

Cluster-Specific Rule Mining for Argumentation-Based Classification

Jonas Klein^[0009–0000–6521–5991], Isabelle Kuhlmann^[0000–0001–9636–122X], and
Matthias Thimm^[0000–0002–8157–1053]

FernUniversität in Hagen, Hagen, Germany
{jonas.klein, isabelle.kuhlmann, matthias.thimm}@fernuni-hagen.de

Abstract. We present a multi-step classification approach that combines classical machine learning methods with computational models for argumentation. In the first step, the dataset is divided into different groups using a clustering algorithm. In the second step, we employ rule-learning algorithms to extract frequent patterns and rules from each resulting cluster. In the last step, we interpret the rules as the input for structured argumentation approaches. Given a new observation, we first assign it to one of the previously generated clusters. Subsequently, the classification of the observation is determined by formulating arguments based on the respective cluster-specific rules for the different classes. Finally, the justification status of the arguments is determined using the argumentative inference method of the structured argumentation approach.

Keywords: Argumentation · Classification · Rule Mining.

1 Introduction

Classification is a widely known problem in the field of artificial intelligence. In recent years, machine learning approaches, in particular different forms of neural networks, have made substantial progress in solving classification tasks for a diverse range of domains—such as computer vision [15], text processing [10], or graph theory [19]. However, although current machine learning methods for classification purposes may yield remarkably accurate results, they are still not *guaranteed* to be correct, and they are not inherently explainable, i. e., no form of justification or rationale is provided. On the other hand, the need for explainable methods is becoming increasingly relevant [4].

To address the problem of lacking explainability in machine learning-based classification approaches, Thimm and Kersting [16] propose an approach that combines machine learning with computational models of argumentation [5]. To be precise, the authors suggest a two-step procedure: first, a rule learning algorithm is applied to extract rules from a given dataset; in the second step, the learned rules are used as input for a structured argumentation system, which then yields a justification status for each class, given a new observation. Thus, this approach does not only deliver classifications but also explanations thereof.

Further, expert knowledge (in the form of additional arguments) can easily be incorporated into the reasoning process.

Thimm and Kersting [16] presented some preliminary experimental results in their study: on the *Animals with Attributes* (AwA) dataset, about 30% of the instances were classified correctly, while the remaining 70% were deemed “undecided”.¹ In this paper, we build explicitly on these results and present an extended approach that likewise includes a rule mining step and an argumentation-based classification step, which introduces a clustering technique for more targeted rule mining. More specifically, the clustering step reduces the number of mined rules to make them more purposeful and additionally counteracts the extraction of contradictory rules. In an experimental analysis, we show that our method can achieve a significantly higher accuracy of 71% on the AwA dataset. To corroborate our observations, we consider additional datasets. Furthermore, we demonstrate that the procedure introduced in this work is potentially significantly more resource-efficient than the approach proposed by Thimm and Kersting.

2 Background

The three main ingredients of the approach presented in this paper are (1) a clustering algorithm, (2) a rule mining algorithm, and (3) a structured argumentation method. Although the choice of each component is generally flexible, we select some concrete instantiations of each component as an example. For the clustering part, we use a simple *k-modes* algorithm [12], which is aimed at clustering categorical variables. As the rule mining algorithm we use *association rule mining*, and as the structured argumentation approach, following [16], we use *defeasible logic programming* [9]. Both latter formalisms are outlined below.

Association Rule Mining Data mining generally encompasses methods for extracting non-trivial patterns from a given dataset. *Association rule mining* [3] aims to uncover interesting relationships among items within extensive databases. Consider $I = \{I_1, I_2, \dots, I_m\}$ as a set comprising m distinct attributes. Let T be a transaction containing a set of items such that $T \subseteq I$, and let D be a database with various transaction records T_s . An *association rule* takes the form of $X \Rightarrow Y$, where $X, Y \subseteq I$ represent sets of items known as *itemsets*, and $X \cap Y = \emptyset$. In this context, X is referred to as the *antecedent*, and Y is termed the *consequent*. The rule $X \Rightarrow Y$ signifies that the presence of X implies the presence of Y . Association rules rely on two fundamental criteria of interestingness: *support* and *confidence*. These criteria help identify relationships and rules by revealing frequently occurring if/then patterns. To be considered, association rules typically must meet both a user-specified minimum *support* and

¹ Note that the authors used defeasible logic programming (DeLP) [9] as the structured argumentation approach, and DeLP does not only use “yes” and “no” as answers, but also “undecided”.

a user-specified minimum *confidence* simultaneously. The *support* of an association rule is defined as the fraction of records that contain $X \cup Y$ relative to the total number of records in the database. The *confidence* of an association rule is defined as the fraction of the number of transactions that contain $X \cup Y$ relative to the total number of records that contain X . For the approach presented here, we use *FP-Growth* [11] as a rule miner.

Defeasible Logic Programming The core idea behind *defeasible logic programming* (DeLP) [9] is to combine concepts from logic programming and defeasible argumentation to allow for dealing with incomplete or contradictory data. A defeasible logic program (de.l.p.) consists of facts and rules which are divided into *strict* rules of the form $l \leftarrow B$ and *defeasible* rules of the form $l \rightarrow B$, with l being a single literal and B a set of literals. Moreover, a *fact* is a single literal (i. e., an atom a or a negated atom $\neg a$). Thus, formally, a de.l.p. $\mathcal{P} = (\Pi, \Delta)$ consists of a set Π of facts and strict rules, and a set Δ of defeasible rules. Further, a literal l is *derivable* by some set of rules R (i. e., $R \sim l$) if it is derivable following the classical rule-based understanding. If both $R \sim l$ and $R \sim \neg l$, then R is *contradictory*. Conventionally, Π is non-contradictory. Further, if $R \sim l$, and $R \not\sim \perp$, we call the literal l *consistently derivable* (denoted as $R \sim^c l$).

For a de.l.p. $\mathcal{P} = (\Pi, \Delta)$ and a literal l , a tuple $\langle \mathcal{A}, l \rangle$ (with $\mathcal{A} \subseteq \Delta$) is an *argument* for l iff $\Pi \cup \mathcal{A} \sim^c l$ and \mathcal{A} is minimal wrt. set inclusion. Further, $\langle \mathcal{B}, q \rangle$ is a *subargument* of $\langle \mathcal{A}, l \rangle$ iff $\mathcal{B} \subseteq \mathcal{A}$. We refer to $\langle \mathcal{A}_1, l_1 \rangle$ as a *counterargument* to $\langle \mathcal{A}_2, l_2 \rangle$ at literal l , iff there is a subargument $\langle \mathcal{A}, l \rangle$ of $\langle \mathcal{A}_2, l_2 \rangle$ with $\Pi \cup \{l, l_1\}$ being contradictory. To deal with counterarguments, we use the *generalized specificity* relation \succ as a formal comparison criterion among arguments. According to this criterion, an argument is preferred over another, if (1) it has a greater information content and is thus more precise, or (2) it uses fewer rules and is thus more concise (see Garcia and Simari [9] for a formal definition and further discussion). We call $\langle \mathcal{A}_1, l_1 \rangle$ a *defeater* of $\langle \mathcal{A}_2, l_2 \rangle$ iff there is a subargument $\langle \mathcal{A}, l \rangle$ of $\langle \mathcal{A}_2, l_2 \rangle$ such that $\langle \mathcal{A}_1, l_1 \rangle$ is a counterargument of $\langle \mathcal{A}_2, l_2 \rangle$ at literal l and either $\langle \mathcal{A}_1, l_1 \rangle \succ \langle \mathcal{A}, l \rangle$ (*proper defeat*) or $\langle \mathcal{A}_1, l_1 \rangle \not\succeq \langle \mathcal{A}, l \rangle$ and $\langle \mathcal{A}, l \rangle \not\succeq \langle \mathcal{A}_1, l_1 \rangle$ (*blocking defeat*).

A finite sequence of arguments $\Lambda = [\langle \mathcal{A}_1, l_1 \rangle, \dots, \langle \mathcal{A}_m, l_m \rangle]$ is an *acceptable argumentation line* iff (1) every $\langle \mathcal{A}_i, l_i \rangle$ with $i > 1$ is a defeater of $\langle \mathcal{A}_{i-1}, l_{i-1} \rangle$ and if $\langle \mathcal{A}_i, l_i \rangle$ is a blocking defeater of $\langle \mathcal{A}_{i-1}, l_{i-1} \rangle$ and $\langle \mathcal{A}_{i+1}, l_{i+1} \rangle$ exists, then $\langle \mathcal{A}_{i+1}, l_{i+1} \rangle$ is a proper defeater of $\langle \mathcal{A}_i, l_i \rangle$, (2) the sets $\Pi \cup \mathcal{A}_1 \cup \mathcal{A}_3 \cup \dots$ and $\Pi \cup \mathcal{A}_2 \cup \mathcal{A}_4 \cup \dots$ are non-contradictory, and (3) there exists no $\langle \mathcal{A}_k, l_k \rangle$ as a subargument of $\langle \mathcal{A}_i, l_i \rangle$ with $i < k$. Thus, intuitively, an argumentation line forms a sequence of arguments, in which each $\langle \mathcal{A}_i, l_i \rangle$ defeats its predecessor $\langle \mathcal{A}_{i-1}, l_{i-1} \rangle$. Moreover, since an argument $\langle \mathcal{A}_i, l_i \rangle$ defeats $\langle \mathcal{A}_{i-1}, l_{i-1} \rangle$, and therefore reinstates $\langle \mathcal{A}_{i-2}, l_{i-2} \rangle$, the sets $\Pi \cup \mathcal{A}_1 \cup \mathcal{A}_3 \cup \dots$ and $\Pi \cup \mathcal{A}_2 \cup \mathcal{A}_4 \cup \dots$ must be non-contradictory in order for the argumentation line to be acceptable. To avoid circular argumentation, we also need to ensure that no subarguments are reintroduced in the same argumentation line.

Finally, a literal l is *warranted* if there is an argument $\langle \mathcal{A}, l \rangle$ which is non-defeated in the end. To decide whether $\langle \mathcal{A}, l \rangle$ is defeated or not, every acceptable

argumentation line starting with $\langle \mathcal{A}, l \rangle$ has to be considered. The answer is to a DeLP query is YES if l is warranted, and NO if $\neg l$ is warranted. Otherwise, the answer is UNDECIDED.

3 Cluster-Specific Rule Mining

The approach proposed in this work is an extension of the *argumentation-based classification* approach (AbC) described by Thimm and Kersting [16]. The AbC approach consists of two steps: (1) Mining of association rules from a given dataset and (2) performing classification using the generated rules as an input to a structured argumentation approach. During the initial phase, algorithms for rule mining are employed to identify frequent patterns and rules from a specified dataset. The result of this step yields a substantial number of rules [17]. However, these rules cannot be directly applied to classification since they often exhibit inconsistencies. Hence, in the subsequent phase, these rules are used as input to structured argumentation methods, such as DeLP. Employing the argumentative inference procedures inherent in these approaches, the classification of the new observation is executed by formulating arguments based on these rules and evaluating their justification status. Using argumentation techniques enables the creation of classifiers explicitly designed to explain their decisions, thus meeting the contemporary demand for explainable AI. These classifiers are able to explain the reasons for favoring arguments supporting the conclusion over counterarguments.

We extend the original two-step argumentation-based classification approach AbC to a multi-step classification method, that combines traditional machine learning methods with structured argumentation. To be precise, we introduce two additional steps. Firstly, we perform a clustering of the input data, resulting in groups of instances with similar properties. Secondly, a feature selection is carried out for each cluster to identify the most informative features for the prediction of the target variable. Subsequently, these features are used to generate cluster-specific association rules for each cluster. Since the number of generated rules significantly influences the classification time, this approach leads to significantly shorter runtimes and is more resource-efficient. In addition, grouping instances with similar properties leads to discovering relationships that are difficult to detect when looking at the entire dataset. This improves the capability to classify datasets where a naive approach may not extract enough rules. Moreover, the generated rule set is more consistent due to the similarity of the instances within a cluster and the emphasis on meaningful features, improving the decidability of instances and thus reducing the number of undecidable instances. In general, the presented approach consists of four steps: (1) Clustering the input data, (2) cluster-specific feature importance analysis to select the most informative features, (3) cluster-specific association rule mining based on these features, and (4) classification of new observations by assigning them to a cluster and using the cluster-specific rules. Each step is outlined in more detail below.

Clustering First, the input data is divided into k groups based on all features (including the target feature), using the k -modes [12] algorithm. The k -modes clustering algorithm modifies the well-known k -means clustering method for partitioning a dataset into distinct groups or clusters based on categorical data. This step aims to divide the input data into smaller, more manageable groups with similar properties to reduce the running time of the rule mining algorithm, reduce the number of rules generated, improve the detection of otherwise hard to find relations and improve the quality of the rules.

Feature Selection This step conducts a feature importance analysis to find the most informative features for classifying the target variable within a cluster using the mutual information score. Mutual information quantifies the relationship between two random variables with a value that is always non-negative, indicating their dependency level. This value is zero exclusively when the two variables are independent, with larger values indicating a greater dependency. The score calculation is based on entropy estimation using distances from k -nearest neighbors, as outlined in [13] and [14]. After calculating the scores for each feature, the top k features are selected. The selected cluster-specific features are used as the input for the rule miner in the next step. The association rule mining step is massively accelerated by reducing the number of features and discarding features with little expressiveness. Furthermore, only the most relevant features are used for rule mining, leading to fewer, more meaningful rules.

Association Rule Mining This step generates cluster-specific association rules for each previously generated cluster based on the most important selected features. In this work, we use the *FP-Growth* [11] algorithm. In principle, however, any rule mining algorithm is usable. To generate rules from the truth values of the features of an instance, these are represented as a set of ground literals. For example, for a dataset of animals with the attributes *swims*, *black*, and *arctic*, the attributes of a dolphin would be represented as *swims(dolphin)*, \neg *black(dolphin)*, and \neg *arctic(dolphin)*. The output of the rule mining algorithm is a set of association rules such as $flippers(X) \rightarrow ocean(X)$, which can be interpreted as “animals with flippers live in the ocean”. Subsequently, the created rules are filtered according to the method of Thimm and Kersting [16]: Rules with more than one element in the conclusion and more than three elements in the body are discarded. All rules with confidence value 1 are interpreted as strict; the remaining rules are interpreted as defeasible.² Occasionally, no or not enough cluster-specific rules are generated for the target variable, resulting in instances assigned to this cluster not being able to be classified. To prevent this, we implemented an adaptive rule mining process, which iteratively adjusts the confidence and support values until at least one rule for the target variable is generated.

² We followed this procedure to ensure the best comparability with the original approach. A systematic analysis of different rule filtering techniques and strict/defeasible thresholds is out of the scope of this work and saved for future work.

Classification In the classification step, the cluster-specific rules are used as input to the structured argumentation approach DeLP (see Section 2). To classify a new instance, it is first assigned to a cluster using the previously trained clustering algorithm to determine the classification rule set. Since the value of the target variable is unknown, the assignment is performed twice, whereby (1) the target variable is assumed to be positive and (2) it is assumed to be negative. If, for example, one aims to classify the edibility of a mushroom with the classes *edible* and *poisonous*, an unseen mushroom is once assumed to be *edible* and another time assumed to be *poisonous*. Two cases can occur: The mushroom is either assigned to the same cluster in both cases or to different clusters. In the first case, the rules of the corresponding cluster are applied, and the classification is carried out. In the second case, two classifications are performed with the different rules of the respective cluster. Since two different sets of rules from different clusters are used, and different assumptions are made about the class of the target variable, conflicting classifications may occur. For example, a mushroom m is assigned to cluster C_0 for the negative assumption (*poisonous*) and to cluster C_1 for the positive assumption (*edible*). The query $poisonous(m)$ returns the answer UNDECIDED for C_0 . For C_1 , the answer is YES. Since the results do not match, one of the two answers must be selected. In general, two types of conflicts can arise: (1) The rules of one cluster return UNDECIDED, and the rules of the other cluster return a concrete answer YES/NO, or (2) one cluster returns YES and the other NO. The first conflict is resolved by choosing the concrete answer (YES/NO) as the final result. In the second case, the answer of the rule set with the higher average confidence is used. If the average confidence matches, the average support is used as a tiebreaker.

4 Experimental Analysis

In this section, we present the results of an experimental analysis, in which we compare our approach³ to AbC [16] in terms of the classification performance on five different datasets. Below, we describe the experimental setup and subsequently discuss our findings.

Datasets and Setup We use five well-known categorical datasets for binary classification as training and test data: *Animals with Attributes*, *Zoo*, *Mushrooms*, *Car Evaluation*, and *Congressional Voting Records*.

All categorical features that are not already in binary form were one-hot encoded by converting each feature into as many 0/1 features as there are different values. For a dataset with, for example, the feature *safety*, which has two different values, low and high, two new dummy features, *safety_low* and *safety_high*, have been introduced. For each instance, the feature’s value was then replaced by the corresponding one-hot encoding. Records with missing values were excluded.

We make use of the following five datasets.

³ <https://github.com/jklein94/Cluster-Specific-Rule-Mining-for-Argumentation-Based-Classification>

Animals with Attributes The *Animals with Attributes (AwA)* dataset consists of 50 different animals with 85 boolean-valued attributes. The dataset was randomly split into 90% training data and 10% test data. The *Zoo* [8] dataset is similar to the *AwA* dataset. It contains 101 instances of animals with 16 attributes. The dataset was randomly split into 80% training data and 20% test data.

Mushrooms The *Mushrooms* [2] dataset comprises descriptions of imaginary samples representing 23 species of mushrooms. Each species is categorized as either edible, poisonous, or of uncertain edibility. The latter category was merged with the poisonous one. The dataset initially consists of 8124 instances with 22 categorical features. After the data cleaning and feature encoding, the dataset contains 5644 instances with 99 features and was randomly split into 90% training data and 10% test data.

Car Evaluation The *Car Evaluation (Car)* [6] dataset was derived from a simple hierarchical decision model. The original dataset contains 1728 instances with 6 categorical features. After one-hot encoding, a total of 22 features resulted. No instance was excluded. The classification target is determining whether a car exhibits a low safety standard. The dataset was randomly split into 80% training data and 20% test data.

Congressional Voting Records The *Congressional Voting Records (Congress)* dataset [1] consists of 1984 US Congressional Voting Records for each of the U.S. House of Representatives Congressmen. Initially, it contains 435 instances and 16 features. After removing the records with missing values and encoding the features, 232 instances with 33 features remain. The classification target is to determine which party (Democratic or Republican) a congressman voted for. The dataset was randomly split into 80% training and 20% test data.

We repeated the classification five times for each dataset according to the procedure described in Section 3. The number of randomly initialized clusters was set to seven. For each cluster, the top four features were selected. We set the minimum support of the rule mining algorithms to 0.7 and the minimum confidence to 0.9.⁴ We use the accuracy, percentage of undecided instances, and percentage of decided, but falsely classified instances to evaluate the performance of the proposed approach. The mean result of the five runs is reported. Note that we randomly selected ten attributes for the *AwA* dataset as target variables. The average of the metrics across all ten selected attributes is reported. We randomly selected three target variables for the *Zoo* dataset. The results for each selected attribute are reported individually.

⁴ The values used showed promising results in preliminary experiments, achieving a good balance between the number of generated rules, classification performance, and runtime. A systematic analysis of the parameters is beyond the scope of this work and will be part of future work.

Table 1. Overview of accuracy (ACC), undecidable instances (UNDEC), and decided but falsely classified instances for our approach and the AbC approach. The results of AbC for the *AwA* dataset are those reported in [16].

Name	Ours			AbC		
	ACC	UNDEC (%)	False (%)	ACC	UNDEC (%)	False (%)
AwA	0.71	17.20	12.0	0.30	70.0	0.00
Zoo_eggs	1.00	0.00	0.00	0.00	100.0	0.00
Zoo_milk	0.96	0.95	2.86	0.00	100.0	0.00
Zoo_fins	0.92	2.86	4.76	0.86	9.52	4.76
Mushrooms	0.88	10.80	1.13	-	-	-
Car	0.82	15.26	2.77	0.0	100.0	0.00
Congress	0.87	4.68	8.09	0.00	100.0	0.00

Results The results in Table 1 show that AbC could not classify even one instance for five of the seven scenarios. To be precise, for the classification of *Zoo_eggs*, *Zoo_milk*, *Car*, and *Congress* all test instances were answered as UNDECIDED, leading to an accuracy of 0. Our approach, on the other hand, consistently achieves high accuracies ranging from 0.82 (*Cars*) to 1.0 (*Zoo_eggs*). For *Zoo_fins*, both approaches show 4.76 % of falsely classified instances. However, our approach exhibits a significantly lower proportion of UNDECIDED instances, reflected in a higher overall accuracy of 0.92 compared to AbC (0.86). In addition, in our experiments, AbC created a very large number of rules for the *AwA* dataset, which precluded classification in a reasonable time, which is why we rely on the results reported in [16]. Although the results can only be compared to a limited extent, our cluster-specific approach shows significantly higher accuracy (0.71 vs. 0.3) and a significantly smaller proportion of undecidable instances (17.2 % to 70 %) than AbC. The most extensive dataset *Mushrooms* could not be classified by AbC because it ran out of memory in the rule mining step. Our method achieves an accuracy of 0.88, with 10.8 % of instances remaining undecided and a low 1.13 % false classification rate.

5 Limitations

The approach presented in this paper achieves promising results in terms of accuracy and the reduction of undecidable instances. However, there is still room for improvement. In the following, we will discuss some main limitations of the proposed approach.

Rule Generation Control The classification performance heavily depends on the generated rules. However, direct control over the rule-generation process is limited to setting support and confidence thresholds. Another way to influence rule generation is through clustering and feature selection. Finding the best parameters for the clustering and feature selection steps is a non-trivial task that ultimately comes down to trial and error as it is very dataset-dependent.

Computational Overhead Compared to traditional machine learning methods, our approach can result in longer classification times due to its multi-step nature. Each step has a computational overhead, and the runtime of the clustering algorithm, the rule mining algorithm, and the DeLP implementations significantly influence our approach’s runtime performance.

Classification Tasks and Data Types Our method’s design primarily targets binary classification tasks, focusing on handling categorical variables. In its current configuration, achieving multi-class classification necessitates multiple invocations of the classification pipeline—one for each class. This requirement can significantly heighten computational demands, potentially detracting from overall performance efficiency. Moreover, the approach’s specialization in categorical variables necessitates that numeric features undergo a binning process to be transformed into categorical equivalents. This transformation can lead to an exponential increase in the number of features, substantially expanding the feature space.

6 Conclusion

In this work, we presented a new approach to argumentation-based classification. Building on the preliminary results of Thimm and Kersting [16], we developed a multi-step classification approach that combines classical machine learning methods with approaches to (structured) argumentation. In an experimental analysis, we examined the classification performance on five different dataset and showed that our cluster-specific rule mining approach achieves significantly better accuracies and lower numbers of undecidable instances than the original AbC approach. In future work, we aim to explore the influence of different configurations for the clustering, feature selection, and rule mining steps and their impact on classification performance. Furthermore, broadening the scope of evaluation to encompass datasets of increased complexity and diversity and a comparative analysis with other argumentation-based methods like ABALearn [18] and AA-CBR [7], other symbolic learners and traditional machine learning approaches would be of great interest. Moreover, efforts to improve scalability and computational efficiency are paramount. Optimizing the approach to handle larger datasets efficiently without sacrificing explainability or classification accuracy is critical for practical use. Finally, extending our methodology to efficiently tackle multi-class classification tasks and accommodate diverse data types, including continuous and multi-modal datasets, represents a significant frontier for exploration.

Acknowledgement. The research reported here was partially supported by the Deutsche Forschungsgemeinschaft (grant 375588274).

References

1. Congressional Voting Records. UCI Machine Learning Repository (1987), DOI: <https://doi.org/10.24432/C5C01P>
2. Mushroom. UCI Machine Learning Repository (1987), DOI: <https://doi.org/10.24432/C5959T>
3. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD international conference on Management of data. pp. 207–216 (1993)
4. Angelov, P.P., Soares, E.A., Jiang, R., Arnold, N.I., Atkinson, P.M.: Explainable artificial intelligence: an analytical review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **11**(5), e1424 (2021)
5. Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics. *The knowledge engineering review* **26**(4), 365–410 (2011)
6. Bohanec, M.: Car Evaluation. UCI Machine Learning Repository (1997), DOI: <https://doi.org/10.24432/C5JP48>
7. Cocarascu, O., Stylianou, A., Čyras, K., Toni, F.: Data-empowered argumentation for dialectically explainable predictions. In: ECAI 2020, pp. 2449–2456. IOS Press (2020)
8. Forsyth, R.: Zoo. UCI Machine Learning Repository (1990), DOI: <https://doi.org/10.24432/C5R59V>
9. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. *Theory and practice of logic programming* **4**(1-2), 95–138 (2004)
10. Gasparetto, A., Marcuzzo, M., Zangari, A., Albarelli, A.: A survey on text classification algorithms: From text to predictions. *Information* **13**(2), 83 (2022)
11. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. *ACM sigmod record* **29**(2), 1–12 (2000)
12. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery* **2**(3), 283–304 (1998)
13. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. *Physical review E* **69**(6), 066138 (2004)
14. Ross, B.C.: Mutual information between discrete and continuous data sets. *PLoS one* **9**(2), e87357 (2014)
15. Sharma, S., Guleria, K.: Deep learning models for image classification: comparison and applications. In: 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering. pp. 1733–1738. IEEE (2022)
16. Thimm, M., Kersting, K.: Towards argumentation-based classification. In: Logical Foundations of Uncertainty and Machine Learning, Workshop at IJCAI (2017)
17. Thimm, M., Rienstra, T.: Approximate reasoning with ASPIC+ by argument sampling. In: Proceedings of the Third International Workshop on Systems and Algorithms for Formal Argumentation (SAFA’20) (September 2020)
18. Tirsi, C., Proietti, M., Toni, F.: Abalearn: An automated logic-based learning system for aba frameworks. In: International Conference of the Italian Association for Artificial Intelligence. pp. 3–16. Springer (2023)
19. Xiao, S., Wang, S., Dai, Y., Guo, W.: Graph neural networks in node classification: survey and evaluation. *Machine Vision and Applications* **33**, 1–19 (2022)