IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 10, No. 6, December, 2023, pp. 1994-2002
Journal homepage: http://iieta.org/journals/mmep

Employing Data Mining Techniques for Engineering Soil Classification: A Unified Soil Classification System Approach

Jose Manuel Palomino Ojeda[1*], Lenin Quiñones Huatangari[1], Billy Alexis Cayatopa Calderón[2]

[1] Data Science Research Institute, National University of Jaen, Jaen 06800, Peru
[2] Seismological and Construction Research Institute, National University of Jaen, Jaen 06800, Peru

Corresponding Author Email: jose.palomino@est.unj.edu.pe

https://doi.org/10.18280/mmep.100609

Received: 16 April 2023
Revised: 13 July 2023
Accepted: 10 September 2023
Available online: 21 December 2023

Keywords:
algorithms, artificial intelligence, engineering applications, system soils unified soil classification

ABSTRACT

This research aimed to classify soils using data mining techniques in accordance with the unified soil classification system (USCS). Data were collated via certified soil mechanics studies and laboratory data collection sheets, providing a comprehensive matrix that reflects various soil properties as categorized by the USCS. The USCS system categorizes soils based on physical properties such as particle size and plasticity. The methodology employed was knowledge discovery in databases (KDD), executed in distinct phases. In the selection phase, influential variables for soil type were identified, with gravel, sand, plasticity index, and maximum dry density offering the most significant information. During the processing phase, anomalous and duplicate data were purged using various Python libraries. In the transformation phase, the variables were condensed by proposing five models. Lastly, during the data mining phase, classification algorithms such as k-nearest neighbors, support vector machine, decision trees, and random forest were implemented from the Scikit-learn library 1.1.2. These algorithms achieved accuracy levels of 68.22%, 73.83%, 84.11%, and 69.16% respectively. The decision trees algorithm, combined with the PL_03 model, predicted soil type with a reliability exceeding 84%. The implemented model is expected to assist engineers in making informed and precise decisions.

1. INTRODUCTION

Soil classification systems serve as a common language among engineers, enabling them to categorize soils based on defining characteristics. Such classification is performed through a systematic approach, focusing on specific properties and usage criteria as stipulated by the American Society for Testing and Materials (ASTM). The physical and mechanical properties of the soil dictate its suitability for use as structural fill or foundation soil [1]. A comprehensive understanding of soil classification thus furnishes professionals with insights into its predicted behavior [2].

In the realm of civil engineering, the Unified Soil Classification System (USCS) and the American Association of State Highway and Transportation Officials (AASHTO) approach are commonly employed. Both methodologies hinge on granulometry and Atterberg limits, fundamental elements in soil characterization. Granulometry provides an understanding of the particle size distribution within the soil, thereby identifying the relative proportions of sand, silt, and clay. Conversely, Atterberg limits signify three specific moisture content points at which the soil's behavior alters: the liquid limit, the plastic limit, and the shrinkage limit. These limits inform about the soil's plasticity and water retention capacity [3].

The USCS classification process commences with soil sampling from the study area, after which the samples undergo laboratory testing. Granulometric analysis separates the sample into distinct sizes (sand, silt, and clay), determining the percentage of each fraction. Concurrently, Atterberg limit tests ascertain the liquid, plastic, and shrinkage limits, offering insights into soil plasticity and its response to water content variations. The outcomes of these tests feed into a flow diagram, with soils coded by symbols and letters indicating their characteristics.

Given its pivotal role in the design and execution of infrastructure projects, accurate soil classification holds immense value in civil and geotechnical engineering. Consequently, advancements in classification systems utilizing novel methodologies can significantly enhance the precision and efficiency of geotechnical analysis and design.

In light of this, the current research proposes a novel classification method employing data mining techniques to analyze and extract patterns and relationships within soil property data. This approach enables soil classification in line with the criteria set by the USCS system. Data mining, a technique adept at uncovering patterns and relationships within large datasets, is particularly beneficial when analyzing and categorizing numerous soil samples with various characteristics. The applied data mining algorithms, namely support vector machine, random forest, k-nearest neighbors, and decision trees, identify intricate relationships between the

physical and mechanical properties of soils.

The study utilizes artificial intelligence (AI) techniques centered on data mining. It applies soil classification algorithms for engineering purposes to enhance and automate the soil classification process by integrating data mining with the USCS. This approach expedites the determination of soil categories based on their properties, offering a more efficient method. Such advancements bear significant practical implications for civil and geotechnical engineering, where accurate soil classification is essential for effective decision-making in construction and infrastructure projects. Additionally, this methodology may reveal patterns and relationships that could potentially remain undetected in manual analysis, thereby enriching our understanding of soil characteristics and behavior across varying conditions and locations.

2. LITERATURE REVIEW

The intersection of data mining and expert systems has been explored extensively in the literature, with various authors employing these techniques to address soil classification and related concerns. Bui et al. [4], for instance, utilized data mining to construct models of soil property distribution in mainland Australia. A linear decision tree model was developed, integrating 19 variables derived from the area's edaphoclimatic traits. Arce [1] presented an alternative approach with the creation of software for soil classification, integrating physical-mechanical soil property data. The methodology underpinning the software included the identification of static equations and dynamic variables for each method, implemented based on algorithms established by the SUCS and AASHTHO systems.

Data mining algorithms have been harnessed to address a variety of engineering problems, as evidenced by the works of several authors. Javadi and Rezania [5] implemented neural networks and data mining to model soil behavior. In contrast, Palomino Ojeda and Rosario Bocanegra [6] applied data mining techniques to estimate seismic vulnerability. Hernández Pereira and Medina González [7] employed data assimilation techniques to estimate soil moisture, while Cortés Henao [8] leveraged data mining methods for predictive maintenance of drinking water distribution networks.

Over the past few years, the development of computational techniques for pattern recognition and data mining has gained momentum, driven by their potential to model diverse engineering problems. A core premise of pattern recognition systems is their adaptive learning capability, which allows them to generate predictions for new scenarios [5].

The field of data mining, which revolves around the unveiling of patterns and relationships within large databases, has its roots in artificial intelligence, machine learning, and pattern recognition literature. It is often associated with the concept of knowledge discovery in databases [4]. Additionally, data mining is perceived as the process of extracting knowledge from large datasets via algorithms that identify patterns integral to informed decision-making [9].

3. MATERIALS AND METHODS

3.1 Soil classification

Soil classification was performed according to ASTM standards to determine physical and mechanical properties. A representative sample of the soil was taken and air-dried to eliminate surface water content. Then the granulometric test (ASTM D 422) was performed using sieves of different sizes, weighing the fractions retained in each sieve. The results were expressed as cumulative percentages of retained weight as a function of particle size. The Atterberg liquid and plastic limit test (ASTM D4318) was then performed. For the liquid limit, a sample is taken that passes the 40 mesh sieve, and the soil is kneaded with water until a homogeneous paste is obtained, using the Casagrande cup, the sample is molded in the Cascador with the spatula and a groove of 2 mm is made along the sample, then it is rotated and the number of blows is noted until it closes at 13 mm, three repetitions are made for a different number of blows, at the end three samples are taken to determine the moisture content (ASTM D 2216). The plastic limit is carried out by taking a wet soil sample in the form of an ellipsoid and then rolling it with the fingers of the hand on a smooth surface with the pressure strictly necessary to form cylinders. If the cylinder has not crumbled before it reaches a diameter of about 3.2 mm (1/8"), it is turned back into an ellipsoid, and the process is repeated as many times as necessary until it crumbles to about that diameter. The portion obtained is placed on watch glasses until about 6g of soil is collected to determine its moisture content (ASTM D 2216).

3.2 Algorithms

Data mining is a set of tools designed for rapid, automated, and exploratory data analysis. These tools include decision trees, rule induction, inductive logic programming, k-nearest neighbors, clustering algorithms, neural networks, genetic algorithms, and Bayesian networks, among others [4].

The algorithms used in the research are k-nearest neighbors, support vector machine, decision trees, and random forest.

K-nearest neighbors (KNN) is a nonparametric classifier that predicts the label of an entry by identifying its k nearest neighbors based on specified distance metrics, such as Euclidean distance or cosine similarity. It then determines the majority vote of the neighboring labels to assign the entry to a particular class [10]. It is often used for its simplicity and ability to obtain faster results, unlike other methods by storing a set of prototypes representing the knowledge of the problem [5]. Based on the idea that patterns closer to a target pattern x', provide more information about the label. KNN assigns the class label of most patterns closest to K the data space. This metric is used to calculate the distance or similarity between data points. Choosing an appropriate distance metric is critical because it determines how close or similar two data points are in the feature space. In this R^q, it is reasonable to employ the Minkowski metric (p-norm) corresponding to the Euclidean distance p=2. In other data spaces, we have to choose suitable distance functions, e.g., the Hamming distance in B^q. In the case of binary classification, the label set y={1, -1} is used, and the KNN is defined as with neighborhood size K and with the index set $N_K(x')$ of the K nearest patterns see Eq. (1) and Eq. (2). It is recommended to employ a value K=5 in the models because it produces the best performance with a low standard deviation between K=1, 2, ..., 20 [11].

$$\left\|x' - x_j\right\|^p = \left(\sum_{i=1}^{q} \left|(x_i)' - (x_i)_j\right|^p\right)^{1/p} \quad (1)$$

$$f_{KNN}(X') = \begin{cases} 1 \text{ if } \sum i \in N_K(x')^{yi} \geq 0 \\ -1 \text{ if } \sum i \in N_K(x')^{yi} < 0 \end{cases} \quad (2)$$

Support vector machine (SVM) is a method used in various fields such as object classification, face recognition, and text categorization. SVM aims to find an optimal hyperplane that best separates data points of different classes in the feature space [12].

The SVM training procedure for pattern recognition involves solving a quadratic optimization problem. The main goal of SVM is to find the optimal hyperplane that maximizes the distance between data points of different classes $\{(x_1, y_1), \ldots, (x_i, y_i)\}$, where $x \in R^N, y \in \{-1, 1\}$. Kernel mapping is a technique used to identify training data from the input space to a higher dimensional feature space where the mapped training data can become linearly separable. This is particularly useful when working with data that is not linearly separable in the original feature space.

Maximize:

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{i=1}^{l} \alpha_i y_i \alpha_j y_j K(X_i . X_j) \quad (3)$$

subject to $\sum_{i=1}^{l} \alpha_j y_j = 0$.

The decision function becomes:

$$f(x) = sng\left(\sum_{i=1}^{l} \alpha_j y_j K(X_i, X) + b\right) \quad (4)$$

Decision trees are classification and prediction techniques that can handle both numerical and nominal data. They are constructed in two stages: Induction and Pruning. During the induction phase, the tree is built by recursively partitioning the data according to features and selecting the best partitioning criterion at each node. On the other hand, in the pruning phase, excessive complexity is removed from the tree to improve its generalizability. The most commonly used decision tree algorithms are C4.5, ID3, CART, and J48 [13].

Random forest is a robust learning algorithm that uses multiple randomized decision trees and combines their predictions by averaging. Its operation consists of two phases: construction, where multiple decision trees are built using different training instances to ensure an accurate and less overfit model, and prediction, which is determined by averaging the results of each tree. Performance is evaluated based on several criteria, variations in classifier parameter values, sensitivity to noise, and changes in training size. To measure efficiency, they are compared to classification trees by analyzing the performance relative to classification trees [14].

3.3 Data matrix

Soil classification information according to SUCS was collected using a data collection sheet, from different laboratories certified by the National Quality Institute (INACAL) and certified soil mechanics files. A total of 474 soil classifications were obtained from 2018 to 2022, forming a data matrix of 474 instances and 9 variables. The variables collected were: gravel (GV), sand (AR), fines (FN), liquid limit (LL), plastic limit (LP), plasticity index (PI), maximum dry density (MDS), optimum moisture content (OCH), and soil type extracted from granulometry (ASTM D 422), Atterberg Limits (ASTM D4318), and Proctor (ASTM D1557) tests, see Figure 1 and Tables 1-2. For repeatability, each test was repeated three times until the results obtained varied within ±0.5% in the laboratory [15].

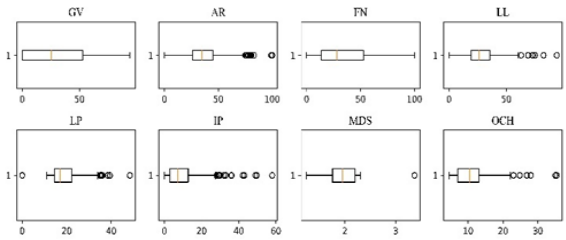

Figure 1. Descriptive statistics of the variables

Table 1. Description of variables collected

Variable	Description	Unit	Type
Gravel	It is the percentage of gravel contained in the soil, obtained by sieving grain size analysis (ASTM D 422).	%	Discreet
Sand	The percentage of sand contained in the soil is obtained by a sieve analysis (ASTM D 422).	%	Discreet
Fines	The percentage of fines contained in the soil is obtained by sieve particle size analysis (ASTM D 422).	%	Discreet
Liquid limit	It is the moisture content in percentage, which defines the boundary between the semi-liquid and plastic consistency states (ASTM D4318).	%	Discreet
Plastic limit	Moisture content in percent, which defines the boundary between the plastic and semi-solid consistency states (ASTM D4318).	%	Discreet
Plasticity index	The moisture content range over which the soil behaves plastically is obtained by the difference between LL and LP (ASTM D4318).	%	Discreet
Maximum dry density	The highest density that soil can reach when compacted to optimum moisture (ASTM D1557).	g/cm^3	Discreet
Optimum moisture content	It is the percentage of moisture necessary to obtain the maximum density of the soil (ASTM D1557).	%	Discreet
Soil type	It is the symbol assigned to a certain soil so that it can be interpreted by everyone.	-	Nominal

Table 2. Statistical data from the database

Variable	Quantity	Mean	Standard Deviation	Minimum	25%	50%	75%	Maximum
Gravel	474	29.76	25.16	0	2.00	27.51	54.30	94.00
Sand	474	36.86	16.23	0	26.93	34.96	45.98	100.00
Fines	474	33.25	24.15	0	13.15	26.40	49.00	100.00
Liquid limit	474	25.99	12.48	0	19.00	25.70	35.00	60.00
Plastic limit	474	16.51	8.84	0	14.05	17.00	22.00	48.40
Plasticity index	474	8.22	6.75	0	3.00	7.00	11.57	29.40
Maximum dry density	474	1.97	0.24	1.32	1.78	1.97	2.20	3.37
Optimum moisture content	474	10.43	4.00	4.50	6.90	10.37	13.00	23.00
Soil type	474	-	-	-	-	-	-	-

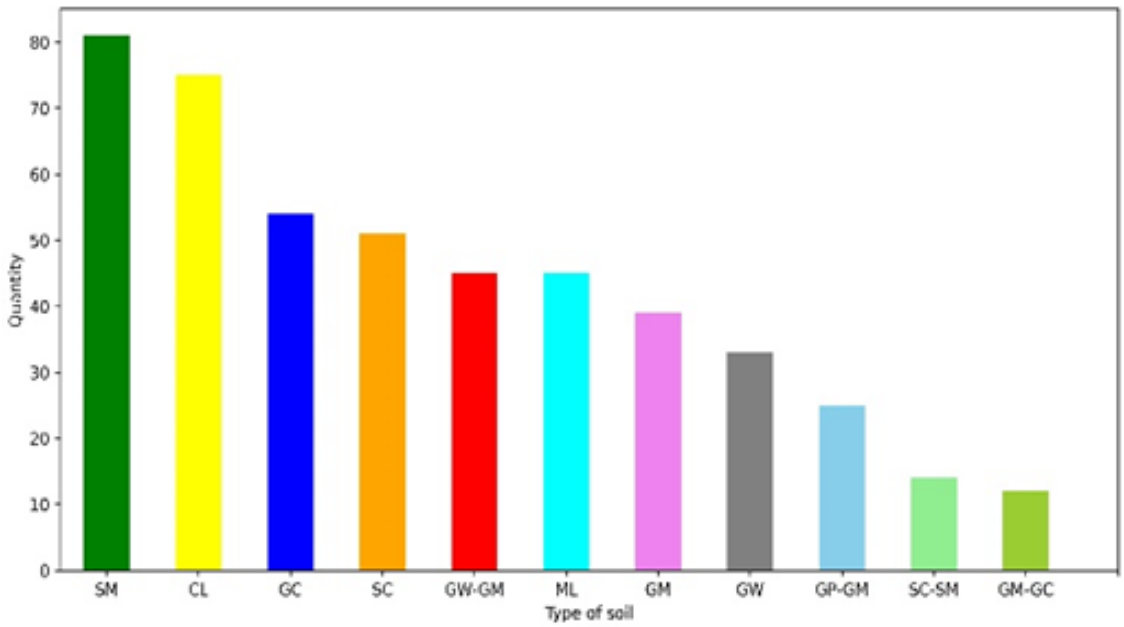

Figure 2. Soil types contained in the database

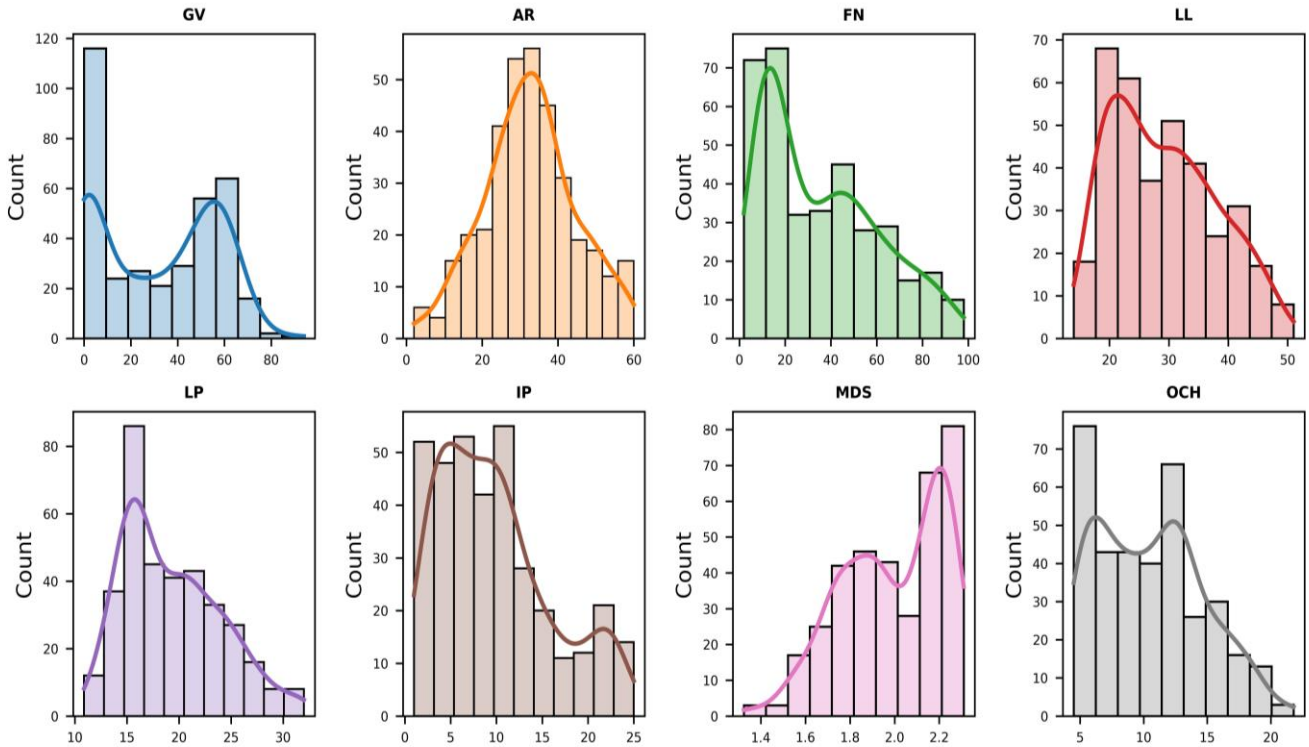

Figure 3. Types of distribution of each variable

The soil classification present in the data matrix contains 11 soil types according to SUCS: silty sand (SM), clay (CL), clayey gravel (GC), clayey sand (SC), gravel graded with silt (GW-GM), silt (ML), silty gravel (GM), well-graded gravel (GW), poorly graded gravel with silt (GP-GM), clayey sand with silt (SC-SM), and silty gravel with clay (GM-GC), see Figure 2.

3.4 Methodology for employing data mining

The knowledge discovery database (KDD) methodology was used, which is known for its iterative nature, where certain phases can lead to revisiting previous steps. In many cases, multiple iterations are required to efficiently extract high-quality, interactive knowledge. The involvement of subject matter experts is crucial throughout the process, as they contribute to data preparation and validate the extracted knowledge [16].

The KDD methodology uses a set of techniques, collectively referred to as data mining, to discover trends in large amounts of data [17]. The term KDD was coined in 1989 to convey the idea that knowledge is the ultimate result of data-driven discovery. The KDD process consists of extracting patterns, in the form of rules or functions, from data for

analysis by the user [18].

The phases involved in the KDD process are:

3.4.1 Selection

Identified the need to classify soils for engineering purposes through data mining.

A data set was created, on which the discovery process was carried out. For the selection of the attributes, the distribution of each variable was analyzed, see Figure 3.

Then the Sklearn library was used in the Jupyter Lab interface and the "RandomForest" algorithm was run, which evaluates the value of the variables by measuring the gain concerning the output variable (soil type) using the command "model.feature_importances_". The variables with a Ranked value greater than 8.0 were selected.

3.4.2 Preprocessing and cleaning

During data collection, it is common for the obtained set to contain null or anomalous instances, which can cause noise in the knowledge extraction process. In this phase, data cleaning was applied to improve the quality of the data using the Python programming language with the Jupyter Lab interface and the Numpy, pandas, and seaborn libraries, which use statistical analysis as a whisker box to clean and remove anomalous data. Missing data were imputed using three different methods: average, median, and regression algorithms. In addition, duplicate data were removed using the Set command in Python. To deal with anomalous data, the Panda's library was used, and the boxplot was used, which allows visualization of the distribution and detection of outliers by identifying points that fall outside the boundaries of the graph (see Figure 4). In addition, the library assigned visual attributes to the data, calculated statistical transformations, and decorated the graph with informative labels on the axes [19].

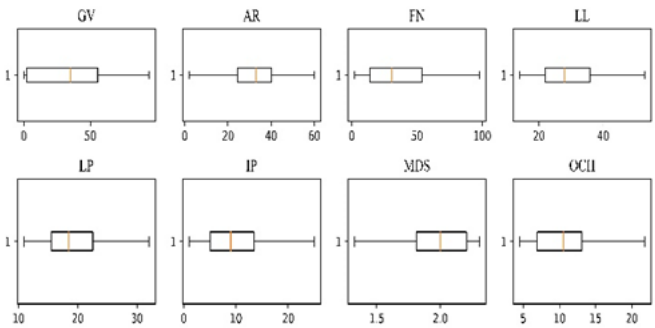

Figure 4. Database without outliers

3.4.3 Transformation and reduction

Useful characteristics were sought to represent the data about soil type, using transformation methods to reduce the effective number of variables and invariant representations of the data (see Table 3); the variables were grouped according to the type of distribution in Figure 3.

3.4.4 Data mining

We used the Python programming language with the Jupyter Lab interface and the Scikit-learn 1.1.2 package developed for machine learning with Python. Scikit-learn offers a wide range of machine learning algorithms, including supervised and unsupervised techniques, all presented through a consistent, task-oriented interface. This design facilitates a direct comparison of methods for a specific application, ensuring ease of use and flexibility in the machine-learning process [20]. The prediction algorithms used for training and pattern extraction were k-nearest neighbors, support vector machine, decision trees, and random forest, which consist of three parts: knowledge representation, evaluation, and search, maintaining minimal bias without affecting variance; adapting to data types.

To build the models, the data were divided into training (70%) and test (30%) sets, and then the hyperparameters were configured as shown in the following Table 4.

Table 3. Grouping of variables according to their importance

Group	Variables
PL_1	• Gravel • Sand • Plasticity index • Maximum dry density
PL_2	• Fine • Plasticity index • Maximum dry density • Optimum moisture content
PL_3	• Gravel • Fine • Plastic limit • Plasticity index
PL_4	• Gravel • Fine • Maximum dry density • Optimum moisture content
PL_5	• Gravel • Sand • Liquid limit • Maximum dry density

Table 4. Hyperparameters of the algorithms

Algorithms	Hyperparameters
Support vector machine	C=1.1 Kernel=rbf (radial basis function) Gamma=0.1
Random forest	n_estimators=120 max_depth=15 max_features=sqrt (square root of the total number of characteristics)
K-nearest neighbors	n_neighbors=5 weights=uniform
Decision trees	max_depth=None criterion=gini (Gini index) min_samples_split=2

Table 5. Confusion matrix

Actual Class	Predicted Class	
	Positive	Negative
Positive	True positives (VP)	False negatives (FN)
Negative	False positives (FP)	True negatives (VN)

3.4.5 Intervention and evaluation of data

After training the five groups PL_1, PL_2, PL_3, PL_4, and PL_5 with the collected database and the proposed data mining models, the predicted results were compared with the actual values using performance metrics such as the confusion matrix, also known as the error matrix. This has a special table design that allows visualizing the performance of the algorithm, where each row represents the instances of an actual class, while each column represents the instances of a predicted class.

This design allows a clear evaluation of the performance of the algorithm in correctly classifying the data points into different classes. The analysis of the confusion matrix provides insight into the accuracy, precision, recall, and other performance parameters of the data mining models [21]. The distribution of the errors committed by the algorithms is expressed by true positives (VP), false negatives (FN), false positives (FP), and true negatives (VN), see Table 5.

Three main metrics were used to evaluate the performance of each algorithm: Recall, Specificity, and Precision. Recall measures the proportion of positive cases correctly identified, Specificity measures the proportion of negative cases correctly identified, and Precision measures the closeness of measured values to established or known values. By using these measures. These metrics allow a better understanding of the capabilities of the algorithms and facilitate informed decision-making in their practical application [22], see Eqs. (5)-(8).

$$Recall\ (R)=\frac{VP}{VP+FN} \quad (5)$$

$$Specificity\ (E)=\frac{VN}{VN+FP} \quad (6)$$

$$\text{Precision}\ (P)=\frac{VP}{VP+FP} \quad (7)$$

$$\text{F-measure}=2*\frac{R+P}{R*P} \quad (8)$$

The various metrics used to evaluate the algorithms were: Recall is the ability of the algorithm to accurately predict the correct results, especially in identifying positive cases. Specificity is the ability of the algorithm to correctly identify negative results, thereby reducing false positives. Precision quantifies the number of correct positive results relative to the total number of positive instances in the population, providing a measure of accuracy in identifying positive instances. The F-measure indicates the overall goodness of fit of the algorithm, combining the precision and recall measures in a balanced manner.

4. RESULTS

The data matrix consisted of 474 soil classifications and 9 variables, compiled from certified laboratories and soil mechanics studies, which were extracted from technical files of public works in Peru. After the selection stage, the relationship between the independent and dependent variables (soil type) was evaluated, see Figure 5.

In the selection stage with the Random Forest algorithm, the importance of the variables concerning the output variable (soil type) was evaluated, obtaining the following results see Table 6.

Table 6. Ranking of variables according to their importance

Variables	Symbol	Ranked
Fine	FN	21.23
Plasticity index	IP	15.60
Gravel	GV	12.56
Optimum moisture content	OCH	11.44
Maximum dry density	MDS	11.25
Sand	AR	9.66
Liquid limit	LL	9.46
Plastic limit	LP	8.80

Table 7. Evaluation of groups during soil type prediction

Group	Well-Classified Instances (%)	Misclassified Instances (%)	F-Measure (%)
PL_1	68.22	31.78	66.99
PL_2	71.03	28.97	69.63
PL_3	84.11	15.89	83.37
PL_4	63.51	36.49	58.83
PL_5	61.68	38.32	56.15

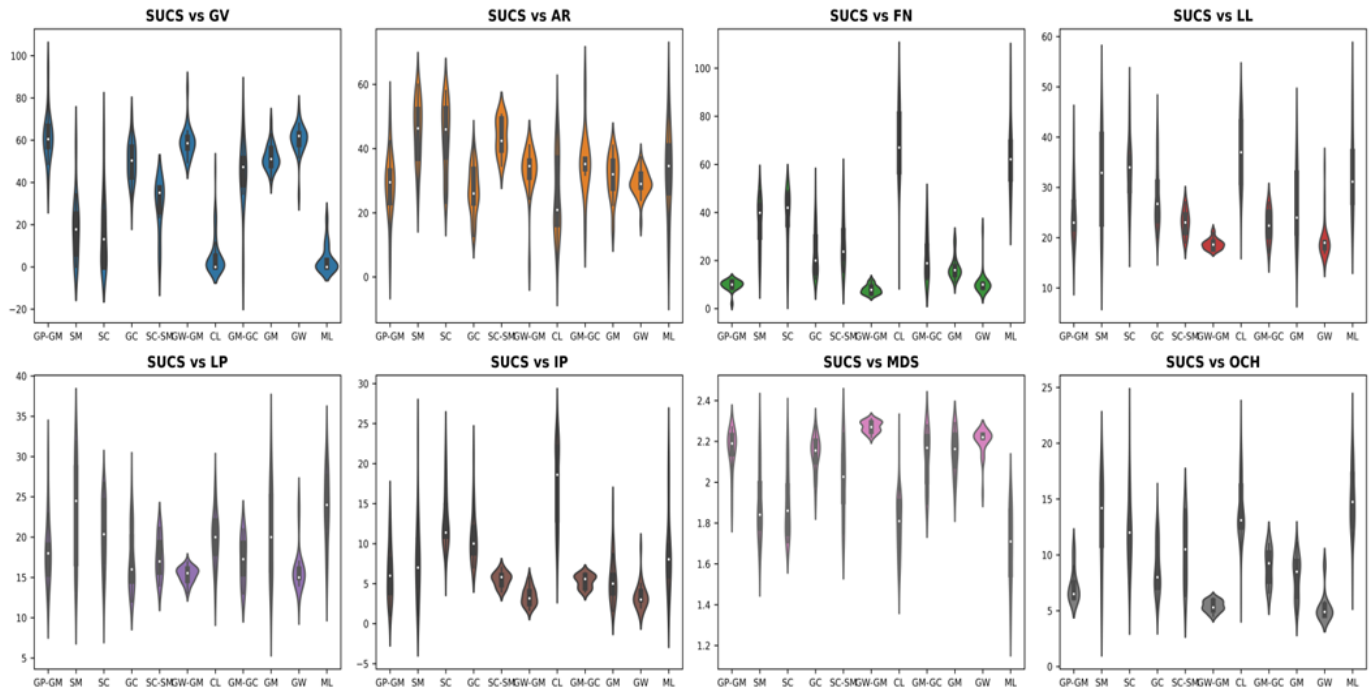

Figure 5. Relationship between soil type and the variables collected

In the preprocessing and cleaning stage, outliers were eliminated using the Python programming language and the Numpy, pandas, and seaborn libraries, obtaining the following high-quality database, see Figure 4.

In the construction phase of the data mining models, the data were divided into training (70%) and testing (30%), using k-fold cross-validation, configured with the following hyperparameters for the support vector machine algorithm, a radial kernel was used, and the regularization parameter C was set to a value of 1.1. For Random Forest, a forest of 120 decision trees was constructed and the maximum depth of each tree was limited to 15 to avoid overfitting. The maximum number of features considered when searching for the best split at each node was set to the square root of the total number of features. For the K-Nearest Neighbors algorithm, 5 nearest neighbors were considered and a uniform weight approach was applied, and for decision trees, trees were allowed to grow without a depth limit and the Gini index was used as a criterion to measure the quality of splits at nodes. Node splitting was performed when there were at least 2 samples at each node.

After programming the code in Jupiter Lab with Python, the database was imported and configured according to the hyperparameters. The results of the soil type prediction, with the different trained and validated algorithms of the Scikit-learn 1.1.2 library are presented in Figure 6 and Table 6, where groups PL_1, PL_2, and PL_3, present the lowest error in soil type classification.

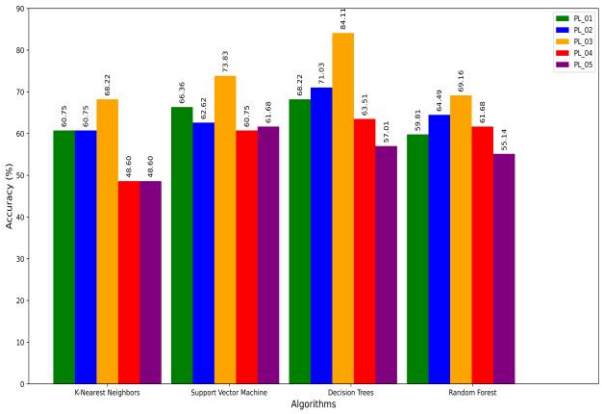

Figure 6. Accuracy of algorithms for each type of group

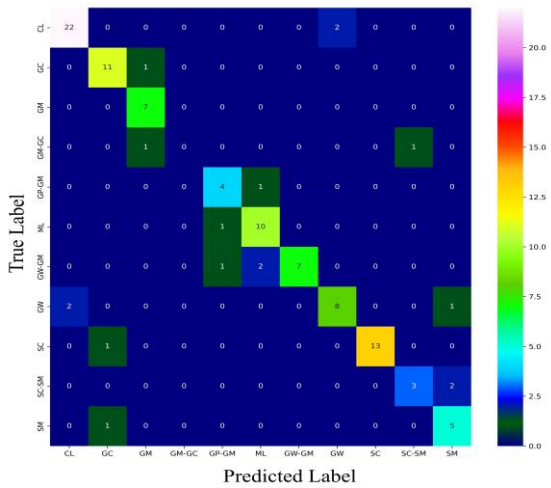

Figure 7. Confusion matrix of PL_3

Table 7 shows the results of the evaluation of the soil type prediction for the different groups by the percentages of correctly and incorrectly classified instances and the F-measure for each group. In group PL_1, 68.22% of the instances were correctly classified, while 31.78% were incorrectly classified. The F-measure, which is the harmonic mean of accuracy and recall, is 66.99%. In the PL_2 group, 71.03% of the instances were correctly classified, while 28.97% were misclassified. The F-measure of this group was 69.63%. The PL_3 group showed a higher accuracy, with 84.11% of the instances correctly classified and only 15.89% misclassified. The F-measure was a remarkable 83.37%. In contrast, the PL_4 group had a lower rate of well-classified instances with 63.51% and 36.49% misclassified instances. The F-measure for this group was 58.83%. Similarly, the PL_5 group also showed moderate accuracy, with 61.68% of instances correctly classified and 38.32% misclassified. The F-measure for this group was 56.15%.

```
|--- GV <= 32.45
|    |--- FN <= 50.01
|    |    |--- IP <= 8.20
|    |    |    |--- class: SM
|    |    |--- IP >  8.20
|    |    |    |--- LP <= 26.10
|    |    |    |    |--- class: SC
|    |    |    |--- LP >  26.10
|    |    |    |    |--- class: SM
|    |--- FN >  50.01
|    |    |--- IP <= 9.85
|    |    |    |--- class: ML
|    |    |--- IP >  9.85
|    |    |    |--- LP <= 27.00
|    |    |    |    |--- class: CL
|    |    |    |--- LP >  27.00
|    |    |    |    |--- class: ML
|--- GV >  32.45
|    |--- FN <= 12.08
|    |    |--- LP <= 16.68
|    |    |    |--- IP <= 6.38
|    |    |    |    |--- FN <= 7.97
|    |    |    |    |    |--- class: GW-GM
|    |    |    |    |--- FN >  7.97
|    |    |    |    |    |--- class: GW
|    |    |    |--- IP >  6.38
|    |    |    |    |--- class: GP-GM
|    |    |--- LP >  16.68
|    |    |    |--- class: GP-GM
|    |--- FN >  12.08
|    |    |--- IP <= 6.66
|    |    |    |--- GV <= 40.55
|    |    |    |    |--- class: SC-SM
|    |    |    |--- GV >  40.55
|    |    |    |    |--- class: GM
|    |    |--- IP >  6.66
|    |    |    |--- LP <= 23.80
|    |    |    |    |--- class: GC
|    |    |    |--- LP >  23.80
|    |    |    |    |--- class: GM
```

Figure 8. The algorithm that predicts soil type for engineering purposes

The F-measure, a balance between accuracy and recall, provides a comprehensive assessment of the predictive performance of each group. The results in Table 6 highlight the different levels of accuracy achieved in predicting soil types for the different groups, with group PL_3 showing the highest overall performance both in terms of well-classified instances and F-measure.

The PL_3 group with the decision trees algorithm is the one that best classifies the soil type for engineering purposes with 84.11 % accuracy. The results are shown in the confusion matrix where the predicted vs. actual classes were compared see Figure 7.

The algorithms with the best performance were decision trees, and support vector machine, presenting higher accuracy in the five proposed groups, unlike the K-Nearest Neighbors algorithm which presents low accuracy in soil type classification see Table 8.

Group PL_03 with the Decision Tree algorithm best predicts the soil type for engineering purposes with 84.11% accuracy. Figure 8 details the decision tree generated from the algorithm.

Table 8. Evaluation of algorithms during soil type prediction

Algorithms	Well-Classified Instances (%)	Misclassified Instances (%)	F-Measure (%)
K-nearest neighbors	68.22	31.78	66.99
Support vector machine	73.83	26.17	69.40
Decision trees	84.11	15.89	83.37
Random forest	69.16	36.49	58.83

5. DISCUSSION

The data matrix consisted of 474 soil classifications and 9 variables collected from 2018 to 2022 from certified laboratories and mechanical studies, based on tests standardized by the Ministry of Transport and Communications (MTC) and the Peruvian Technical Standard (NTP). The variables collected were of a discrete and nominal type, similar to Bui et al. [4], who use discrete variables in their research.

The algorithms chosen to estimate the engineering soil type were k-nearest neighbors, support vector machines, decision trees, and random forests recommended by Al-Shamiri [9]. Al-Shamiri conducted research on pattern extraction from databases using various algorithms, such as decision tree algorithms, k-nearest neighbors, neural networks, naive Bayes, support vector machines, random forests, regression, AIS, SETM, Apriori, FP-Growth, and K-Means and found that these algorithms performed best during the training and validation phase. Like Rao and Chaparala [23], they used a support vector machine to construct decision tables that minimize features extracted from large data sets.

The modeling of soil types for engineering purposes was performed with 5 groups (PL_1, PL_2, PL_3, PL_4, and PL_5) conformed by 4 different variables and trained with four algorithms from the Scikit-learn library in Jupyter Lab with the Python programming language, obtaining a percentage of correct classification of 68. 22%, 71. 03%, 84.11%, 63.51%, and 61.68%, respectively, which exceed by 84.11% the results obtained by Manjula and Narsimha [24], of 75.39% and 75.38% of correctly classified instances, and are in the range of the results obtained by Palomino Ojeda and Rosario Bocanegra [6], Hernández Pereira and Medina González [7] and Cortés Henao [8], guaranteeing the validity of the models.

Promising results have been obtained in soil classification by data mining. However, it is important to consider some limitations. To improve the generalization and robustness of the models, it is necessary to collect other variables that represent all the conditions and types of soils present in the region studied, such as chemical compounds and electrical resistivity, among others. This will allow for obtaining a broad and diverse database. In future research, it is recommended to vary the hyperparameters of the models. Although hyperparameter optimization was performed, the entire search space may not have been exhaustively explored, and advanced optimization techniques, such as Bayesian optimization, could be used to find more optimal hyperparameter combinations and further improve model performance.

6. CONCLUSIONS

Data mining is related to several fields, the most important being artificial intelligence (AI), databases, mathematical modeling, machine learning, and management science.

The soil type for engineering purposes was determined by a data mining algorithm that created 5 groups; for this purpose, the KDD methodology and the algorithms k-nearest neighbors, support vector machine, decision trees, and random forest were used, obtaining the highest accuracy in PL_3 of 84.11% with the decision trees algorithm.

The models made it possible to determine soil type for engineering purposes by applying them to characterization studies in other geographic areas. The methodology is adaptable and can be extended to classify soils in other regions, provided quality and representative data are available.

The application of data mining techniques to SUCS soil classification has valuable practical implications in the fields of engineering and geotechnics. The developed models are useful tools to improve decision making, reduce risks in construction projects and soil analysis. As research and development of new methodologies continue, opportunities for better understanding and use of soils in different geotechnical applications and contexts will open up.

These algorithms can be used by engineering firms to improve decision-making in infrastructure and geotechnical projects. By having an accurate soil classification model, they could reduce the risks and costs associated with the design and construction of structures, roads, and foundations.

The study provides a solid basis for soil classification using data mining techniques according to SUCS. However, there are limitations that need to be addressed in future research. Obtaining new variables, optimizing hyperparameters, comparing systems and algorithms, and validating in different geotechnical contexts can significantly contribute to the advancement and improvement of soil classification and its practical applicability in geotechnical engineering.

REFERENCES

[1] Arce, W. (2021). Diseño de software de clasificación de suelo programado en la plataforma. net de Visual Studio bajo las normas de los sistemas SUCS y AASTHO. Revista Tierra, 1(1).

[2] Al Rawas, A. (1998). Soil classification decision support system using an expert system approach.

[3] Guerrero, C.C., Cruz Velasco, L.G. (2018). Estudio experimental de clasificación de suelos derivados de cenizas volcánicas en el suroccidente colombiano con el método SUCS, el AASHTO y un nuevo método de clasificación de suelos. Ingeniería y Desarrollo, 36(2):

378-397. https://doi.org/10.14482/inde.36.2.10377
[4] Bui, E.N., Henderson, B.L., Viergever, K. (2006). Knowledge discovery from models of soil properties developed through data mining. Ecological Modelling, 191(3-4): 431-446. https://doi.org/10.1016/j.ecolmodel.2005.05.021
[5] Javadi, A.A., Rezania, M. (2009). Applications of artificial intelligence and data mining techniques in soil modeling. Geomechanics and Engineering, 1(1): 53-74. https://doi.org/10.12989/gae.2009.1.1.053
[6] Palomino Ojeda, J.M., Rosario Bocanegra, S. (2021). Estimación de la vulnerabilidad sísmica en viviendas de albañilería confinada, mediante técnicas de minería de datos, en el sector Pueblo Libre, Jaén-2020. Universidad Nacional de Jaén.
[7] Hernández Pereira, Y., Medina González, H. (2012). Estimación de la humedad del suelo mediante técnicas de asimilación de datos. Revista Ciencias Técnicas Agropecuarias, 21(4): 30-35.
[8] Cortés Henao, M. (2015). Minería de datos para el mantenimiento predictivo de redes de distribución de agua potable. Universidad de Los Andes.
[9] Al-Shamiri, A.Y.R. (2021). Artificial intelligence and pattern recognition using data mining algorithms. International Journal of Computer Science & Network Security (IJCSNS), 21(7): 221-232. https://doi.org/10.22937/IJCSNS.2021.21.7.26
[10] Sitawarin, C., Wagner, D. (2019). On the robustness of deep K-Nearest Neighbors. In 2019 IEEE Security and Privacy Workshops (SPW), Francisco, CA, USA, IEEE, 1-7. https://doi.org/10.1109/SPW.2019.00014
[11] Xu, Y.M., Klabjan, D. (2018). K-Nearest Neighbors by means of sequence to sequence deep neural networks and memory networks. arXiv preprint arXiv: 1804.11214. https://doi.org/10.48550/arXiv.1804.11214
[12] Zheng, S., Ding, C. (2018). Minimal support vector machine. arXiv preprint arXiv: 1804.02370. https://doi.org/10.48550/arXiv.1804.02370
[13] Han, J., Kamber, M., Pei, J. (2012). Front matter. Data Mining (Third Edition), The Morgan Kaufmann Series in Data Management Systems, i-v. https://doi.org/10.1016/B978-0-12-381479-1.00016-2
[14] Zhou, Y.M., Qiu, G.P. (2018). Random forest for label ranking. Expert Systems with Applications, 112: 99-109. https://doi.org/10.1016/j.eswa.2018.06.036
[15] Palomino Ojeda, J.M., Cayatopa Calderon, B.A., Quiñones Huatangari, L., Rojas Pintado, W. (2023). Determination of the California bearing ratio of the subgrade and granular base using artificial neural networks. International Journal of Engineering & Technology Innovation, 13(3): 175-188. https://doi.org/10.46604/ijeti.2023.11053
[16] Mariscal, G., Marban, Ó., Fernández, C. (2010). A survey of data mining and knowledge discovery process models and methodologies. The Knowledge Engineering Review, 25(2): 137-166. https://doi.org/10.1017/S0269888910000032
[17] Maniatty, W.A., Zaki, M.J. (2000). A requirements analysis for parallel kdd systems. In Parallel and Distributed Processing, Springer, Berlin, Heidelberg, 358-365. https://doi.org/10.1007/3-540-45591-4_47
[18] Sarwar, A.M., Shaiban, S.M., Biswas, S., Promiti, A.S., Faysal, T.I., Bazlul, L., Hossain, M.S., Rahman, R.M. (2020). Soil analysis and unconfined compression test study using data mining techniques. In Advances in Computational Collective Intelligence: 12th International Conference, Springer International Publishing, pp. 38-48. https://doi.org/10.1007/978-3-030-63119-2_4
[19] Bisong, E. (2019). Matplotlib and seaborn. Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners, 151-165. https://doi.org/10.1007/978-1-4842-4470-8_12
[20] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É. (2011). Scikit-learn: machine learning in python. Journal of Machine Learning Research, 12: 2825-2830.
[21] Corso, C.L. (2009). Aplicación de algoritmos de clasificación supervisada usando Weka. Córdoba: Universidad Tecnológica Nacional, Facultad Regional Córdoba.
[22] Yusro, M., Suryana, E., Ramli, K., Sudiana, D., Hou, K.M. (2019). Testing the performance of a single pole detection algorithm using the confusion matrix model. In Journal of Physics: Conference Series, IOP Publishing, 1402(7): 077066. https://doi.org/10.1088/1742-6596/1402/7/077066
[23] Rao, M.V.V., Chaparala, A. (2021). An efficient data mining technique for structural strength monitoring system. Ingénierie des Systèmes d'Information, 26(2): 237-243. https://doi.org/10.18280/isi.260211
[24] Manjula, A., Narsimha, G. (2018). Using an efficient optimal classifier for soil classification in spatial data mining over big data. Journal of Intelligent Systems, 29(1): 172-188. https://doi.org/10.1515/jisys-2017-0209