Determining chromatic index of cubic graph with the use of explainable classifiers: A comparative study

A. DUDÁŠ AND B. MODROVIČOVÁ

Abstract

Proper edge 3-coloring of a cubic graph is an NP-complete problem, which can be used in order to model several interesting practical problems. A method for correct and efficient assigning of variables used in the program to registers of the system, scheduling of a set of tasks to a set of processors while each task has to be executed on a number of processors simultaneously, or frequency assignment of radio stations without interference are all typical instances of problems modelled by graph coloring. When considering cubic graphs, which consist of vertices incident to precisely three edges, we can identify two distinct graph groups divided by their chromatic index, a minimal number of colors needed for the proper coloring of such graph. In the research presented in this article, we conduct a comparative study of the effectiveness of machine learning classifiers in the task of determining the chromatic index of cubic graphs, present an evaluation of the accuracy and precision of these models, and use the Shapley Additive Explanations model for the identification of graph attributes and their values crucial in the models' decision making.

Mathematics Subject Classification 2000: 68T20, 68T10
General Terms: Predictive Data Analysis, Cubic Graph Classification
Keywords: Chromatic Index, Explainable Artificial Intelligence, Classification Models, Cubic Graphs.

1. INTRODUCTION

The chromatic index of the graph determines a minimal number of colors needed for its proper edge coloring. Such coloring of a graph is an NP-complete problem, which consists of assigning colors to the edges of the cubic graph in such a way that none of the vertices of the graph is incident to two edges colored with the use of the same color [Marx 2004]. In this way, we are able to distinguish between two discrete groups of cubic graphs:

—standard edge 3-colorable cubic graphs, where the number of colors needed for proper edge coloring of a graph is equal to three,
—and snarks, or edge 3-uncolorable cubic graphs, where the number of colors is higher - more specifically four (see Section 1.1).

The determination of the chromatic index of a graph can, therefore, be projected into the classification task of determining whether an input graph G is an instance of

10.2478/jamsi-2024-0006

a standard cubic graph class or snark class. The research presented in this article is focused on searching for properties and design of methods, which could be used to identify the number of colors needed to edge color a graph more effectively. Since modern machine learning classification models reach a high level of accuracy in classification but are hard to interpret, we aim to use techniques of explainable artificial intelligence to understand how the model classifies the graphs into considered classes.

The presented research contains a comparative study of the effectiveness of machine learning models in the task of determining the chromatic index of a cubic graph. The contribution of the article can be summarized as follows:

—Presentation of original cubic graph property dataset usable in any machine learning model. For the proper description of the dataset in the context of the selected problem, we present a correlation analysis of this dataset and we visualize the dataset with the use of two- and three-dimensional Principal Component Analysis method.

—Description of interpretable machine learning model that uses created datasets in order to classify input graph sets into one of two considered classes: properly edge 3-colorable graphs or improperly edge 3-colorable graphs. The model uses standard well-regarded classifiers from Naïve Bayes to Multilayer Perceptron Networks.

—Evaluation of the model on the basis of its classification accuracy, precision, and interpretability of the decisions made. The interpretability of the model is important from the point of view of identification of properties which are significant in the context of graph edge coloring and values of these attributes which contribute to the decisions made by the classification model. For these purposes, the Shapley values are used through the Shapley Additive Explanation method. In the case, that these properties can be computed in lower time complexity than edge coloring itself, we can obtain the chromatic number of the graph in lower time.

Other than the introduction itself, the introductory section of the work contains a brief description of the determining of the chromatic index of cubic graphs and considered classification models. In Section 2 we describe the cubic graph property dataset and model used in the classification of cubic graphs with classifiers specified in Section 1.2. Section 3 of the presented article contains an evaluation and comparison of machine learning classifiers in the selected tasks. The evaluation is done from the point of view of accuracy, precision, and interpretability of each model. In the Conclusion section, we summarize the findings presented in the article,

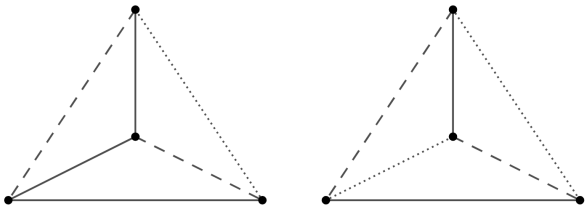

Fig. 1. Example of improperly edge 3-colored cubic graph (L) and properly edge 3-colored cubic graph (R)

describe the strengths and weaknesses of the work, and offer several other directions for the development of research in the given area.

1.1. Determining Chromatic Index of Cubic Graph

Cubic graph G consists of sets V (vertices) and E (edges) where [Caro et al. 2023]

$$G = (V,E), E \subseteq V^2$$

and each of the vertices of G is incident to exactly three edges - every vertex is of degree three.

As described in [Chaitin 1982], proper edge coloring of the graph is a problem consisting of assigning colors to the edges of the graph in such a way, that there is one instance of each color incident to every vertex (see Fig. 1). This condition makes the edge coloring of a graph an instance of an NP-complete problem. Vizings' theorem for graph G is formulated as [Vizing 1964]

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$$

where $\chi'(G)$ is the chromatic index of the graph G (minimal number of colors needed in order to properly edge color the graph G) and $\Delta(G)$ is the maximum degree of the graph G. From the point of view of this theorem, we consider two possibilities of proper edge coloring of cubic graphs, either with the use of three or four colors.

From the point of view of this article, Vizings' theorem is used to determine two classes of cubic graphs:

—Standard cubic graphs, where $\Delta(G) = 3$ and $\chi'(G) = 3$.

—Snarks or edge 3-uncolorable cubic graphs, where $\Delta(G) = 3$ but $\chi'(G) = 4$.

There is a number of algorithms and optimizations proposed for the task of chromatic index identification. The most basic approach to this problem is the naive backtracking algorithm, which works on the principle of the gradual coloring of the edges of the graph with a predetermined sequence of three colors. When the algorithm finds a contradiction in the coloring of the graph, it returns to the previous edge, which is recolored and the algorithm continues with the next coloring of the graph. If neither of the possible colorings of the problematic edge is proper, the algorithm returns to the edge preceding both recolored edges. The algorithm continues this procedure until the entire graph is properly edge colored or until the algorithm goes through all the possibilities of coloring the graph. The time complexity of edge backtracking is $O(2^{|E(G)|})$, where $|E(G)|$ is the number of edges in graph G.

Better time complexity ($O(2^{0.427|V(G)|})$, where $|V(G)|$ denotes the number of vertices of the colored graph G) was reached by Kowalik's EdgeColor algorithm [Kowalik 2009]. The algorithm itself consists of EdgeColor, FittingMatching, and SetSwitches procedures while using the approach of generating inclusion-maximum matchings of the graph.

In our previous work presented in [Dudas and Modrovicova 2023], we examined the possibilities of increasing the efficiency of the computation of proper edge $k-$coloring of cubic graphs with the use of machine learning methods. The main focus of the research is the use of a machine learning model of decision trees for the problem of identification of properly edge 3-uncolorable graphs (snarks). This approach used datasets with homogeneous sizes of graphs and therefore was able to reach an average accuracy of 93%. In the presented work, we build on these results with the objective of comparing and increasing the accuracy and precision of the binary classification of heterogeneously sized cubic graphs.

1.2. Considered Classifiers

Classification is a task in which we consider a pattern x and space X and an algorithm estimates which value of the associated attribute $y \in \{1,...,n\}$ will be acquired by the pattern x [Al-Azawi 2022]. Therefore, in the problem of classification, we allocate entities to a set of pre-defined classes of individuals who are, in a certain sense, similar to other individuals within a given class.

There is number of classification models, whether machine learning-based or not. This study considers the following classifiers for the purposes of classification of the chromatic index of cubic graphs:

—**Naïve Bayes** classifier is a machine learning model used for probabilistic classification based on Bayes' theorem [Al-Azawi 2022]:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where B represents values of a given set of graph properties, A denotes one of two considered classes and P denotes probability. The classification with the use of Naïve Bayes classifier is based on a computation of probabilities of input data belonging to each of the existing classes. The class with the highest measured probability is the class to which the input data is classified [Al-Azawi 2022].

—**Support Vector Machine** is a kernel-based classification model that uses a hyperplane to separate the data from one dimension to high dimensional space. In the case the values of attributes are not linearly separable in the input space, the Support Vector Machine can transform the data to the high dimension space through nonlinear transformations [Pineiro and Portillo 2022].

—**Decision tree** is a graph-based classification model, which is constructed from the root to the leaves of the tree. Each node in the graph represents an attribute of a given dataset with a set border for the value of the attribute, while branches connect nodes that are above or below the set border of attribute value. The decision tree model separates the dataset into multiple subsets where each of the subsets represents one branch from the root node. In this way, the method creates a classification tree with class designations stored in the leaves of the graph [Custode and Iacca 2023].

—**Random forest** is an ensemble classification model based on decision trees. The model constructs decision trees on subsets of the input dataset, produces and accumulates predictions on each as described above, and then votes on the final classification. The class defined by the majority of trees generated by random forests is considered to be the classification of the input data by random forest itself [Pineiro and Portillo 2022].

—The last classification model considered in this study is the neural network approach, more specifically **Multilayer Perceptron Network**. This neural network model consists of three types of layers: the input layer, which receives the input dataset for processing; an arbitrary Number of hidden layers that perform data transformation; and the output layer which is responsible for the classification of input values [Guellil et al. 2020].

2. GRAPH DATASET AND PROPOSED CLASSIFICATION MODEL

The first part of this work is focused on a collection of datasets that are created for the purpose of building classification models for the identification of the chromatic index of cubic graphs. These datasets need to contain the proper amount of graphs and the proper amount of graph properties measured on these graphs.

As the main source of data for interesting graphs (snarks and some of the standard cubic graphs) studied in this work, we used the *House of Graphs* portal [Coolsaet et al. 2023]. However, the portal does not list the properties of the graphs in a consistent format, and it does not contain a sufficient number of standard cubic graphs that meet our criteria. Therefore, we used the software tools *graphFilter* [GraphFilter 2021] and *SageMath* [SageMath 2005] to compute some of the considered graph properties and to generate the number of cubic graphs.

2.1. Description of Graph Properties Dataset

The presented dataset consists of 2000 cubic graphs in equal distribution of data between the two considered graph classes - 1000 standard cubic graphs and 1000 snarks. For each graph G from the created dataset, we measured ten indicative properties. In the following list, we present characteristics of the property and describe measured values by aggregation as the set $\langle minimum, mean, median, maximum\rangle$ [Dudas and Modrovicova 2023]:

—**Number of vertices** of the graph G. The study uses 30, 32, 34 and 36 vertex cubic graphs in equal distribution of 500 each. These are the most frequented sizes of snarks and, therefore, represent the best examples of the given graph type. Values of the number of vertices in the dataset can be aggregated to the set:

$$\langle 30, 33, 32, 36\rangle$$

—**Diameter** of graph G is the length of the longest path in the graph G. Statistical values of central tendency measures for the property are:

$$\langle 4, 6.29, 6, 11\rangle$$

—**Edge connectivity** of the graph G is the minimum number of edges, which can be deleted in such a way, that the graph G is disconnected into more than one component. Values of edge connectivity are described as follows:

$$\langle 1, 2.8, 3, 4\rangle$$

—**Matching number** of graph G is the number of edges that do not have a set of common vertices. The property can be aggregated to the set:

$$\langle 1, 16.48, 16, 18 \rangle$$

—**Radius** of the graph G is the minimum graph eccentricity of any graph vertex in a graph. The eccentricity of the graph vertex is the maximal number of edges between the vertex and any other vertex of G. Central tendency measures for the property reach the following values:

$$\langle 3, 4.78, 5, 8 \rangle$$

—**Largest L-eigenvalue** is largest eigenvalue of Laplacian matrix L of the graph G, while $L = D - A$, where A is adjacency matrix of G and D is diagonal matrix containing degree of the vertex i on each position $D_{i,i}$. Aggregation of values of the largest L-eigenvalue are:

$$\langle 5.36, 5.68, 5.68, 6.83 \rangle$$

—2^{nd} **largest eigenvalue** of graph G is the second largest eigenvalue of the adjacency matrix of G. The values of the property can be described as:

$$\langle -2.39, 2.65, 2.7, 5.75 \rangle$$

—**Smallest eigenvalue** of graph G is the smallest eigenvalue of the adjacency matrix of G. The values of the property are aggregated to the set:

$$\langle -3, -2.67, -2.68, -2 \rangle$$

—**Girth** of graph G is the length of the shortest cycle (in the case there is a cycle in the graph) in G. The girth of the graph measured in the presented dataset is described in the following set:

$$\langle 3, 5.2, 5, 8 \rangle$$

—**Chromatic index** of a graph G is the minimal number of colors needed in order to color the edges of the graph G properly. This property is used as a class identifier for the specified task, either G is a snark (chromatic index of four) or a standard cubic graph (chromatic index of three).

The prediction potential of the created dataset measured by Pearson, Spearman rank, and Kendall rank correlation coefficients is presented in Fig. 2. This correlation heatmap presents values of individual correlation coefficients with the use of the blue-to-red color scheme, the dark blue color signifying strong correlation between

graph properties and dark red color signifying the opposite - strong anticorrelation of pair of graph properties.

As can be seen in all of the presented heatmaps, most of the correlation values are not strong enough for simple linear models to be effective, even though there are some decent correlation values such as the correlation between the number of vertices and the matching number of a graph. However, from the point of view of chromatic index identification, the strongest of correlation coefficient values are in the area of moderate correlation of values, which is traditionally not deemed satisfactory for building prediction models. Therefore, methods which are more complex, and take into account more than one graph property at a time, should be used to classify the data as needed.

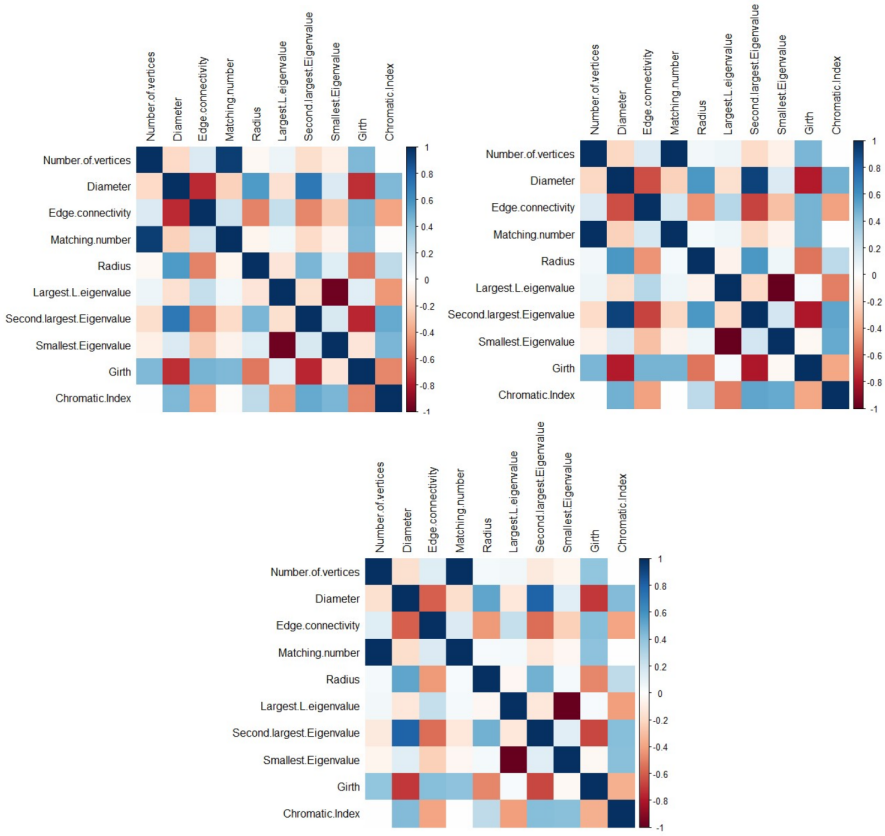

Fig. 2. Correlation heatmaps for the created graph property dataset: Pearson method (top, left), Spearman Rank method (top, right), and Kendall method (bottom)

For the visual representation of the created dataset, we used two- and three-dimensional Principal Component Analysis (PCA) method (Figs. 3, 4). In both of these visualizations of the dataset, we used color of points in order to separate classes of graphs. We can see that the dataset contains significant overlap between the values of principal components of both considered classes. This property of the dataset is noticeable mainly in $\mathbb{R}^3$ visualization presented in Fig. 4.

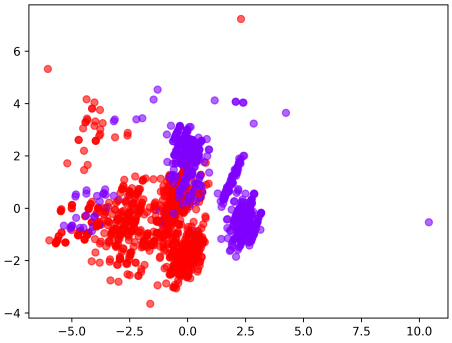

Fig. 3. Visualization of projection of created dataset into $\mathbb{R}^2$ with the use of PCA method

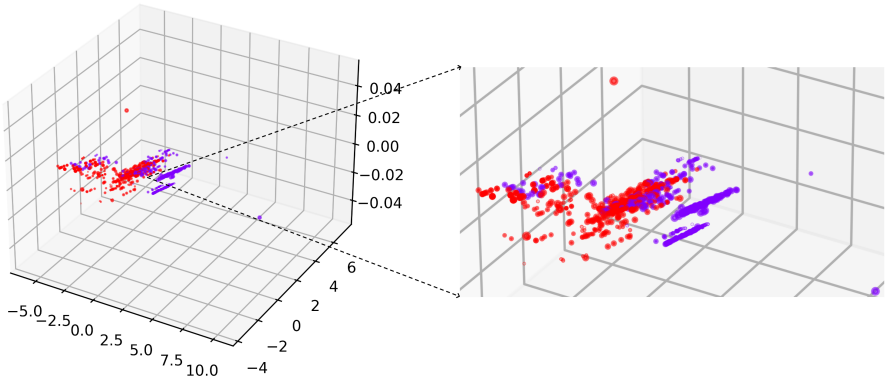

Fig. 4. Visualization of projection of created dataset into $\mathbb{R}^3$ with the use of PCA method

2.2. Proposed Classification Model

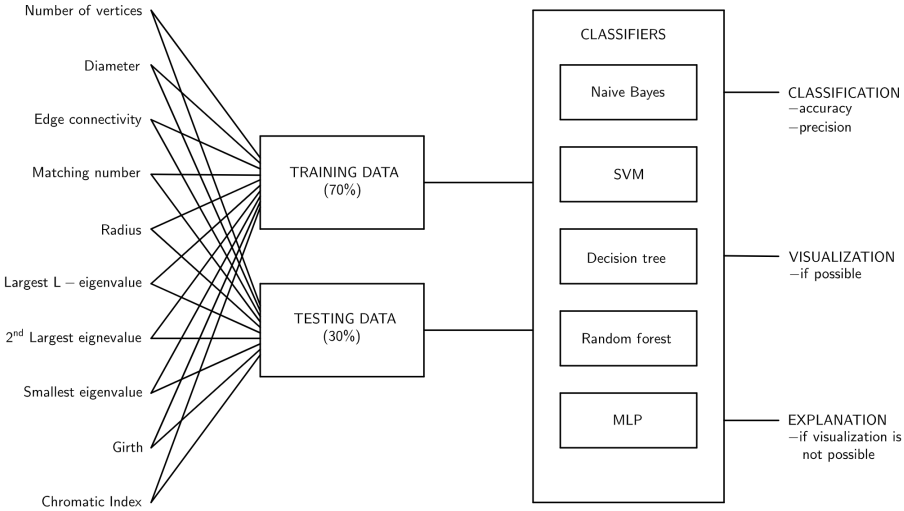

Fig. 5. Schema of used classification model

The research presented in this article aims towards the use of interpretable classifiers in order to identify the chromatic index of the input graph. Fig. 5 describes the proposed machine learning system as follows:

—The created data is randomly divided into two data subsets: training data, on which the classification model is learned, and testing data, which are used to evaluate the quality of the predictions of the created model. The ratio of the distribution of input data to the training and testing subset is 70% to 30%.

—Both subsets of the input data are subsequently used in the classification module of the system. This module is based on one of the considered classifiers (Naïve Bayes, Support Vector Machine, Decision tree, Random forest or Multilayer perceptron network) used with various settings of model parameters. The output of this module consists of two values - classification of graph determined by the used model and explanation or visualization of the decision process.

—The quality of the model is evaluated based on three metrics: accuracy of the model, precision of the classification of input graph into individual classes and importance of graph property values measured by Shapley values.

3. EVALUATION OF THE CONSIDERED CLASSIFIERS

To evaluate the decision-making quality of the created classification model, we measured the accuracy and precision of the classification of graphs from the created dataset. By accuracy we denote the decision-making quality of the model as a whole - we measure the closeness of the predicted value to the real value of the feature. Precision refers to the ability of the created classification model to identify only the relevant entities [Guellil et al. 2020].

To compute the accuracy and precision of the classification model, we need to compile confusion matrices for all measurements. Accuracy is computed on the basis of such matrix as follows [Vashisht and Jatain 2023]:

$$accuracy = \frac{t_n + t_p}{t_n + t_p + f_p + f_n}$$

where t_n is the number of true negative samples, t_p is the number of true positive samples, f_p is the number of falsely positive samples and f_n is the number of false negative samples.

Similar to the accuracy, the precision of the model is computed from the values of confusion matrices as follows [Li et al. 2023]:

$$precision(C) = \frac{t_p}{t_p + t_n}$$

where C is considered class (in our case standard cubic graph or snark), t_p is the number of true positive samples and t_n is the number of true negative samples.

To interpret the decisions of the created model, we use Shapley values through the Shapley Additive Explanations technique. This method measures how individual graph properties or sets of graph properties contribute to the overall quality of the created random forest model.

Even though the Shapley values are a technique of local interpretation of the model, by correct aggregation, we get a usable global interpretation of the model's decisions. The Shapley value ϕ for the i-th graph property (feature) is computed as [Molnar 2019]:

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

where x is the specific graph considered by the model, f is the created random forest model, z' is the subset of features whose contribution to the decision is measured, x' is the simplified form of the graph x (sum of its' features) and M is the number of graph

properties active in the used model.

3.1. Naïve Bayes

To classify input data instances, Naïve Bayes classifier uses computation of probabilities of input data belonging to each of the existing classes. The class with the highest measured probability is the class to which the input data is classified. The confusion matrix of the Naïve Bayes classifier used on the graph property dataset is presented in Tab. 1.

Table I. Confusion matrix for Naïve Bayes classifier

	standard	snark
standard	277	34
snark	88	202

Even though the true positive and true negative classifications are much higher in number compared to false positive or false negative ones, the Naïve Bayes model reached unsatisfactory accuracy and precision of classification:

$$accuracy = 80\%$$

$$precision(standard) = 76\%$$

$$precision(snark) = 86\%$$

Figure 6 presents the distribution of Shapley values for individual graph properties used in the model. As we can see in the figure, the most important features in the decision-making process were the edge connectivity of the graph and the smallest and largest eigenvalue of the adjacency matrix of the graph. With the graph property of edge connectivity, we can see a clean separation between the high (pink) and low (blue) values of this graph property. From this, we can conclude that low values of edge connectivity of the graph have the greatest impact on the decision-making process of the Naïve Bayes classifier.

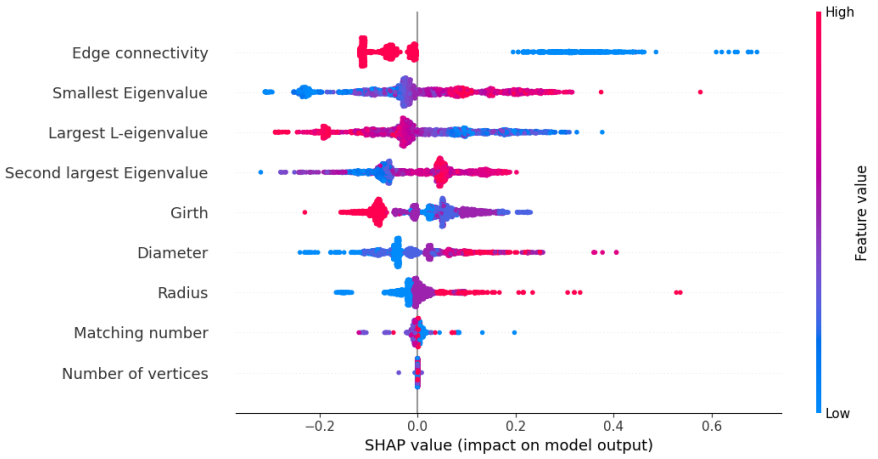

Fig. 6. Visualization of Shapley values for Naïve Bayes classifier

3.2. Support Vector Machine

A Support Vector Machine (SVM) constructs a hyper-plane (or set thereof) in a high dimensional space, which can be used for classification purposes. As can be seen in Figs. 3 and 4, the presented dataset is not linearly separable and therefore such a hyper-plane could be a tool to achieve desired results. Table 2 contains a confusion matrix for the Support Vector Machine classifier of the graph property dataset.

Table II. Confusion matrix for Support Vector Machine classifier

	standard	snark
standard	185	126
snark	0	290

As can be seen in the confusion matrix, this model is less accurate in overall classification, but achieves very high precision of identification of standard cubic graphs (with chromatic index equal to 3), more specifically:

$$accuracy = 79\%$$

$$precision(standard) = 100\%$$

$$precision(snark) = 70\%$$

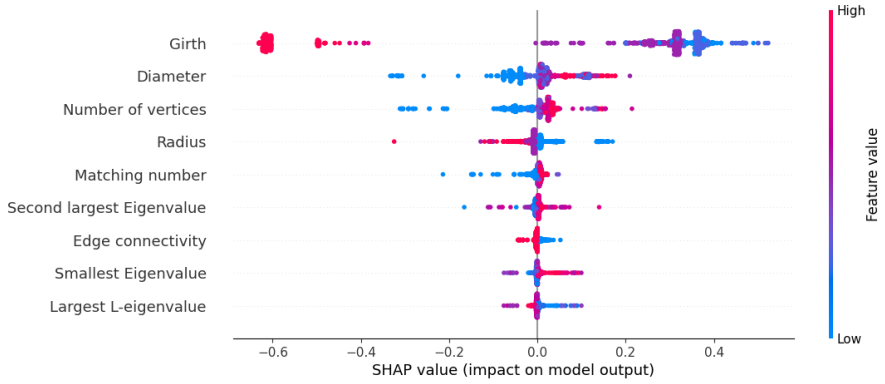

Fig. 7. Visualization of Shapley values for Support Vector Machine classifier

Interestingly enough, in Figure 7, we can see completely different graph properties on the top of importance than in the previous case. For classification with the use of a Support Vector Machine classifier, the most important graph properties are the girth of the graph, the diameter of the graph and the number of vertices of the graph. Similarly, as with the Naïve Bayes classifier, with the most important graph property we see a clear separation of Shapley values - high values of girth reach low Shapley values and low values of girth reach high Shapley values. The average values of this property move in the interval between zero impact on model output and high values of the girth of the graph.

3.3. Decision Tree and Random Forest

A decision tree is a classification model where each node in the graph represents an attribute of the given dataset with a set border for the value of the attribute, while branches connect nodes that are above or below the set border of attribute value. The decision tree trained on the presented dataset is structured as shown in Fig. 8.

The ensemble of decision trees constructed on subsets of the input dataset is called a random forest, in which the output class is defined by the majority of trees by a voting mechanism. Tab. 3 contains confusion matrices for the classification of graph property dataset for single decision tree and random forest of sizes 10, 100, and 500 respectively.

The confusion matrices point to generally decent accuracy and precision in all used models and forest sizes. The comparison of accuracy and precision of the classification with the use of individual tree-based models is presented in Tab. 4 and

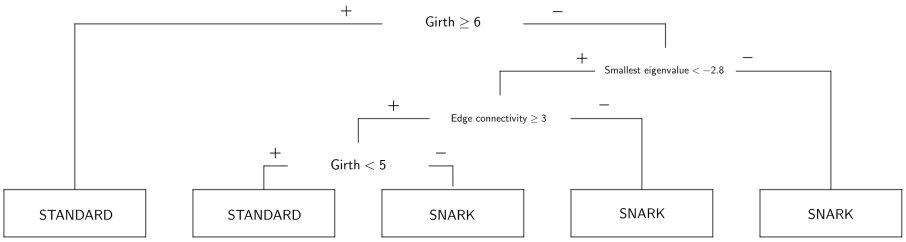

Fig. 8. Decision tree for the graph property dataset

Table III. Confusion matrices for decision tree and random forest of three sizes (10, 100, and 500 trees)

	1 tree		10 trees		100 trees		500 trees	
	standard	snark	standard	snark	standard	snark	standard	snark
standard	280	31	285	26	282	29	282	29
snark	31	259	24	266	21	269	19	271

Fig. 9.

Table IV. Comparison of accuracy and precision of the decision tree and random forest classifiers

	Accuracy	Precision (standard)	Precision (snark)
1 tree	90	90	89
10 trees	**92**	92	**91**
100 trees	**92**	93	90
500 trees	**92**	**94**	90

Clearly the accuracy and precision of classification of graphs with the use of a single decision tree is lower when compared to any of the random forest models. The accuracy of the random forest classifier was constant for all the presented sizes of the forest, but the precision of the classification of graphs into standard cubic graph class increased with each size increment. The precision of classification of input data into snark class was highest in the random forest of size 10.

When we compare the structure of the decision tree in Fig. 8 with the distribution of Shapley values in the random forest of the size 500 (Fig. 10), we see that one of the three properties most essential for decision-making processes, the girth of the graph, coincide. The decision tree, then, uses the smallest eigenvalue of the adjacency matrix of the graph, while the random forest model deems the second largest eigenvalue of the adjacency matrix as more important. The third graph property used in models also varies. In the decision tree, edge connectivity of the graph is used, similarly to Naïve Bayes classifier, where the property reaches the highest importance levels. For

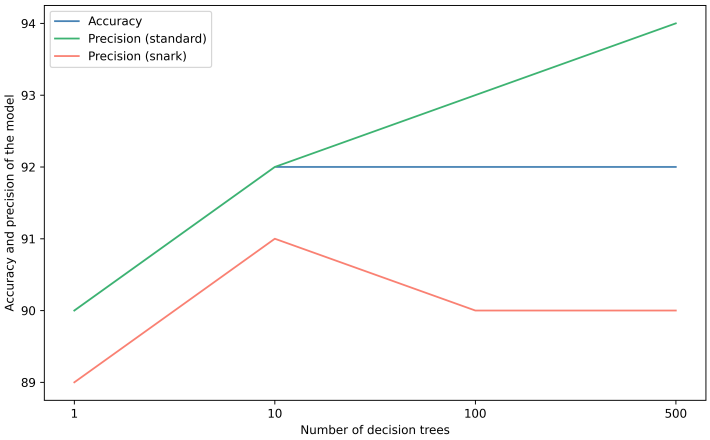

Fig. 9. Comparison of accuracy and precision of the decision tree and random forest classifiers

random forest, the third most important graph property is the number of vertices of the graph. We can see slightly different behavior with the girth of the graph compared to the previous cases. With random forest, the average values of girth contribute the most to the model output, low values are located close to the zero impact line and high values of this property reach negative Shapley values.

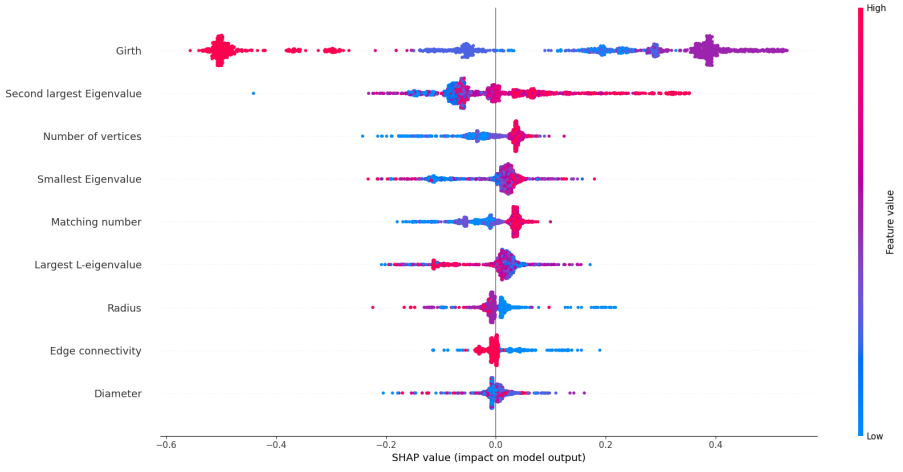

Fig. 10. Visualization of Shapley values for Random Forest classifier of size = 500

3.4. Multilayer Perceptron Network

The presented multilayer perceptron network consists of the following:

—Input layer with nine input neurons, one for each graph property.

—Number of hidden layers that perform data transformation. The number, size, and activation function used in the hidden layers are set experimentally and presented in Tab. 5 and Fig. 11.

—Output layer which is responsible for the identification of the chromatic index of the graph with the use of classification.

Table V. Comparison of accuracy and precision of various multilayer perceptron network configurations

$\mathbb{R}$(hidden)	f	Accuracy	Precision (standard)	Precision (snark)
10, 8	ReLU	79	81	78
10, 10	ReLU	79	**100**	70
10, 10	Sigmoid	92	92	91
10, 10	tan_h	**93**	94	**92**

The first classification results comparable to other presented classifiers were achieved with a multilayer perceptron network with two layers of 10 and 8 neurons respectively using the ReLU activation function. By adding a pair of neurons to the second layer, we achieved an increase in the precision of classification to the class of standard graphs, but we also noticed a decrease in the precision of classification of snarks. When replacing the ReLU function with a sigmoid function, we achieved overall results of accuracy and precision comparable to the random forest model presented in the previous section of the work. However, the overall performance of the multilayer perceptron model was best with the use of the hyperbolic tangent activation function. The structure of this neural network is presented in Fig. 12.

The confusion matrix from the classification process with the use of two hidden layer, hyperbolic tangent multilayer perceptron network is presented in Tab. 6.

Table VI. Confusion matrix for Multilayer Perceptron ($\mathbb{R}^9 \to \mathbb{R}^{10} \to \mathbb{R}^{10} \to \mathbb{R}^1$) classifier

	standard	snark
standard	286	25
snark	19	271

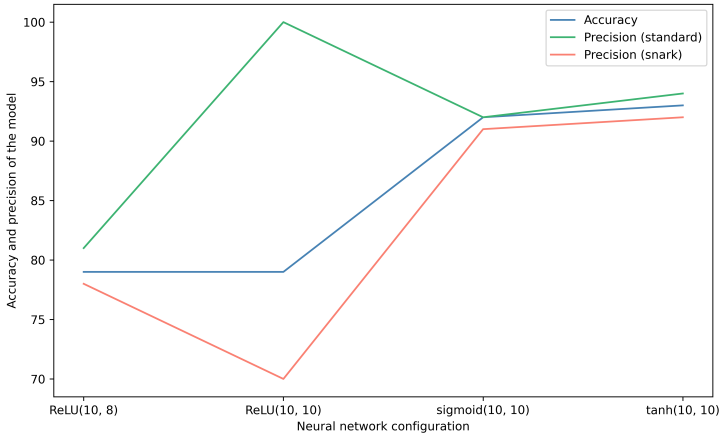

Fig. 11. Comparison of accuracy and precision of various multilayer perceptron network configurations

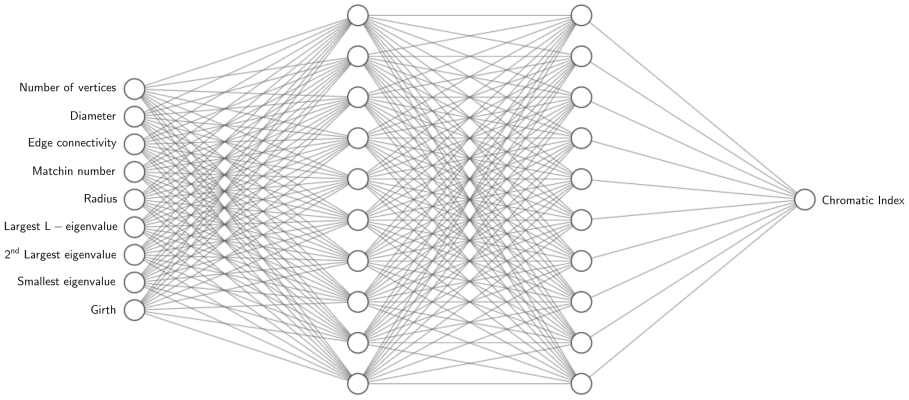

Fig. 12. Structure of Multilayer Perceptron Network (nine input neurons, two hidden layers of size 10 and one output neuron)

Figure 13 presents the distribution of Shapley values for the multilayer perceptron network, which matches the previous classifiers in the top graph properties. The girth was present to a certain extent in each of the previous classification processes, edge connectivity reached high importance in Naïve Bayes and decision tree classifiers and second largest eigenvalue was used in the classification using random forests. In previous cases, however, we saw a somewhat clear separation in Shapley values of the girth of the graph. However, with a multilayer perceptron network, we can see the intermingling of high, low, and even medium values of this property.

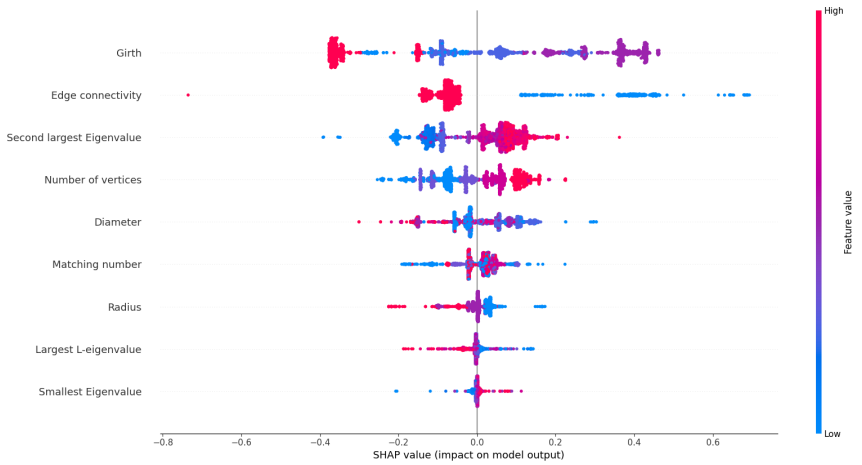

Fig. 13. Visualization of Shapley values for the proposed MLP network

3.5. Comparison of Considered Classifiers

Tab. 7 and Fig. 14 present a comparison of the accuracy and precision of the classification of input graphs into one of the considered classes for five well-known machine learning algorithms. We can conclude, that not all of the considered classifiers were sufficient for the task of determining the chromatic index of a cubic graph. The highest overall accuracy of classification and precision of the classification of snarks was reached with the use of a multilayer perceptron neural network with the following structure:

—Input layer containing one neuron for each one of nine input graph properties.

—Two hidden layers consisting of ten neurons each, using the hyperbolic tangent activation function.

—Output layer of one neuron responsible for the classification of graphs into one of two considered classes.

The highest precision of classification of input graphs into standard cubic graph class was reached with the use of the Support Vector Machine model.

Regarding the time of training of the model, simpler models such as Naïve Bayes, Support Vector Machine and single decision tree were less time-consuming in comparison to more complex models like random forest of 500 trees and multilayer perceptron network with the structure specified above. However, the difference between the training times of these two groups of models was not significant at all.

With the higher number of graphs in the training set of data, the difference might become more notable. Classification times were comparable with the use of both, simple and complex classification models.

Table VII. Comparison of accuracy and precision of the considered classifiers

	Accuracy	Precision (standard)	Precision (snark)
Naïve Bayes	80	76	86
SVM	79	**100**	70
Decision Tree	90	90	89
Random Forest (500)	92	94	90
MLP (10,10, tan_h)	**93**	94	**92**

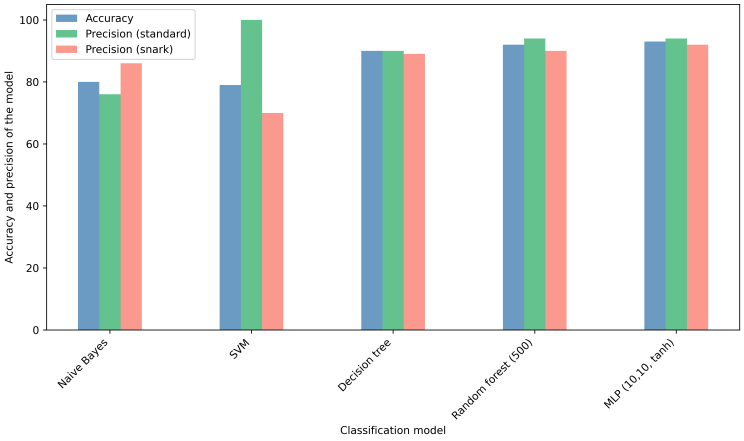

Fig. 14. Comparison of accuracy and precision of the considered classifiers

4. CONCLUSION

The presented research points to the possibility of applying prediction analysis techniques to the precise estimation of graph property values. Since most of the algorithms that are used in the measurement of graph properties do not use machine learning and classification models, the potential of prediction analysis described in this article can be used in the optimization of the measurement of graph property values.

In the work, we present a comparison of five machine learning classifiers, which use several graph properties in order to determine the chromatic index of cubic graphs,

while using Shapley values in order to estimate the importance of individual graph properties and their values for the machine learning decision process.

We conclude the following:

—The highest overall accuracy of classification (93%) and precision of classification of snarks (92%) was reached with the use of a simple multilayer perceptron neural network containing two hidden layers.

—The highest precision of classification of input graphs into standard cubic graph class (100%) was reached with the use of the Support Vector Machine classifier.

—With the use of Shapley Additive Explanations, we identified a set of graph properties, which were most important in the decision-making process of each classifier. The most interesting graph properties from this point were the girth of the graph, eigenvalues of the adjacency matrix of the graph, the diameter of the graph, and edge connectivity of the graph.

Table 8 contains the time complexity of computation of various properties which reached the highest Shapley values in the described decision-making process and their comparison to the time complexity of standard edge coloring algorithms presented in [Tantau 2008] and [Roditty and Williams 2012]. As can be seen in this table, all used properties can be computed in a lower time complexity compared to the chromatic index of the graph.

Table VIII. Time complexity of graph property computations

Graph property	**Time complexity**
Eigenvalues of graph	$O(V)$
Diameter of graph	$O(V\sqrt{E})$
Girth of graph	$O(VE)$
Edge connectivity of graph	$O(E+k^2Vln(\frac{V}{k})$ for k edges
Chromatic index - naive backtracking	$O(2^{\lvert E(G)\rvert})$
Chromatic index - Beigel & Eppstein	$O(2^{\frac{\lvert V(G)\rvert}{2}})$
Chromatic index - Kowalik	$O(2^{0.427\lvert V(G)\rvert})$

Future work in the research area is inspired by the fact that achieved results are measured on a relatively small sample of data. Therefore, it is necessary to create a similar graph dataset in the size of millions of graphs in the future. In further research, it is necessary to verify the connection between the prediction potential of graph property values and the chromatic index of the graph, and the way this potential behaves with the growing size of graphs.

ACKNOWLEDGEMENT

This research was funded by a grant from the Grant Agency of the Ministry of Education, Science, Research, and Sport of the Slovak Republic (KEGA No. 014UMB-4/2023).

REFERENCES

M. Al-Azawi. 2022. Symmetry-Based Brain Abnormality Detection Using Machine Learning. *Inteligencia Artificial* 24, 68 (2022), 138–150. https://doi.org/10.4114/intartif.vol24iss68pp138-150

Y. Caro, M. Petrusevski, and R. Skrekovski. 2023. Remarks on proper conflict-free colorings of graphs. *Discrete mathematics* 346, 2 (2023). https://doi.org/10.1016/j.disc.2022.113221

G. J. Chaitin. 1982. Register allocation & spilling via graph colouring. In *Proc. 1982 SIGPLAN Symposium on Compiler Construction*. 98–105.

K. Coolsaet, S. D'hondt, and J. Goedgebeur. 2023. House of Graphs 2.0: a database of interesting graphs and more. *Discrete Applied Mathematics* 325 (2023), 97–107. https://houseofgraphs.org

L. L. Custode and G. Iacca. 2023. Evolutionary Learning of Interpretable Decision Trees. *IEEE Access* 11 (2023), 6169–6184. https://doi.org/10.1109/ACCESS.2023.3236260

A. Dudas and B. Modrovicova. 2023. Decision trees in Proper Edge k-coloring of Cubic Graphs. In *Proceedings of 33rd Conference of FRUCT Association*. 21–29.

GraphFilter. 2021. https://github.com/GraphFilter/GraphFilter

I. Guellil, M. Mendoza, and F. Azouaou. 2020. Arabic dialect sentiment analysis with ZERO effort. Case study: Algerian dialect. *Inteligencia Artificial* 23, 65 (2020), 124–135. https://doi.org/10.4114/intartif.vol23iss65pp124-135

L. Kowalik. 2009. Improved edge-coloring with three colors. *Theoretical Computer Science* 410, 38-40 (2009), 3733–3742. https://doi.org/10.1016/j.tcs.2009.05.005

Y. Li, Z. Tang, and J. Yao. 2023. DE_PSO_SVM: An Alternative Wine Classification Method Based on Machine Learning. *Inteligencia Artificial* 26, 71 (2023), 131–141. https://doi.org/10.4114/intartif.vol26iss71pp131-141

D. Marx. 2004. Graph colouring problems and their applications in scheduling. *Periodica Polytechnica. Electrical Engineering* 48, 1-2 (2004), 11–16.

C. Molnar. 2019. *Interpretable Machine Learning*. Published independently. https://christophm.github.io/interpretable-ml-book/

J. Lamas Pineiro and L. Wong Portillo. 2022. Web architecture for URL-based phishing detection based on Random Forest, Classification Trees, and Support Vector Machine. *Inteligencia Artificial* 25, 69 (2022), 107–121. https://doi.org/10.4114/intartif.vol25iss69pp107-121

L. Roditty and V. V. Williams. 2012. Fast approximation algorithms for the diameter and radius of sparse graphs. In *STOC '13: Proceedings of the forty-fifth annual ACM symposium on Theory of Computing*. 515–524. https://doi.org/10.1145/2488608.2488673

SageMath. 2005. https://www.sagemath.org/index.html

T. Tantau. 2008. Complexity of the Undirected Radius and Diameter Problems for Succinctly Represented Graphs. *Technical Report SIIM-TR-A-08-03, Universität zu Lübeck, Lübeck, Germany.*

P. Vashisht and A. Jatain. 2023. A Novel Approach for Diagnosing Neuro-Developmental Disorders using Artificial Intelligence. *Inteligencia Artificial* 26, 71 (2023), 13–24. https://doi.org/10.4114/intartif.vol26iss71pp13-24

V. G. Vizing. 1964. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz.* 3 (1964), 25–30.

Adam Dudáš
Department of Computer Science,
Faculty of Natural Sciences, Matej Bel University
Tajovského 40, 974 01 Banská Bystrica, Slovakia
E-mail: adam.dudas@umb.sk (corresponding author)

Bianka Modrovičová
Department of Computer Science,
Faculty of Natural Sciences, Matej Bel University