Grenze International Journal of Engineering and Technology, January Issue

Automation of Classification Tasks with Imbalanced Datasets using an Adapted CRISP-DM Methodology

Sasa Mitrovic[1] and Neven Vrcek[2]
[1]Faculty of Economics in Osijek, University of Osijek, Trg Lj. Gaja 7, 31000 Osijek, Croatia, Faculty of Organization and Informatics, University of Zagreb
Email: sasa.mitrovic@efos.hr
[2]Faculty of Organization and Informatics, University of Zagreb Pavlinska 2,42000 Varaždin, Croatia
Email: nvrcek@foi.hr

***Abstract*— The authors of this paper propose and evaluate the performance of an automated approach for solving classification tasks with imbalanced datasets using an adapted CRISP-DM methodology. Structured programming in Python was used to develop an object-oriented system that operationalises each phase of the CRISP-DM methodology - from business understanding to deployment - using specifically defined methods inside a class. Automation of the process is enabled by iteration over multiple datasets and by multiple oversampling techniques - SMOTE and ADASYN - with evaluation metrics such as balanced accuracy, F1 score for each class individually class and AUC-ROC score calculated in a standardised format. The results show consistent performance of the SMOTE method compared to ADASYN, with statistical validation using the McNemar test repeatedly confirming no significant difference ($p > 0.05$). The work highlights the importance of integrating statistical model testing into the CRISP-DM methodology and shows that a high degree of replicability, scalability and transparency in the analysis of imbalanced classification problems can be achieved using a modular approach. A Python implementation of an automated approach for solving classification tasks with imbalanced datasets using an adapted CRISP-DM methodology is available at the following GitHub link: https://github.com/smitroviefos/crisp_disbl_paper2025.**

***Index Terms*— CRISP-DM, automation of the analytical process, imbalanced datasets, classification, model evaluation and statistical validation.**

I. INTRODUCTION

The increasing application of machine learning methods in various industries and scientific fields has highlighted the need to address the problem of highly imbalanced classification tasks, which pose a particular challenge in this context. The asymmetry of distribution is present in numerous real-world business domains in the production environment, such as the detection of fraudulent credit card transactions [1] or malicious (phishing) websites [2], where underrepresented classes often have the highest semantic value. Although oversampling techniques have improved significantly, certain methodological shortcomings remain unresolved, as traditional methodological approaches often neglect the need for systematic integration of imbalance processing techniques into broader data project management.[3] To address this problem, the present paper proposes an automated system using the extended Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology [4-7], which is adapted for dealing with classification tasks with imbalanced datasets.

Grenze ID: 01.GIJET.12.1.326_1

Using an object-oriented approach in Python, a modular class was developed in which all phases of the CRISP-DM process (from business understanding to evaluation and deployment) [4-7] are operationally defined as methods. The automation enables the multiple evaluation of different data balancing techniques - SMOTE [8] and ADASYN[9] - over selected reference datasets, while performing evaluation metrics and statistical validation tests (McNemar[10] – binary classification tasks, Cochran Q [11] – multi-classification tasks). The results obtained indicate high accuracy and stability of the model using the SMOTE [8] technique, while the statistical analysis confirms no significant difference in performance compared to the ADASYN [9] technique in most cases, emphasising the need for a multidimensional approach to model evaluation. The present study shows that by combining machine learning methods, automated execution and statistical validation, one can improve the reliability, transparency and reproducibility of the classification process in class imbalance environments.

II. Literature Review

This study analyses the relationships between the automation of classification procedures, the presence of imbalance in the distribution of classes within datasets and the application of the modified CRISP-DM methodology. The focus is on identifying and systematising scientific findings that combine the above concepts in the context of current approaches in data science and machine learning. The study is based on a systematic analysis of the relevant literature available in the scientific databases SCOPUS and Web of Science. The search is performed using structured logical queries containing keywords relevant to the field of work. The first query focuses on the connection between the CRISP-DM approach and the problem of class imbalance in the context of data mining and machine learning. The second query additionally integrates terms related to the SMOTE technique and the evaluation of classification models. This makes it possible to identify research that deals with the integration of methodological frameworks and evaluation strategies in the processing of imbalanced data. This approach enables the development of a theoretical-empirical basis for further operationalisation of automated classification systems in the context of formalised data processes.

In analysing the relevant scientific literature, the paper focuses on the identified research gaps that have been recognised and highlighted by a number of authors.

In their article, authors Dablain, et al. [12] identify a methodological problem in the field of interpretable machine learning (IML), related to the limited adaptability of existing techniques to the specificities of imbalanced datasets. It was found that there are no systematic frameworks for deep neural networks and high-dimensional data spaces. Most of the available solutions focus exclusively on binary classification tasks in low-dimensional environments. These limitations point to the need to develop integrative approaches that extend the application of IML techniques to more complex scenarios, including the analysis of adversarial examples and their application to complex visual tasks such as the recognition of objects in indoor environments. In particular, they emphasise the need for a deeper understanding of the causes of misclassification within adversarial examples, which would enable a more robust interpretation of model behaviour and strengthen their reliability in security-sensitive environments .[12]

In their work, the authors Hancock, et al. [13] identify a research gap in the field of evaluating the efficiency metrics for classification tasks when dealing with highly imbalanced high-dimensional datasets (big data). The authors Hancock, et al. [13] point out that previous research has often ignored the influence of the method of random undersampling (RUS) of the majority class on the overall model performance, leading to insufficiently informed conclusions about the actual robustness of the machine learning classification method. The need for a systematic comparison between the Area Under the Precision-Recall Curve (AUPRC) and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) metrics is particularly emphasised, as the AUPRC provides a more sensitive and informative evaluation in the context of high class imbalance. The research by Hancock, et al. [13] aims to fill this methodological gap by analysing how different ratios between classes quantitatively affect the behaviour of the evaluation metrics, thus contributing to the development of standardised procedures for model evaluation in real-world scenarios with imbalanced data.[13]

The authors Lobo, et al. [14] point out that in the context of modern approaches for classifying imbalanced datasets, there is a need for the systematic exploration of alternative machine learning algorithms and the extension of the hyperparameter search space, which allows the identification of models with greater robustness and adaptability to class imbalance. The research by Lobo, et al. [14] highlights the importance of including different oversampling and undersampling strategies such as SMOTE, ADASYN or RUS in the modelling process to improve the representation of underrepresented classes. The authors Lobo, et al. [14] also point out the need for theoretical and applied development of non-sampling approaches, including the adjustment of decision boundaries and the sensitivity of the algorithm to imbalanced distribution, which remain understudied in

research. This extended methodology, according to Lobo, et al. [14], would provide a deeper understanding of the performance of models in complex classification environments and contribute to the development of more comprehensive frameworks for analysing imbalanced data.[14]
The research presented in this paper aims to specifically address the identified research gap resulting from the lack of formalised, repeatable and scalable approaches when dealing with imbalanced data distribution problems in classification tasks. The analysed literature mainly features a fragmented approach where individual data balancing techniques such as SMOTE and ADASYN are applied in isolation, without integrating them into a broader methodological process context. This paper therefore proposes a systematic methodological integration of oversampling, modelling and evaluation techniques within an adapted CRISP-DM methodology, which is rarely operationalised in its original form when dealing with class imbalance. By automating all phases of the process – from data preparation, modelling and evaluation to statistical validation of the results – a reproducible analytical pipeline is created that enables an objective comparison of models and algorithmic strategies. This not only fills the theoretical gap in the application of the CRISP-DM methodology for imbalanced classification tasks, but also improves the practice of information programming using the Python code available on GitHub link by introducing evaluation standards and reducing methodological subjectivity.

III. METHODOLOGY

In this work, the authors use the CRISP-DM methodology [4-7] that was extended and adapted for classification tasks with imbalanced datasets. It is implemented in six interconnected phases (business understanding, data understanding, data preparation, modelling, evaluation, deployment)[4-7], where each phase is operationalised in a Python class CRISPDMImbalancedClassification developed for the purpose of automated analysis.

A. Business understanding

The main goal was to develop a replicable/reproducible and scalable system for comparing the effectiveness of upsampling techniques for dealing with imbalanced data in classification models. The focus is on problematic business domains where class imbalance significantly affects prediction quality – in particular fraud detection [1] and phishing website detection [2].

B. Data understanding

Real-world, available datasets with high class imbalance were analysed. Visualisation and descriptive statistics revealed differences in class frequencies and confirmed the need for using data balancing techniques.

C. Data preparation

In the third phase of the adapted CRISP-DM methodology, i.e. data preparation, data processing was performed with the aim of providing a consistent and numerically suitable dataset for modelling and evaluation of machine learning models using the proposed framework. First, the dataset is divided into features $X \in \mathbf{R}^{n \times p}$ and a target variable $y \in \mathbf{R}^{n}$, where the attribute of the target variable is removed from the feature matrix: X = *df.drop(target_column) and* y = *df[target_column]*.
If y is a nominal variable, i.e. discrete and of the string type, a transformation into a numerical format is performed using an externally referenced method - LabelEncoder [15]:

$$\tilde{y} = LabelEncoder(y) \in \{0,1,...,k-1\} \qquad (1)$$

The data set is then stratified into a training subset and a test subset, whereby the original class distribution is retained in both subsets. The split ratio is 80:20, and the stratification can be formally presented as follows [16]:

$$(X_{train}, X_{test}, y_{train}, y_{test}) = train\_test\_split(X, y, stratify = y, test\_size = 0.2) \qquad (2)$$

For machine learning methods that are sensitive to a feature scale, scaling of the training and test data was performed. The standard scale is calculated based on the mean μ and standard deviation σ, where x_j is calculated for each feature [17]:

$$z_j = \frac{x_j - \mu_j}{\sigma_j} \quad \text{where} \quad \mu_j = \mathrm{E}\left[x_j\right], \quad \sigma_j = \sqrt{Var\left[x_j\right]} \qquad (3)$$

This transformation was applied separately to the training set (calculation of μ and σ) and then used for the transformation of the test set to avoid data leakage and ensure methodological consistency.

The resulting matrices X^*_{train} and X^*_{test} are prepared for the next phase of the adapted CRISP-DM methodology, i.e. modelling.

This approach to data processing ensures not only numerical stability and convergence of the model, but also methodological consistency within the adapted methodological framework based on the classification task automation.

D. Modelling

In the fourth, i.e. modelling phase of the adapted CRISP-DM methodology, an approach was used involving the pre-balancing of the class over the machine learning data followed by the training of the prediction model. The starting point is a training set of features $X_{train} \in \mathbf{R}^{n \times p}$ and associated classes $y_{train} \in \{0,1\}^n$, where n is the number of instances and p is the number of features. As the data exhibit high class imbalance, oversampling was performed using the method defined by the argument *self.sampler_name* - *SMOTE[8]* or *ADASYN[9]*. Oversampling generates an extended dataset [8, 9]:

$$(\tilde{X}, \tilde{y}) = sampler.fit\_resample(X_{train}, y_{train}) \tag{4}$$

where $\left|\tilde{y}_0\right| = \left|\tilde{y}_1\right|$, i.e. the number of instances of each class is balanced, mitigating the effects of class imbalance on the learning algorithm.

After balancing, the classification model defined in the variable *self.model* is trained on the newly created dataset $\left(\tilde{X}, \tilde{y}\right)$, within the adapted methodological framework. This is the random forest method of machine learning *[18-20]*, which is included in the ensemble method based on the bagging strategy. In mathematical terms, the decision function is trained $f : \mathbb{R}^p \to \{0,1\}$ in such a way that [18-20]:

$$f(x) = \text{argmax}_{k \in \{0,1\}} \text{P}(y = k \,|\, x; \theta) \tag{5}$$

where θ is the set of parameters of the machine learning model. The training duration is measured as the difference between the time after and before the fit method is called, which is recorded as follows:

$$t_{train} = t_{end} - t_{start} \tag{6}$$

and is stored in the *self.training_time* attribute, which enables subsequent evaluation of the time complexity of the random forest machine learning method *[18-20]*.

To summarise, in terms of methodology, this phase represents the integration of the process of balancing and training the model into a single automated sequence, which increases the robustness of the classification when dealing with imbalanced data and ensures the quantification of the time requirements of the machine learning process over the dataset.

E. Evaluation

In the fifth phase of the adapted CRISP-DM methodological framework, i.e. evaluation, a quantitative assessment of the performance of the trained classification model $f : \mathbb{R}^p \to \{0,1,...,K-1\}$ on unseen data from the test set is performed. First, discrete predictions $\hat{y} = f(X_{test})$ and corresponding class probabilities are obtained $P = f_{proba}(X_{test})$, where $P \in \mathbb{R}^{n \times K}$, and each row contains the distribution of probabilities by class for a given instance.

For a more accurate assessment when the data are imbalanced, metrics were used that take into account the difference in the representation of the classes.

The first of these metrics is the F1score by class, defined as the harmonic mean of precision and recall for each class $k \in \{0,1,...,K-1\}$ [21]:

$$F1_k = \frac{2 \times \text{Precision}_k \times \text{Recall}_k}{\text{Precision}_k \times \text{Recall}_k} \text{ gdje je } \text{Precision}_k = \frac{TP_k}{TP_k + FP_k},\ \text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \quad (7)$$

Another important metric is balanced accuracy, defined as the arithmetic mean of the sensitivity by class [22]:

$$\text{Balanced Accuracy} = \frac{1}{K} \sum_{k=1}^{K} \frac{TP_k}{TP_k + FN_k} \quad (8)$$

which ensures a fair evaluation for all classes, regardless of their frequency.
In addition, the AUC-ROC score [21] is calculated, which measures the ability of the model to distinguish between positive and negative classes. In binary classification, the probability of belonging to the positive class is used $P(y=1|x)$, i.e. the second column of the matrix $P[:,1]$, while in multi-class classification the target variables are binarised [23]:

$$Y_{bin} = label\_binarize(y_{test}) \quad (9)$$

and the AUC score is calculated using the *one-vs-rest* (*OvR*) method [24]:

$$AUC - ROC_{macro} = \frac{1}{K} \sum_{k=1}^{K} AUC_k \quad (10)$$

All results are recorded in the form of a classification report, including a confusion matrix [21]:

$$C = confusion\_matrix(y_{test}, \hat{y}) \quad (11)$$

which shows the distribution of correct and incorrect classifications by class. At this stage, one can make informed decisions about the acceptability of the model for business or scientific application and determine whether the model tends to favour the dominant class, which is a common problem with imbalanced datasets.

F. Deployment

The automated system is designed as a generic tool that can automatically perform all phases of the adapted CRISP-DM process for any classification dataset. The evaluation results and statistical comparisons are automatically generated in .csv and .log file formats and as graphical visualisations in .png format.
The deployment phase, i.e. the final phase of the adapted CRISP-DM process, focuses on interpreting the model performance for the purpose of making business, scientific or operational decisions. In the context of predictive analysis with imbalanced classification datasets, an important visual tool is the confusion matrix $C \in \mathbb{R}^{K \times K}$, where K is the number of classes and each element $C_{i,j}$ represents the number of instances of the actual class i that are predicted as class j.
Here, visualisation is performed using the function *sns.heatmap()*, which displays the matrix C in the form of a heatmap where the size and colour intensity of each square is related to the frequency of occurrence of a particular combination of actual and predicted class. This visualisation facilitates the detection of systematic model errors as well as the precise localisation of classification weaknesses, especially in the case of minority classes that are frequently misclassified.
The visualisation elements – the axis titles ("Actual" and "Predicted") and the graph title itself, which contains the name of the oversampling method and the dataset used – contribute to the comprehensive interpretability of the results and thus ensure transparency of the conclusion drawing process. The deployment phase thus goes beyond the purely technical implementation and becomes an important phase for interpreting the results of the analysis together with the stakeholders, thereby ensuring methodological consistency and the possibility of replication/reproducibility.

IV. Results and Discussion

The experimental part of the research is based on the application of the developed automated system on two real-world datasets: creditcard.csv[1] and phishing.csv[2]. The analysis was performed in the Python environment, where the CRISPDMImbalancedClassification class was developed, which in terms of methodology includes all phases of the adapted CRISP-DM process.

The models were trained on data that are oversampled by using SMOTE and ADASYN sampler methods and then evaluated on the same test division – 20 %. Training time was recorded for each model, as well as all accuracy and discrimination ability metrics. The evaluation results were structured and recorded to a .csv file for later analysis.

For each pair of models (SMOTE vs. ADASYN), statistical validation was performed using the McNemar test. The results of the p-value are stored and visualised to more clearly identify significant differences in the performance of the two classifiers.

TABLE I. CRISP-DM EVALUATION RESULTS

Dataset	Sampler	Training Time (s)	Balanced Accuracy	AUC-ROC	F1_Class_0	F1_Class_1	Compared With	Statistical Test	P-Value	Significant Difference
creditcard	SMOTE	280.7136	0.9132	0.9685	0.9997	0.8482				
creditcard	ADASYN	303.4711	0.8978	0.9638	0.9997	0.8168				
creditcard	SMOTE	280.7136	0.9132	0.9685	0.9997	0.8482	ADASYN	McNemar	0.0703	NO
creditcard	SMOTE	277.3088	0.9132	0.9685	0.9997	0.8482				
creditcard	ADASYN	300.2094	0.8978	0.9638	0.9997	0.8168				
creditcard	SMOTE	277.3088	0.9132	0.9685	0.9997	0.8482	ADASYN	McNemar	0.0703	NO
phishing	SMOTE	0.3785	0.9746	0.9975	0.9723	0.9782				
phishing	ADASYN	0.4022	0.973	0.9974	0.9703	0.9765				
creditcard	SMOTE	277.3088	0.9132	0.9685	0.9997	0.8482	ADASYN	McNemar	0.0703	NO
phishing	SMOTE	0.3785	0.9746	0.9975	0.9723	0.9782	ADASYN	McNemar	0.424	NO

The results presented in Table 1 represent a quantitative and statistically validated comparison of the performance of two oversampling approaches - Synthetic Minority Over-sampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN) - on two datasets with high class imbalance: creditcard.csv and phishing.csv. Given the nature of the problem, metrics sensitive to class imbalance were used in the evaluation: balanced accuracy, AUC-ROC score and F1 score by class to ensure methodological correctness and fair model evaluation.

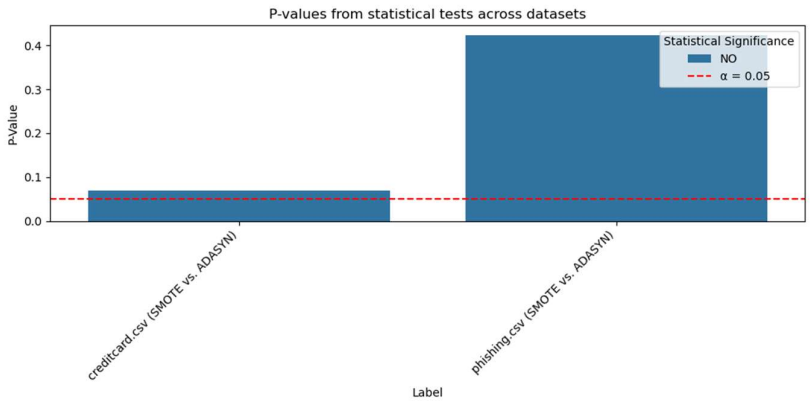

Figure 1. p-values from statistical tests across datasets and samplers

For the creditcard.csv the SMOTE method consistently achieves a higher balanced accuracy (0.9132) as well as higher F1 score for the positive class (0.8482) compared to ADASYN (0.8978, 0.8168), indicating a better

ability to detect the sparse class (fraud). The AUC-ROC score also confirms this superiority (SMOTE: 0.9685, ADASYN: 0.9638), although the difference is quantitatively small but still consistent. However, the results of the McNemar test (p=0.0703) show that despite the numerical differences, the difference in classification errors between the SMOTE and ADASYN models is not statistically significant at the 0.05 significance level. This indicates that the differences are due to chance rather than systematic superiority of one method.

For the phishing.csv dataset, both methods achieve extremely high evaluation results (e.g. F1 score > 0.97 for both classes), with SMOTE again being slightly superior to ADASYN in almost all metrics. The training time is negligible due to the small size of the dataset. The McNemar test again showed no statistically significant difference (p=0.424), confirming that both methods are comparable in their performance, with no evidence of superiority.

Overall, the results confirm that SMOTE, although methodologically simpler, achieves stable and robust performance on different data types. However, the lack of statistical significance suggests that additional criteria such as robustness to noise and time complexity need to be considered when choosing the optimal oversampling method in practice. This approach, which includes evaluation against multiple metrics and validation through inferential statistical testing, illustrates the application of a comprehensive and scientifically sound methodological framework in the field of automated access by means of an adapted CRISP-DM method for the systematic analysis of imbalanced data using machine learning methods.

The confusion matrices shown in Figures 2 and 3 provide a visual insight into the effectiveness of classification models trained on imbalanced datasets using the SMOTE and ADASYN oversampling methods. Each matrix shows the number of correctly and incorrectly classified instances by class, with the classifier axis represented by “predicted” and the actual class distribution represented by “actual”. These matrices allow a detailed analysis of the model's ability to distinguish between majority and minority classes, which is crucial in the context of imbalanced datasets.

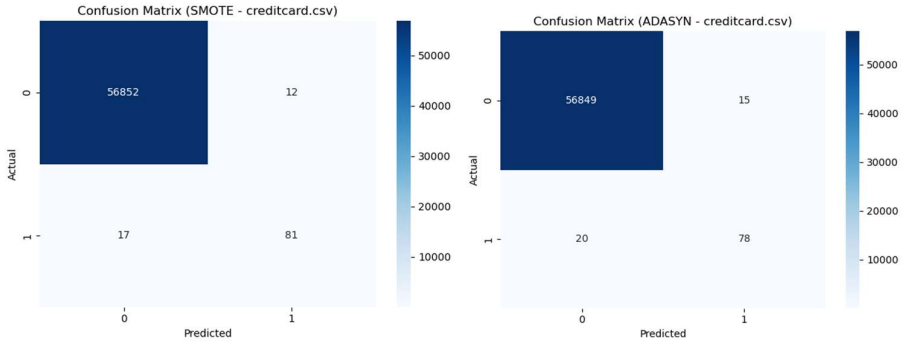

Figure 2: Confusion Matrix for creditcard dataset

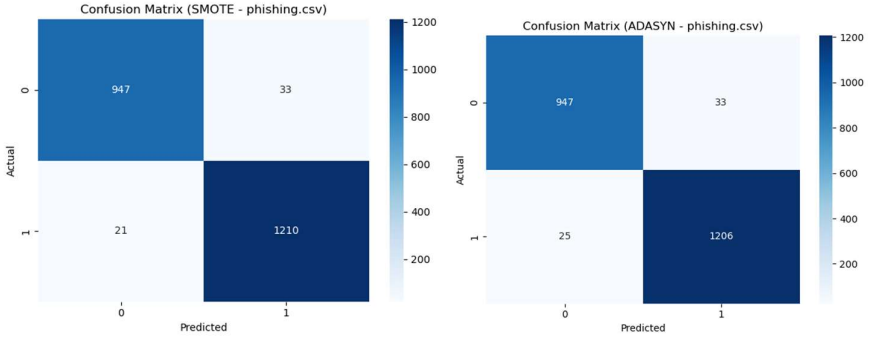

Figure 3: Confusion Matrix for phishing dataset

For the creditcard.csv dataset, which is extremely imbalanced, the SMOTE method correctly flagged 81 transactions as fraud (true positive, TP) with 17 false negatives (FN), while ADASYN correctly identified 78 TPs and generated 20 FNs. Although these differences are small in absolute terms, they are also reflected in the slightly lower F1 score for class 1 in ADASYN, indicating a slightly lower sensitivity of the model. The number

of FPs remains similar for both methods (12 for SMOTE and 15 for ADASYN), confirming the consistency of the model's accuracy for the majority class.
In the case of phishing.csv, which has a much more balanced distribution of classes compared to the previous dataset, both methods achieve high performance. The SMOTE model correctly classifies 1,210 phishing examples with 21 FN classifications, while ADASYN makes 1,206 correct classifications and 25 incorrect classifications. The number of FP classifications (33 in both cases) remains the same, confirming the stable accuracy in recognising non-phishing websites. These differences are numerically marginal and methodologically they do not represent a significant difference in the error distribution, which was also confirmed by the McNemar test.
To summarise, the visual analysis of the confusion matrices confirms the quantitative results - the SMOTE method shows a slightly higher sensitivity in detecting the minority class on both sets, but with no statistically significant difference. Methodologically, confusion matrices are an important tool for interpreting the imbalance between precision and recall and allow the detection of specific weaknesses of classifiers in real production environments.
The added value delivered by the proposed automated system, developed on the basis of the adapted CRISP-DM methodology, is demonstrated by its scalability – a new combination of model, dataset or balancing method can be easily integrated without the need to change the basic logic. This modularity and reproducibility form the basis for application in real-world business and scientific contexts where automated machine learning (AutoML) systems are used.

V. Conclusions

This paper discusses an extended and automated CRISP-DM methodology intended for classification tasks with imbalanced data. The object-oriented Python implementation operationalises all phases of the adapted CRISP-DM process, enabling a fully automated execution of the analytical cycle - from data loading, oversampling, modelling and evaluation to statistical validation and visual interpretation of the results.
The results for two real-world datasets show that the SMOTE method consistently delivers better metrics than the ADASYN method, although statistical tests do not confirm the significance of these differences. The inclusion of statistical validation into the adapted CRISP-DM framework further emphasised the importance of interpreting model performance not only through evaluation metrics but also through scientifically sound inference methods.
The proposed approach ensures a high level of reproducibility, transparency and flexibility in analysing imbalanced classification problems and represents a valuable scientific contribution to the methodological improvement of data science in both scientific and industrial settings.

APPENDIX A PYTHON IMPLEMENTATION OF AN AUTOMATED APPROACH TO SOLVING CLASSIFICATION PROBLEMS WITH IMBALANCED DATASETS USING AN ADAPTED CRISP-DM METHODOLOGY

Link: https://github.com/smitroviefos/crisp_disbl_paper2025

References

[1] U. L. B. Machine Learning Group, "Credit Card Fraud Detection," 2018.
[2] A. Prasad and S. Chandra, "PhiUSIIL Phishing URL (Website)," 2024.
[3] H. Ali, M. Salleh, K. Talpur, A. Ullah, A. Ahmad, and R. Naseem, "A review on data preprocessing methods for class imbalance problem," pp. 390-397, 10/04 2019, doi: 10.14419/ijet.v8i3.29508.
[4] E. Bratkovsky, "A simple explanation of CRISP-ML for beginners," vol. 2024, ed. Kaggle: Kaggle, 2024.
[5] P. Chapman, CRISP-DM 1.0: Step-by-step Data Mining Guide. SPSS, 2000.
[6] R. Costa, The CRISP-ML Methodology: A Step-by-Step Approach to Real-World Machine Learning Projects. https://www.amazon.com/CRISP-ML-Methodology-Step-Step-Real-World/dp/B0BR9DQMW5: Independently published, 2022, p. 75.
[7] C. Shearer, "The CRISP-DM Model: The New Blueprint for Data Mining," Journal of Data Warehousing, vol. 5, no. 4, 2000.
[8] K. W. Bowyer, N. V. Chawla, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," CoRR, vol. abs/1106.1813, 2011 2011. [Online]. Available: http://arxiv.org/abs/1106.1813.
[9] H. Haibo, B. Yang, E. A. Garcia, and L. Shutao, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 1-8 June 2008 2008, pp. 1322-1328, doi: 10.1109/IJCNN.2008.4633969.

[10] J. Perktold, S. Seabold, J. Taylor, and statsmodels-developers. "statsmodels.stats.contingency_tables.mcnemar." statsmodels. https://www.statsmodels.org/dev/generated/statsmodels.stats.contingency_tables.mcnemar.html (accessed 6.5.2025., 2025).
[11] J. Perktold, S. Seabold, J. Taylor, and statsmodels-developers. "statsmodels.stats.contingency_tables.cochrans_q." statsmodels. https://www.statsmodels.org/dev/generated/statsmodels.stats.contingency_tables.cochrans_q.html (accessed 6.5.2025., 2025).
[12] D. Dablain, C. Bellinger, B. Krawczyk, W. Aha, and N. Chawla, "Understanding imbalanced data: XAI & interpretable ML framework," Machine Learning, vol. 113, pp. 1-19, 01/16 2024, doi: 10.1007/s10994-023-06414-w.
[13] J. T. Hancock, T. M. Khoshgoftaar, and J. M. Johnson, "Evaluating classifier performance with highly imbalanced Big Data," Journal of Big Data, vol. 10, no. 1, p. 42, 2023/04/11 2023, doi: 10.1186/s40537-023-00724-5.
[14] A. Lobo, P. Sampaio, and P. Novais, "Enhancing quality 4.0 and reducing costs in lot-release process with machine learning-based complaint prediction," (in English), TQM J., Article vol. 36, no. 9, pp. 175-192, 2024, doi: 10.1108/TQM-10-2023-0344.
[15] s.-l. developers. "LabelEncoder." scikit learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html (accessed 20.6.2024., 2024).
[16] s.-l. developers. "train_test_split." scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (accessed 9.5.2025., 2025).
[17] s.-l. developers. "StandardScaler." scikit learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html (accessed 6.6.2024., 2024).
[18] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001/10/01 2001, doi: 10.1023/A:1010933404324.
[19] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, Classification and Regression Trees. Taylor & Francis, 1984.
[20] s.-l. developers. "RandomForestClassifier." scikit learn. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed 6.6.2024., 2024).
[21] V. Jayaswal. "Performance Metrics: Confusion matrix, Precision, Recall, and F1 Score." Insight Media Group. (accessed 8.5.2025., 2025).
[22] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The Balanced Accuracy and Its Posterior Distribution," in 2010 20th International Conference on Pattern Recognition, 23-26 Aug. 2010 2010, pp. 3121-3124, doi: 10.1109/ICPR.2010.764.
[23] s.-l. developers. "label_binarize." scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.label_binarize.html (accessed 8.5.2025., 2025).
[24] s.-l. developers. "OneVsRestClassifier." scikit-learn https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html (accessed 8.5.2025., 2025).