik Informatika Sekolah Tinggi Teknologi Pelita Bangsa*, *14*(5), 2–5.