This is an Author Accepted Manuscript version of the following chapter: de Arriba-Pérez, F., García-Méndez, S., Carou, D., Medina-Sánchez, G, Optimization of the Turning Process by Means of Machine Learning Using Published Data, published in Notes for Manufacturing Instructors. Materials Forming, Machining and Tribology, edited by Carou, D., Davim, J.P., 2024, Springer, Cham reproduced with permission of Springer, Cham. The final authenticated version is available online at: https://doi.org/10.1007/978-3-031-48468-1_13.

Optimization of the turning process by means of Machine Learning using published data

Francisco de Arriba-Pérez[1*], Silvia García-Méndez[1], Diego Carou[2] and Gustavo Medina-Sánchez[3]

[1]Information Technologies Group, atlanTTic, University of Vigo, Vigo, Spain.

[2]Universidade de Vigo, Departamento de Deseño na Enxeñaría, Ourense, Spain.

[3]Mechanical and Mining Engineering Department, University of Jaén, Jaén, Spain.

*Corresponding author(s). E-mail(s): farriba@gti.uvigo.es - 0000-0002-1140-679X;

Contributing authors: sgarcia@gti.uvigo.es - 0000-0003-0533-1303; diecapor@uvigo.gal - 0000-0002-6395-304X; gmedina@ujaen.es - 0000-0003-4925-2818;

Abstract

Machining parameters play a critical role in the results of the turning process: cutting forces, dimensional accuracy, surface roughness, tool wear, etc. Manufacturers offer recommendations for their tools, but the complex relations between machining parameters make the process optimization process not straightforward. Researchers usually opt for performing experimental studies to optimize specific or multiple outputs of these processes. However, this approach is costly and time-consuming. Thus, in the present chapter, we show a methodology to use Machine Learning, taking advantage of the vast amounts of published data in the literature. Particularly, the chapter aims to study the surface roughness attained in turning the Ti6Al4V alloy.

Keywords: Machine Learning, surface roughness, Ti6Al4V, titanium alloys, turning

1 Introduction

Turning is one of the main subtractive manufacturing processes used in industry. Despite the availability of other options, turning usually starts with cylindrical bars. Several metallic materials, such as aluminum, steel, and titanium, can be turned. The process is of high complexity, and the cutting mechanism involves high temperatures and mechanical loads. Accordingly, the quality of the machined surface is one of the key issues. Moreover, surface integrity is a critical aspect since it affects the functional performance of the components in aspects such as fatigue, creep, corrosion, and wear resistance [1].

Conventionally, researchers try to optimize machining processes using experimental approaches based on the Design of Experiments (DoE). In this sense, they perform experimental plans, record the results and analyze them by statistical methods [2]. However, this approach is costly and time-consuming.

In the last years, Artificial Intelligence (AI) and, mainly, Machine Learning (ML) have emerged as valuable tools for analyzing large amounts of data in a wide variety of use cases (Natural Language Processing - NLP [3], sports [4], health [5]).

The sensorization of the Industry 4.0 environment and the need to parameterize the performance of the equipment have led to the exchange of information between intelligent objects for subsequent processing [6]. The need for parameterization has encouraged the implementation of Machine Learning (ML) techniques for the optimization of different industrial processes [7].

Ti6Al4V is the most generally used alpha-beta alloy [8] and, specifically, it has been extensively used in research in machining. Titanium alloys are difficult-to-cut materials because their increased strength and hardness generate high temperatures during machining and accelerate tool wear [9]. The importance of titanium alloys can be understood when attending to their applications in sectors such as aerospace, automotive, biomedical, military, petrochemical, and sports [10]. Thus, there is a large number of published studies that can provide experimental results of surface roughness in turning the Ti6Al4V alloy to feed ML algorithms.

This chapter presents a guideline for developing a lab session in which the students will learn how to use ML to optimize the parameters in titanium alloys' turning process using only published data. SMOreg, Decision Stump, and Random Forest algorithms are compared, and the results are evaluated using proper metrics.

2 Background

In turning, the material removal mechanism varies from rough turning to finish turning [11]. In this sense, Figure 1 shows the ideal cutting of both processes. The major flank plays an essential role in the cutting in rough turning. However, the cutting occurs in the region dominated by the tool nose radius in finish turning.

The quality of the surface is conventionally assessed by analyzing the roughness of the surface. It is generally accepted that surface roughness is mainly affected by the feed movement of the cutting tool and the tool nose radius. This relationship was modeled by Knight and Boothroyd [12]. In this sense, the theoretical arithmetical average value (Ra) can be calculated using Equation 1, in which f is the feed rate and re the tool nose radius:

$$Ra = 0.0321 f^2/re \tag{1}$$

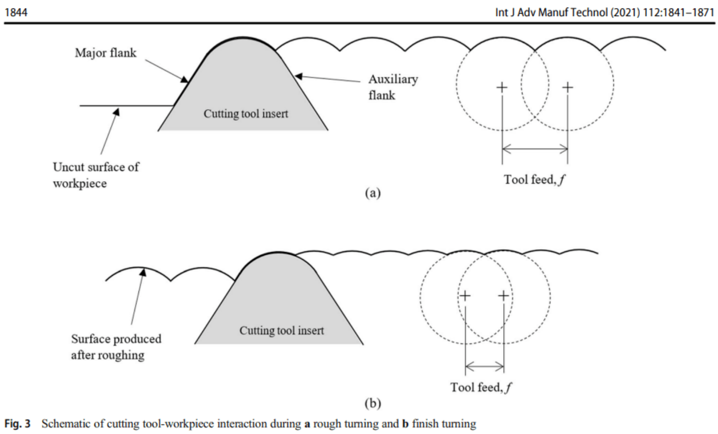

Fig. 1 Surface profile in rough turning (up) and finish turning (down). Permission granted by Derani and Ratnam [11].

The ideal surface roughness cannot easily be achieved because of the complexity of the actual turning process. Several factors affect the cutting process; therefore, the results usually vary from the theoretical ones. In this sense, external and internal loads (mechanical, thermal, chemical) may cause changes (anomalies) to the workpiece during machining. Among these anomalies, changes in surface topography are relevant [1].

Several researchers have tried identifying the main factors that may affect the turning process. Those related to the tool, workpiece, cutting parameters, and cutting phenomena are recognized among them. Some of these factors are classified and listed in Figure 2, allowing us to understand the complex nature of turning. However, when conducting turning experiments, feed rate [13] and tool nose radius [14] are critical for surface roughness, thus confirming the model established by Equation 1.

ML is an alternative approach to using traditional experimental and statistical techniques for surface roughness prediction. ML models allow detection patterns even in scenarios where there is no linearity between the features and the variable to be predicted (target feature) [16]. These techniques are generally

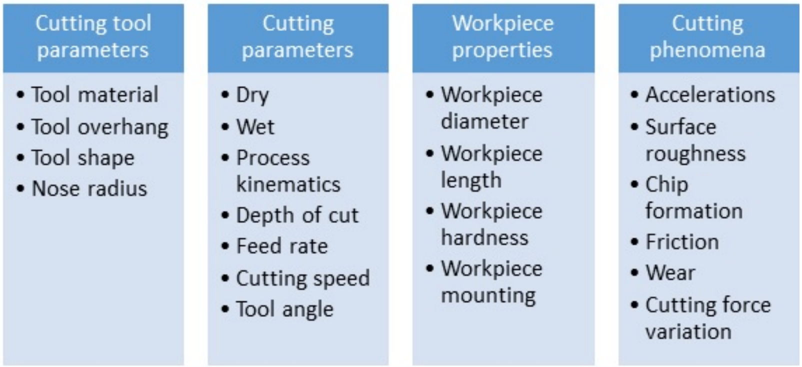

Fig. 2 Factors affecting the turning process [15].

divided into supervised and unsupervised. In the former supervised approach, an annotated data set, either by human experts or automatically labeled, is exploited. The data for training the model are called features, while the prediction is called target. The ultimate objective is to discover hidden patterns in the features that allow us to understand the relationship between them and the target. Conversely, the unsupervised approach involves discovering clusters, that is, closely related entries in an unlabeled data set using a certain distance metric. This academic practice will focus on studying supervised ML models.

As previously mentioned, to predict the target value, supervised ML models need to be trained using labeled data. In particular, the most popular ML supervised models use different mathematical approaches to obtain these patterns. In this line, regressors or the features of the Support Vector Machine (SVM) system are used based on the intersection of hyper-planes. Another usual technique uses Decision Trees. These methods generate branches based on the input features. More in detail, each fork generates a new branch in which a different feature is evaluated. These classifiers aim to divide the problem into subgroups by a range of values defining a category. These previous approaches can be combined, creating meta classifiers, as in the popular Random Forest algorithm, which combines n Decision Trees [17].

ML techniques vary depending on the type of feature of the target: (i) numerical or (ii) categorical. In the first case, the target feature corresponds to real numerical values (*e.g.*, measurements gathered from sensors, probabilities, etc.), and the classifier must be configured to act as a regressor. Conversely, when the target feature is categorical, the classifier is trained to detect patterns characteristic of each category. This chapter focuses on the first scenario [18].

3 Learning objectives

This session will help students:

- To be familiar with the evaluation of surface roughness in turning.
- To identify factors that potentially may affect surface roughness.
- To understand the foundations of ML.
- To compare various ML algorithms and evaluate their suitability using proper metrics.
- To apply ML in a data set and extract information for defining the setup and optimizing the results of the turning process.

4 Resources and organization

The session was specially designed for groups of 10-20 undergraduate engineering students. A slot of 2 hours is recommended for developing the session.

Hardware and software requirements are not highly demanding. Thus they are adequate for implementation in conventional lab spaces. The main requirements are listed below:

- Operating system: Windows 10 64 bits.
- Processor: Intel i3.
- RAM: 8 GB DDR4.
- Disk: 10 GB of free space.
- Software for data analysis: Weka[1]. This free available, portable, and easy-to-use software provides a comprehensive collection of data analysis techniques for preprocessing and modeling. Moreover, it provides visualization tools such as the graphical user interface.

One computer per student is required since students will have to follow the steps introduced by the instructors and then carry out their analysis.

There exist two alternatives to obtain data for the analysis. Firstly, to gather the data set from published data. Secondly, to obtain the data set by direct measurements on turned samples in the workshop. In addition, the two strategies can be combined, as shown by García-Martínez et al [19] for the material extrusion process, where data is gathered from the literature and by direct measurements. The approach to follow in this lab session will require only published data. However, this lab session might be enhanced, including complimentary experimental turning tests and surface roughness measurement.

The data set required for developing the session will be gathered without making any distinction between the application of the process (i.e., finishing or roughing)[2]. Based on that, the students will be provided with a data set of experimental results published in the literature on turning off the Ti6Al4V

[1] Available at https://waikato.github.io/weka-wiki/downloading_weka, June 2023.

[2] Note that this chapter is just an introduction to the methodology, not an in-depth research study.

alloy. The data set[3] proposed includes 138 values (103 for training and 35 for testing) of the *Ra* output [20–22]. The machining parameters used for obtaining these values included: cutting speed (m/min), feed rate (mm/rev), depth of cut (mm), and an individual column for each cooling technique[4]. The output is the *Ra* (µm). This data set will be used for training the model.

5 Session development

The instructor will present the background and learning outcomes to the students and the following steps within the experimental pipeline in a practical use case using Weka and a train data set. The practice is expected to last one hour and a half. After that, each student will test their knowledge with the test data set.

The ML process is divided into six fundamental stages: (*i*) data analysis, (*ii*) feature engineering, (*iii*) feature selection, (*iv*) classification, (*v*) hyperparameter optimization, and (*vi*) evaluation.

5.1 Data analysis

Data analysis represents the initial and essential step to ensure the high quality of the input data. The analyses consist of:

- **Feature mapping**. Categorical and nominal features are transformed into numeric features. Accordingly, ML models exploit numerical values to infer relevant behavior patterns based on distances, error minimization, correlations, etc. To perform feature mapping with spreadsheets, additional numerical columns must be created for each textual one, considering the ordering of the possible values.
  Using Weka, students must click the `Explorer` option and select the data set to be loaded (in `csv` or `arff` format, the latter used by Weka). In the `Filter` functionality, a list of possible filtering options will be displayed, as shown in Figure 3. The filtering options must be selected under the path `filters/unsupervised/attribute` path. Typically `StringToNominal` or `OrdinalToNumeric` are exploited. If the option `StringToNominal` is used, the data set will be saved using the `Save` option in `arff` format. Note that the unsupervised filter is unrelated to the classification task. It executes a specific rule that transforms a column from one type to another. Finally, students must open the file and modify the features to numeric format as in the Listing 1.
- **Interpolate missing data**. Several options exist to ensure that the input data have no empty values: average, minimum, maximum, etc. The selection of the strategy depends on the classification problem. Particularly, the `ReplaceMissingValues` filter in the path `filters/unsupervised/attribute` replaces the missing values with the

[3] Available at bit.ly/3CeWoD3, June 2023.
[4] Most studies do not include specific details of the tools. Thus, it is sometimes impossible to have relevant data such as tool nose radius or other turning parameters.

average. Other replacements than the average can be obtained with a formula in a spreadsheet.

- **Homogenize experimental data**. It ensures the input data is aligned among the different sources used regarding feature name, normalization, or binarization. Firstly, the same column name must be used in a spreadsheet or `csv` file. Note that the transformations must be applied in all new data sets students create.

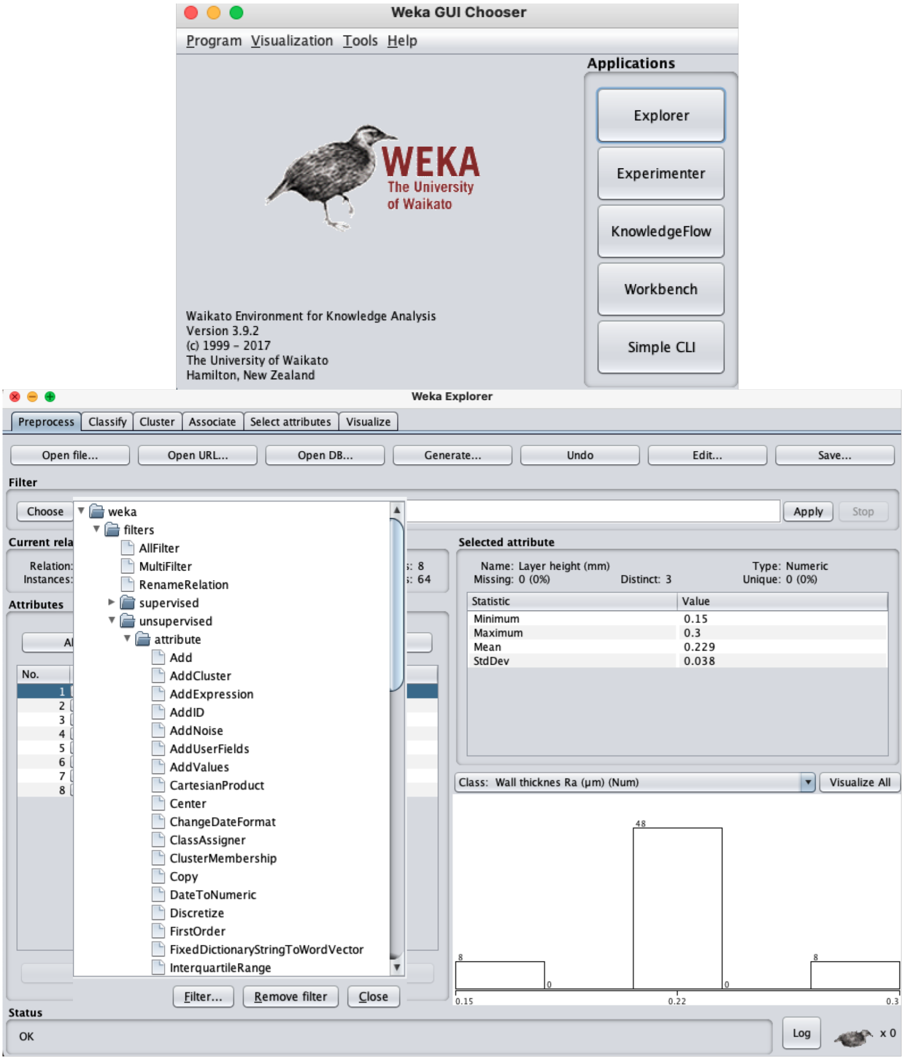

Fig. 3 Weka interface to choose a feature transform approach.

Listing 1 Nominal to numeric transformation.

```
@attribute 'Cutting speed (m/min)' {40,60}
@attribute 'Cutting speed (m/min)' numeric
```

5.2 Feature engineering

In data science, the final goal is to generate knowledge from the data [23], *i.e.*, engineer features: accumulated values (*e.g.*, from time series), the difference between two features, statistical values (*e.g.*, quartiles). This process also applies to changes in the existing data, for example, by creating a new feature by establishing ranges for the values of another feature. These changes can be done through simple functions created on spreadsheets at the end.

5.3 Feature selection

Once the input data were analyzed and the features were engineered, it was the turn to select the most representative features to train the ML models. There exist two main options in the literature: (*i*) statistical thresholding and (*ii*) model-based selection. Both approaches allow to the detection of non-duplicate feature values and highly correlated features concerning the target and are complemented by visual representation of the most relevant features. Moreover, in the model-based selection approach, a model is used to infer from a training subset the most relevant features. It serves a dual purpose: (*i*) reduction of the training samples, which results in more efficient models, and (*ii*) classification performance improvement by removing unrelated or misleading features that may lead to incorrect predictions.

Initially, in `Select attributes - Weka` tab, students must select the `Ranker` method as `Search Method` under the path `attributeSelection/Ranker` and `CorrelationAttributeEval` as `Attribute Evaluator` under the path `attributeSelection/CorrelationAttributeEval`. The latter evaluator shows results between -1 and 1. The closer to the extremes, the more correlated the features are with the target and, therefore, more representative.

For the second analysis, students must use `ClassifierAttributeEval`, as indicated in Figure 4. A tree-based classifier such as `trees/RandomForest` offers competitive performance in most scenarios. This evaluator uses a measure similar to correlation. However, its values will not be bounded. In this sense, the greater the absolute value's relevance. To launch the test, students must select the target feature from the list displayed above the `Start` button, by default the last column of the data set, and press this button. Given the results, students can move to `Preprocess` tab and remove certain features during exportation using the `Remove` button.

5.4 Classification

The models used for classification will be selected depending on the data's nature and the experimental plan's needs or limitations. Moreover, baseline models are used for comparative purposes. In this lab practice, the following algorithms will be exploited[5]:

- **SMOreg**[6] [24]. It is a regression implementation of SVM.
- **Decision Stump** (DS[7]) [25]. It is a basic tree model.
- **Random Forest** (RF[8]) [26]. It is a more complex tree-based model since it is composed of n trees.

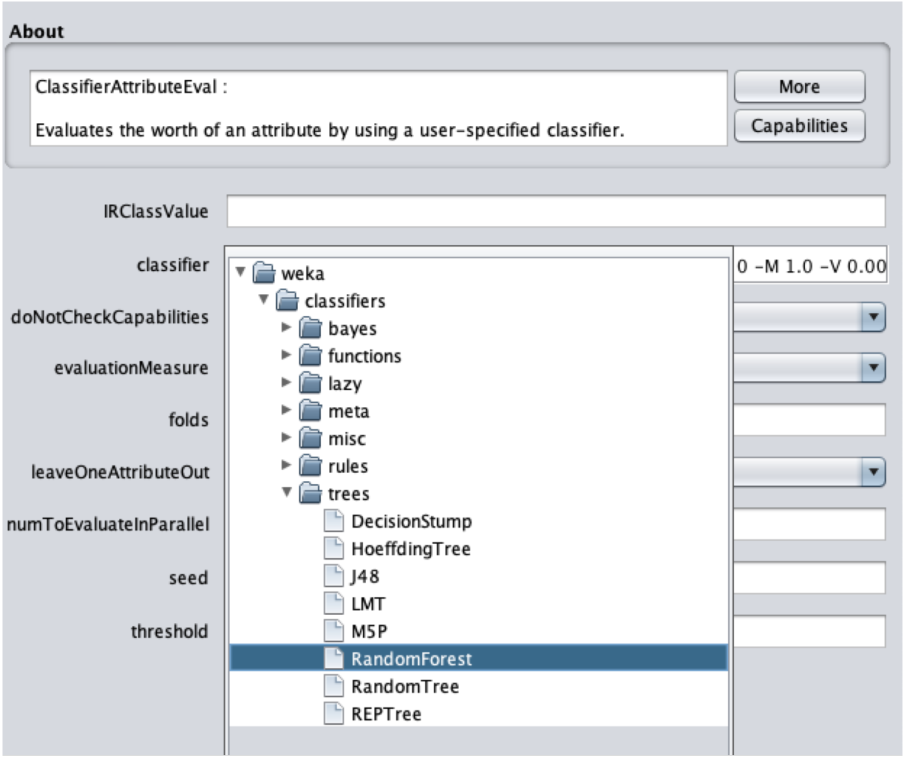

Fig. 4 Weka interface to choose a feature selection approach.

[5]However, the Weka tool provides numerous types of classifiers, the student is encouraged to explore other alternatives.
[6]Available at https://weka.sourceforge.io/doc.dev/weka/classifiers/functions/SMOreg.html, June 2023.
[7]Available at https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/DecisionStump.html, June 2023.
[8]Available at https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/RandomForest.html, June 2023.

To test classifiers on the experimental data set, students must select the `Classify` tab and choose the model by pressing the `Choose` button. The models available are in `funtions/SMOreg`, `trees/DecisionStump`, and `trees/RandomForest`.

5.5 Hyperparameter optimization

The last process prior to training and testing consists of identifying the classifiers' optimal hyperparameters. Accordingly, each classifier model has default initialized attributes that can be modified to improve performance. In line with the above, a subset of the complete data set is extracted to avoid bias and different configurations of the classifiers are tested. To select the parameters of the classifiers, students should pick the model on the right of the `Choose` button. A very common parameter to modify in the RF classifier is the `nunIterations`, as can be seen in Figure 5. In data sets larger than the one used, for example, its effect and improvement in the results are usually notable when iterating n times in search of the optimal configuration. It must be considered that the execution time and resources consumed will increase the larger this value is.

5.6 Evaluation

Traditional evaluation methods divide the data set into partitions for training and testing, like 50 %-50 %, 60 %-40 %, 70 %-30 %, or 80 %-20 %. However, this approach only ensures that the entire data set is tested and that the chosen partitions do not produce biased results in the evaluation metrics. Consequently, the most appropriate method for evaluation to avoid over-fitting and minimize over- and underestimation is cross-validation [27].

Cross-validation uses different partitions of the experimental data set over multiple iterations for training and testing the models. The latter creates a more realistic evaluation plan. The traditional 10-fold cross-validation technique shuffles the experimental data set and divides it into segments of equal length, using 9 for training and 1 for testing, without overlapped testing partitions. This process is repeated 10 times. In the end, the overall evaluation metrics are averaged.

To experiment with `Classify` tab following the train/test split procedure, `Percentage split` must be selected. The latter value corresponds to the part of the data set dedicated to training, and the rest will be used for testing. If the experimental data set is split into train and test partitions, students can use the `Supplied test set` option. In addition to the fact that both data sets must match in terms of columns, column names, and data typology, it is recommended that both the train and test files have been previously saved in `arff` format and loaded into Weka.

To use cross-validation, students must select `Cross-validation` and choose the number of folds. Exploit the `Use training set` option to check that the data set is not randomized. If, when running Weka with this option enabled, the results are far from 80 %, it means that the model cannot find patterns even when trained and evaluated with the same data set. To launch a first

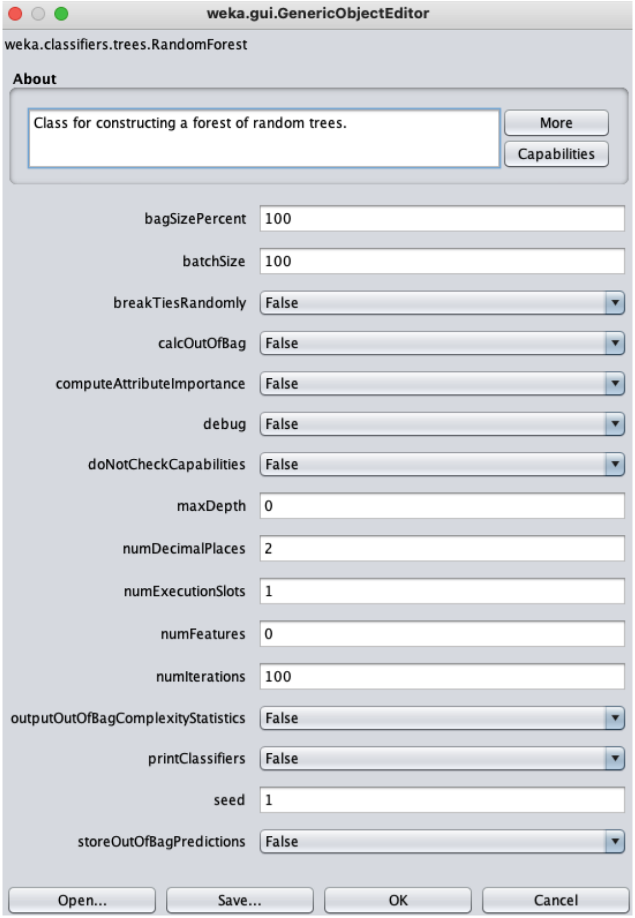

Fig. 5 Weka interface to modify the hyperparameters of the RF model.

test, choose the target variable from the list displayed under `More options` and press the `Start` button.

Since the ML models will be used as regressors, the most appropriate evaluation metrics are the relative absolute error (RAE, Equation 2) and the mean absolute percentage error (MAPE, Equation 3) instead of the accuracy, precision, and recall computed when these models are used as classifiers. The models act as regressors because the target value is numeric and may present infinite feasible values.

$$RAE(\%) = 100\frac{\sum_{i=1}^{n}|f_i - a_i|}{\sum_{i=1}^{n}|\overline{a} - a_i|}, \text{ being } f_i \text{ forecast values, and } a_i \text{ actual values.} \quad (2)$$

$$MAPE(\%) = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{f_i - a_i}{a_i}\right| \tag{3}$$

The RAE metric measures the model's adaptability regarding feature-target value changes. More in detail, it computes the average-percentual deviation between the predicted value and the expected one. Moreover, we are using the MAPE metric since the surface roughness values are expected to be highly sparse. More in detail, the MAPE metric assesses the same deviation (*i.e.*, between predicted and actual values) but is normalized to the actual values rather than the average as in the case of the RAE.

The correlation variable and RAE will be printed in the display window by default. To select new metrics, click on the `More options` button and `Evaluation metrics` as shown in Figure 6. Note that the MAPE metric is not included. Therefore, students must use a spreadsheet to calculate it. To export the prediction in the `Classifier evaluation options` window, click on `Output predictions` and select `csv`. When executing, the id of the instance, the real value, the predicted value, and its difference will be printed on the screen, each parameter separated by commas. The student can select this data, copy and paste it to a `csv` file and open it with a spreadsheet application.

The results using the cross-validation method with the proposed data sets are presented in Table 1. The behavior of RF is superior in all metrics. RF correlates 0.96 with the surface roughness target feature. Moreover, a MAPE value of 19.82 % represents a deviation between actual and predicted values lower than 20%. In contrast, the 31.12 % of RAE metric is expected because the actual and predicted measures are not close to the mean. For this reason, the MAPE measure is a more accurate metric in this type of problems.

6 Outcomes

Table 1 Result values for the selected ML models using 10-fold cross-validation technique.

	SMOreg	**DS**	**RF**
Correlation	0.84	0.88	0.96
RAE (%)	55.99	53.56	31.12
MAPE (%)	36.28	35.63	19.82

Once the instructors have delivered the guidelines to the students, they are expected to use the test data set. Students must use the original data set to train the model and the test data set to validate it using `Supplied test set option`. Verify that this data set has the same column names and types, or solve any problem following the steps described in Section 5.1. The next step will be to analyze the information and select the relevant features. Moreover,

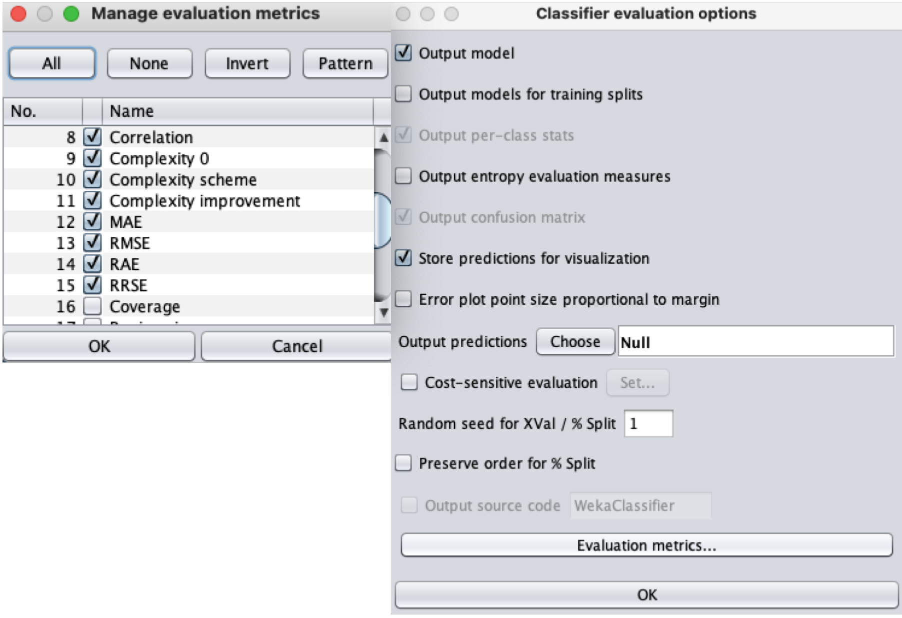

Fig. 6 Weka interface to choose the evaluation metrics.

they must evaluate the three proposed classification models using the evaluation metrics provided by Weka and including the RAE and MAPE results obtained. They are expected to discuss performance differences among the classifiers. Students can improve the latter performance using hyper-parameter tuning and checking their effect on the evaluation metrics. The results with all features and with the default configuration are shown in Table 2. As in the same case of cross-validation, RF performs better than the rest of the classifiers. The student shall make as same discussion analyses as in the above experiment.

Table 2 Result values for the selected ML models using `Supplied test set option`.

	SMOreg	DS	RF
Correlation	0.87	0.88	0.98
RAE (%)	48.45	56.86	32.32
MAPE (%)	21.47	25.58	12.69

7 Deliverable and assessment

At the end of the session, students must deliver a summary of the techniques and tools used for data analysis, feature engineering, and selection (*e.g.*, interpolate missing data, homogenize experimental data, etc.). Additionally, this summary must include the features engineered and selected and their relevance. Results must be displayed in a table containing the correlation values, the RAE, and the MAPE metrics for each classifier used. Moreover, an additional table with the results obtained with hyper-parameter tuning must be included. A deliverable template is provided to the students[9].

8 Conclusions

The present chapter provides an example of the use of ML to analyze published data to optimize the turning conditions of Ti6Al4V alloy. The main conclusions of the chapter are the following:

- The main mechanisms that generate the surface roughness in turning are well known. The ideal surface roughness profile primarily depends on the feed rate and tool nose radius.
- The actual surface roughness differs from the ideal values. Thus, the setup for turning cannot be arranged using only feed rate and tool nose radius.
- ML is suitable for analyzing complex relationships between multiple factors and outcomes, especially when large data sets are available.
- The students will learn how to use and apply ML methods in manufacturing. Specifically, they will create a setup for the turning process of Ti6Al4V alloy using as the outcome, the surface roughness.

Acknowledgements

This study was partially supported by Xunta de Galicia grants ED481B-2021-118 and ED481B-2022-093, Spain.

References

[1] Liao Z, la Monaca A, Murray J, et al (2021) Surface integrity in metal machining - Part I: Fundamentals of surface characteristics and formation mechanisms. International Journal of Machine Tools and Manufacture 162:103,687–103,737. https://doi.org/10.1016/j.ijmachtools.2020.103687

[2] Lauro CH, Pereira RBD, Brandão LC, et al (2016) Design of Experiments—Statistical and Artificial Intelligence Analysis for the Improvement of Machining Processes: A Review. Springer, https://doi.org/10.1007/978-3-319-23838-8_3

[3] Bashir MF, Arshad H, Javed AR, et al (2021) Subjective Answers Evaluation Using Machine Learning and Natural Language Processing. IEEE Access 9:158,972–158,983. https://doi.org/10.1109/ACCESS.2021.3130902

[9] Available at bit.ly/3X45Wdr, June 2023.

[4] Nguyen NH, Nguyen DTA, Ma B, et al (2022) The application of machine learning and deep learning in sport: predicting NBA players' performance and popularity. Journal of Information and Telecommunication 6:217–235. https://doi.org/10.1080/24751839.2021.1977066

[5] Ngiam KY, Khor IW (2019) Big data and machine learning algorithms for health-care delivery. The Lancet Oncology 20:262–273. https://doi.org/10.1016/S1470-2045(19)30149-4

[6] Jasperneite J, Sauter T, Wollschlaeger M (2020) Why We Need Automation Models: Handling Complexity in Industry 4.0 and the Internet of Things. IEEE Industrial Electronics Magazine 14(1):29–40. https://doi.org/10.1109/MIE.2019.2947119

[7] Carou D, Sartal A, Davim JP (2022) Machine Learning and Artificial Intelligence with Industrial Applications. Springer, https://doi.org/10.1007/978-3-030-91006-8

[8] Alam ST, Tomal AA, Nayeem MK (2023) High-Speed achining of Ti–6Al–4V: RSM-GA based Optimization of Surface Roughness and MRR. Results in Engineering 17:100,873–100,883. https://doi.org/10.1016/j.rineng.2022.100873

[9] Revuru RS, Posinasetti NR, VSN VR, et al (2017) Application of cutting fluids in machining of titanium alloys—a review. The International Journal of Advanced Manufacturing Technology 91:2477–2498. https://doi.org/10.1007/s00170-016-9883-7

[10] Pushp P, Dasharath S, Arati C (2022) Classification and applications of titanium and its alloys. Materials Today: Proceedings 54:537–542. https://doi.org/10.1016/j.matpr.2022.01.008

[11] Derani MN, Ratnam MM (2021) The use of tool flank wear and average roughness in assessing effectiveness of vegetable oils as cutting fluids during turning—a critical review. The International Journal of Advanced Manufacturing Technology 112:1841–1871. https://doi.org/10.1007/s00170-020-06490-5

[12] Knight WA, Boothroyd G (2019) Fundamentals of Metal Machining and Machine Tools. CRC Press, https://doi.org/10.1201/9780429114243

[13] Anurag, Kumar R, Sahoo AK, et al (2022) Comparative Performance Analysis of Coated Carbide Insert in Turning of Ti-6Al-4V ELI Grade Alloy under Dry, Minimum Quantity Lubrication and Spray Impingement Cooling Environments. Journal of Materials Engineering and Performance 31:709–732. https://doi.org/10.1007/s11665-021-06183-4

[14] Schultheiss F, Hägglund S, Ståhl JE (2015) Modeling the cost of varying surface finish demands during longitudinal turning operations. The International Journal of Advanced Manufacturing Technology 84:1103–1114. https://doi.org/10.1007/s00170-015-7750-6

[15] Serra R, Chibane H (2010) Effects of cutting parameters during turning 100C6 steel. In: EPJ Web of Conferences, pp 1–8, https://doi.org/10.1051/epjconf/20100613004

[16] Thongpeth W, Lim A, Wongpairin A, et al (2021) Comparison of linear, penalized linear and machine learning models predicting hospital visit costs from chronic disease in Thailand. Informatics in Medicine Unlocked 26:100,769–100,776. https://doi.org/10.1016/j.imu.2021.100769

[17] Witten IH, Frank E, Hall MA, et al (2016) Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, https://doi.org/10.1016/C2009-0-19715-5

[18] Subasi A (2020) Practical Machine Learning for Data Analysis Using Python. Elsevier, https://doi.org/10.1016/B978-0-12-821379-7.00008-4

[19] García-Martínez F, Carou D, de Arriba-Pérez F, et al (2023) Toward data-driven research: preliminary study to predict surface roughness in material extrusion using previously published data with machine learning. Rapid Prototyping Journal pp 1–13. https://doi.org/10.1108/RPJ-01-2023-0028

[20] Chen Y, Sun R, Gao Y, et al (2017) A nested-ANN prediction model for surface roughness considering the effects of cutting forces and tool vibrations. Measurement 98:25–34. https://doi.org/10.1016/j.measurement.2016.11.027

[21] Mia M, Gupta MK, Lozano JA, et al (2019) Multi-objective optimization and life cycle assessment of eco-friendly cryogenic N2 assisted turning of Ti-6Al-4V. Journal of cleaner production 210:121–133. https://doi.org/10.1016/j.jclepro.2018.10.334

[22] Amrita M, Kamesh B, Sree KLS (2022) Multi-response optimization in machining Ti6Al4V using graphene dispersed emulsifier oil. Materials Today: Proceedings 62:1179–1188. https://doi.org/10.1016/j.matpr.2022.04.352

[23] Moreno-Mateos MA, Carou D (2022) A Note on Big Data and Value Creation. Springer, https://doi.org/10.1007/978-3-030-91006-8_1

[24] Assim M, Obeidat Q, Hammad M (2020) Software Defects Prediction using Machine Learning Algorithms. In: Proceedings of the International Conference on Data Analytics for Business and Industry: Way

Towards a Sustainable Economy. IEEE, pp 1–6, https://doi.org/10.1109/ICDABI51230.2020.9325677

[25] Uskov VL, Bakken JP, Putta P, et al (2021) Smart Education: Predictive Analytics of Student Academic Performance Using Machine Learning Models in Weka and Dataiku Systems. In: Proceedings of the Smart Innovation, Systems and Technologies Conference, pp 3–17, https://doi.org/10.1007/978-981-16-2834-4_1

[26] Schonlau M, Zou RY (2020) The random forest algorithm for statistical learning. The Stata Journal: Promoting communications on statistics and Stata 20(1):3–29. https://doi.org/10.1177/1536867X20909688

[27] Berrar D (2019) Cross-Validation. In: Encyclopedia of Bioinformatics and Computational Biology. Elsevier, p 542–545, https://doi.org/10.1016/B978-0-12-809633-8.20349-X