Real-time prediction of Wikipedia articles' quality

Pedro Miguel Moás[1][0009−0000−1659−3590] and Carla Teixeira Lopes[2][0000−0002−4202−791X]

[1] Faculdade de Engenharia, Universidade do Porto, R. Dr. Roberto Frias, 4200-465 Porto, Portugal
[2] INESC TEC, Faculdade de Engenharia, Universidade do Porto, R. Dr. Roberto Frias, 4200-465 Porto, Portugal, Portugal
{pmmoas,ctl}@fe.up.pt

Abstract. Wikipedia is the largest and most globally well-known online encyclopedia, but its collaborative nature leads to a significant disparity in article quality. In this work, we explore real-time and automatic quality assessment within Wikipedia through machine-learning. We first constructed a dataset of 36,000 English articles and 145 features, then compared the performance of multiple classification and regression algorithms and studied how the number of classes and features affects the model's performance. The six-class experiments achieved a classifier accuracy of 64% and a mean absolute error of 0.09 in regression methods, which matches or beats most state-of-the-art approaches. Our model produces similar results on some non-English Wikipedias, but the error is slightly higher on other versions. We have also determined that the features measuring the article's content and revision history bring the largest performance boost.

Keywords: Wikipedia · Quality Assessment · Machine Learning.

1 Introduction

Wikipedia is the world's largest and most well-known online encyclopedia and has kept its growing pace for years. As of June 2025, it contains over 7 million English articles [29], with versions across a list of 342 active languages [25]. Additionally, from June 2024 to May 2025, Wikipedia totalized close to 300 billion page views across 2 billion unique devices [22], suggesting that many people worldwide consider Wikipedia a viable source of information.

Wikipedia relies on human volunteers to add and maintain its content, averaging a rate of 8.8 edits per second from June 2024 to May 2025 [22]. However, since there is no centralized authority over the article editors, ensuring quality throughout the website is a challenging task.

Another central issue is the substantial discrepancy between the quality of English and non-English articles. For starters, non-English Wikipedias cover a much smaller number of articles [25]. Furthermore, they are often much more incomplete as well, as demonstrated by Roy et al. [17]. For instance, they determined that German articles are, on average, 30% shorter than English ones, and

Moás, P.M., Teixeira Lopes, C. (2026). Real-Time Prediction of Wikipedia Articles' Quality. In: Balke, WT., Golub, K., Manolopoulos, Y., Stefanidis, K., Zhang, Z. (eds) Linking Theory and Practice of Digital Libraries. TPDL 2025. Lecture Notes in Computer Science, vol 16097. Springer, Cham. https://doi.org/10.1007/978-3-032-05409-8_18

for Spanish articles, that value increases to 47%. Considering that around 57% of total Wikipedia traffic goes to non-English articles [22], this indicates a significant problem for international readers. These points show that Wikipedia's quality may vary significantly across different articles and languages. However, readers have no easy methods for quickly assessing quality, so they must rely on their judgment to determine a Wikipedia page's value.

Envisioning the construction of a Machine Learning (ML) model that automatically predicts the quality of Wikipedia articles, we conducted a thorough experimentation guided by the following research questions:

RQ1. Which ML algorithm yields the best performance in predicting the quality of Wikipedia articles?
RQ2. How does model performance vary with the number of output classes?
RQ3. Which subset of features produces the best performance?
RQ4. How effective is our model at predicting the quality of non-English articles?

In summary, in this article, we make 2 main contributions. First, we identify 145 features that can be automatically computed to predict the quality of Wikipedia articles. These features are grouped into five categories: content (33 features), style (72 features), readability (10 features), history (22 features), and network (8 features). We computed all the features for a set of 36,000 English articles and some features for four smaller groups of articles used in the multi-language evaluation: English (11,096 articles), French (11,195 articles), Portuguese (10,525 articles), and Russian (10,341 articles). These datasets are publicly available in a research data repository and can be used in future research [16]. Second, we design a machine-learning solution that effectively predicts Wikipedia article quality into six classes (FA, GA, B, C, Start, and Stub), outperforming most state-of-the-art approaches. The source code for the development and testing of the model is publicly available.

2 Background and Related work

This section discusses different meanings of "quality" and analyzes the existing literature relating to the subject of this paper.

2.1 Information Quality

Information Quality (IQ) is an extraordinarily researched topic, and numerous ways exist to measure it. Lee et al. [13] systematize that process by representing quality as 15 different properties, such as *Accessibility*, *Believability*, and *Interpretability*. We do not plan to assess properties relating to the integrity of the information, as that would be more suitable for manual approaches. Regardless, there might be some correlation between some quality properties and others. Due to Wikipedia's collaborative aspect, contributors will, in theory, cooperate

to improve the article continuously, so it is reasonable to assume that well-written articles will be trustworthy more often than not.

Some may say there is no truly objective way to quantify quality. However, there are undoubtedly measurable characteristics that people often relate to outstanding quality, and we plan to determine those and their correlation with excellence in written documents.

2.2 Wikipedia's Quality Scale

To monitor and maintain content quality, Wikipedia developed a nine-grade scale that rates articles from incomplete entries documents(Stubs and Starts) to comprehensive, well-written ones (Featured Articles) [27]. Among the six intermediate levels, this work focuses on three: Good Articles (GA), B-class, and C-class, listed in descending order of quality. Notably, most English Wikipedia consists of lower-quality articles, with Starts and Stubs accounting for more than 80% of its content, while the share of Featured and Good Articles is around 0.6% [27]. These values are only meant for internal use by Wikipedia's contributors and are not to be taken as official ratings. Besides, not every English article is rated, and the other versions of Wikipedia that assess their content will have distinct quality scales, evaluated with different criteria.

2.3 State of the Art

A systematic literature review was conducted to explore current methods for automatically assessing Wikipedia article quality [15]. The review included 149 papers, 71 of which employed machine learning techniques. While most studies focus on English Wikipedia, 35 examined non-English versions, and 7 addressed multilingual quality assessment.

We collected 321 unique features, all of which consider either the text of an article, its revision history, or how it relates to other articles within Wikipedia or the internet. The textual features, which factor in the content of the Wikipedia article, are the most recurring of the ones we collected, forming almost 90% of the set. Some studies also proposed IQ metrics—formula-based approaches that can function independently or support machine-learning models.

Both deep learning and classical algorithms have shown promising results. Deep learning achieved 68–76% accuracy in 6-class setups, though these methods are less suited for real-time applications. so we are better off comparing ourselves with the classical learning accuracies, which range between 62% and 73% in the top 3. In simpler 2-class scenarios with balanced datasets, multiple studies reported accuracy above 95%.

In summary, the study of the automatic quality assessment of Wikipedia has become more prevalent over the years. Feature-based machine learning models appear to have been a solid option throughout the entire period, though. We will use these findings as a basis for our work, conducting further experimentation to develop better quality predictors.

3 Methodology

This section explains in detail all the steps involved in developing our solution. We opted for using a machine-learning approach, so we needed to define a dataset, automate the calculation of features, and then compare the performance of different algorithms, maximizing the model's accuracy. The datasets we constructed are entirely available in a research data repository [16], and the source code is fully accessible from a GitHub repository [3].

3.1 English Dataset Definition

Using MediaWiki's API, we first obtained a list of all the titles of each quality class. Wikipedia provides a category page for each quality class, listing the pages assessed as such, so we used the API to fetch the elements of each category.

Owning all the Wikipedia titles and the quality of their respective pages allowed us to create a randomized dataset more effortlessly. We decided not to include the FL and A-class items in our experiments for distinct reasons. First, the goal of Featured Lists is to thoroughly list other items, and often do not contain much more content. Regarding the A-class tier, they correspond to a tiny portion of Wikipedia [27] and do not follow the linear quality path, as being a GA (Good Article) is not a requirement for being A-class.

Collecting the articles was a straightforward process. We selected 6,000 random titles for each class and used MediaWiki's API to obtain their contents. We arrived at that number because we wanted to keep the dataset balanced, and the relatively low supply of FA's does not allow us to expand it further.

Network Graph Generation We can view Wikipedia as an "interconnected network of articles", where each entry can link to other Wikipedia articles. For that reason, we can consider a collection of hyperlinked pages as a directed graph, where the nodes represent the articles and edges indicate the existence of a link between them [1]. Certain features require knowing some properties of the article's graph, such as the in-degree and out-degree. Therefore, before feeding the page contents to the feature calculator, we construct a network graph of each article we assess.

The Laboratory for Web Algorithms provides datasets of several networks [6], including Wikipedia. We used the `enwiki-2022` dataset, which contains over 6 million nodes and 159 million arcs, then processed them with the WebGraph framework I, a library that eases the manipulation of large Web graphs [5].

The `enwiki-2022` dataset was produced in the same week we ran the experiments described in this document, so the values are up-to-date. However, when we randomly select the 36,000 articles to include in our dataset, we sometimes encounter titles that are not present in `enwiki-2022`. Those occurrences are rare, so we skip the missing items and do not include them in the dataset.

[3] GitHub Repository URL: `https://github.com/MOAAS/WikiQuality`

Training and testing datasets After calculating each article's entire set of features, our final step was to partition our data into a training set and a testing set. We decided on a 70/30 split, 70% for training and 30% for testing data, as previously done by multiple authors [20,14]. Table 1 shows the final distribution of articles in our dataset. We initially picked 6,000 titles from each level of quality, but we did not compensate for missing items during the generation of network graphs. Fortunately, that step showed an insignificant number of missing results, so we consider our dataset practically balanced.

Table 1: Article quality distribution within the dataset

	FA	**GA**	**B**	**C**	**Start**	**Stub**	**Total**
Training	4,200	4,200	4,197	4,199	4,196	4,198	25,190
Testing	1,800	1,800	1,798	1,799	1,797	1,798	10,792
Total	6,000	6,000	5,995	5,998	5,993	5,996	35,982

3.2 Feature Set

This section describes our feature list and the processes involved in calculating it, which account for over 80% of the dataset's build time. Each feature belongs to one of five different categories: Content, Style, Readability, History, and Network, and every feature has an assigned identifier that always starts with the first letter of its category. We provide an exhaustive detailing of all the used features and their identifiers at our source code repository [4].

Content Features (33 features) Content features analyze the length and structure of the article (e.g., Character Count, Citation Count, Number of Images). All of them can be directly obtained from the plain text and original wikitext, so they are the fastest to compute.

Style Features (72 features) Style features evaluate how contributors write their sentences, measuring the used word classes and sentence types (e.g., Question Count, Verb Count). They can all be derived from the article's plain text. These features are also relatively fast to compute. However, the NLP process with the NLTK tool sometimes takes a while (up to a couple of seconds for very long articles, which potentially amounts to hours with 36,000 articles).

Readability Features (10 features) Readability features assess how easy a text is to read, using carefully designed formulas that combine word, syllable, and character count. This includes metrics like Automated Readability Index [18], Coleman-Liau[7], and Flesh-Kincaid [11].

[4] `https://github.com/MOAAS/WikiQuality/blob/main/ml/datasets/README.md`

Applying the formulas was a swift and straightforward process, considering they only require values we had previously determined with the assistance of NLP tools (e.g., number of syllables and number of sentences). Calculating the number of complex words was an exception, though, for we had also to collect the list of easy words[5] initially proposed by Dale and Chall [9].

History Features (22 features) History features measure an article's revision history and analyze its authors and contributions (e.g., Number of Revisions, Number of contributors). Stable articles with trustworthy authors tend to be better than controversial articles with frequent edits. The features within this category aim to measure articles based on this principle.

Measuring history features takes significant time because we must first obtain the entire revision history of its respective article through MediaWiki's API. Pages with more contributions will take longer to acquire. For example, the United States article[6] has the second-largest history between English Wikipedia articles, reaching 52,329 revisions (June 2025) [28]. During experimentation, fetching all its contributions took around 60 seconds. However, analyzing our dataset, most articles rarely have that many edits. For instance, considering only the Stubs, Starts, and C-class articles, which comprise the majority of Wikipedia, 92.8% of them have under 500 revisions.

Network Features (8 features) Network features consider Wikipedia as a graph, where its nodes represent articles and the edges are the citations between them. Some examples include the in-degree and out-degree of an article. We can derive nearly all the features from the article's network graph, except for NT (Number of Translations), which we obtain from the MediaWiki API. However, graph generation is a highly time-consuming step, so this category is likely the least appropriate for real-time quality assessment.

Feature Computation Some features required us to isolate the text of the article from all the wikitext tags and templates. Therefore, we clean the article's wikitext before extracting its features, obtaining the filtered plain text. We remove some irrelevant sections during the cleaning process, such as references and footnotes. However, since we need those to compute certain features, we still preserve the original wikitext. We also remove many non-textual templates, like `{chess diagram}` and `{location map}`, and ignore formatting tags, such as `<small>` and `<center>`, only keeping the text inside them. Wikitext has a complex syntax and countless tags, so it is difficult to determine all the edge cases. However, even if our wikitext cleaner is not flawless, empirical findings show almost perfect results.

Several feature categories demand a specialized analysis of the plain text we extract from the wikitext (i.e., Natural Language Processing, NLP). We used

[5] `http://countwordsworth.com/download/DaleChallEasyWordList.txt`
[6] Wikipedia article is available at: `https://en.wikipedia.org/wiki/United_States`

NLTK (Natural Language Toolkit) [3,4], a library that allows Python programs to process natural language data. It provides tokenization for isolating sentences and words, and POS (part-of-speech) tagging for identifying word classes. We also relied on the `syllables` python package[7] to estimate the number of syllables in each word. Although these tools are powerful, they are mainly designed to work for English texts. Using them for a few other Western languages is possible, but they will not perform as accurately, so we must take that issue into account during experimentation.

3.3 Experimentation

This phase of our study was extensive, as it was essential for developing an accurate machine-learning model. This section details the experiments we conducted to answer the research questions we defined. Each experiment and respective model will have an assigned identifier with the following format:

`<CATEGORIES><CLASSES>/<ALGORITHM>_<TASK>`

We also explain this naming convention in Table 2. By default, our model will train with the six classes and all the described features (i.e., `CSRHN6/*`).

Table 2: Training Experiments: Naming Convention

Element	Description	Example
`<CATEGORIES>`	Feature categories included in the dataset. Each category is represented by its initial letter. Possible categories: Content, Style, Readability, History, Network.	CSRHN
`<CLASSES>`	Number of output classes considered for training the model. If it is less than 6, multiple levels of quality may have the same class (more information on Tables 5 and 6).	6
`<ALGORITHM>`	Algorithm used to train the model.	tree
`<TASK>`	Whether the chosen algorithm solves classification (c) or regression (r) tasks.	c

Algorithm Comparison We used 16 machine-learning algorithms—eight for classification and eight for regression—and conducted extensive experiments to identify the most effective models and hyperparameters, as algorithm choice greatly influences performance.

We trained the models using *scikit-learn* [12]. Table 3 lists the different algorithms we experimented with, along with the hyper-parameters we found to be most suitable (unmentioned parameters were left to *scikit-learn*'s default values). We determined them using a combination of manual and grid search, which

[7] The `syllables` package is available at: `https://pypi.org/project/syllables/`

do not necessarily give the optimal values but are still widely used in the current state-of-the-art [2].

Furthermore, many of these algorithms require the normalization of the input data to achieve maximum performance [10], so we invariably applied z-score scaling before training, which normalizes the features to have a mean value of 0 and a standard deviation of 1 [19].

Table 3: Used Machine Learning Algorithms and Hyper-Parameters

Algorithm		**Parameter Name**	**Parameter value for**	
Name	**ID**		**Classification**	**Regression**
Decision Tree	tree	criterion	entropy	squared_error
		max_depth	15	15
Random Forest	forest	n_estimators	150	150
		criterion	entropy	squared_error
		max_depth	15	15
Ada Boost	ada	n_estimators	150	150
		learning_rate	0.1	0.1
		base_estimator	tree_c	tree_r
Gradient Boost	gboost	n_estimators	150	150
		learning_rate	0.01	0.01
Support Vector Machine	svc/svr	kernel	rbf	rbf
		C	2	2
Multi-Layer Perceptron	mlp	hidden_layer_sizes	(100)	(100)
		alpha	0.1	0.1
		max_iter	10,000	10,000
K-Nearest Neighbors	knn	n_neighbors	10	-
Gaussian Naive-Bayes	gnb	-	-	-
Logistic Regression	logreg	max_iter	10,000	-
Linear Regression	linreg	-	-	-

Quality Classes Mapping Since most regression algorithms struggle with categorical labels, we converted quality labels to numerical values. Following Teblunthuis [21], who notes Wikipedia's quality levels aren't evenly spaced, we used a non-linear scale based on his ordinal regression model. This mapping, shown in Table 4), does not affect classification experiments because those algorithms do not consider the values of each label.

Table 4: Regression Experimentation Class Mapping vs. Linear Class Mapping

Class	FA	GA	B	C	Start	Stub
Default	1.0	0.9	0.7	0.6	0.4	0.0
Linear	1.0	0.8	0.6	0.4	0.2	0.0

Although the dataset distinguishes between six distinct classes of quality, we also decided to study the model performance on a smaller number of classes. We thus conducted four different experiments for every algorithm, varying the number of classes in each one, but keeping all the features. Table 5 shows how we aggregated quality classes for each regression experiment. For example, in 5-class experiments, the classes B and C are merged in one. It is also worth noting that the 5-class and 2-class mappings lead to an unbalanced dataset, as we merge the examples into the same classes. Therefore, in those cases, we randomly removed excess rows from every label until the classes were distributed evenly.

Table 5: Quality-to-class Mapping in Regression

# Classes	FA	GA	B	C	Start	Stub
6	1.0	0.9	0.7	0.6	0.4	0.0
5	1.0	0.9	0.6	0.6	0.4	0.0
3	1.0	1.0	0.6	0.6	0.0	0.0
2	1.0	1.0	0.0	0.0	0.0	0.0

Table 6: Quality-to-class Mapping in Classification

# Classes	FA	GA	B	C	Start	Stub
6	C_1	C_2	C_3	C_4	C_5	C_6
5	C_1	C_2	C_3		C_4	C_5
3	C_1	C_1	C_2		C_3	
2	C_1		C_2			

Although classification experiments do not look at the value numbers, they will be based on the regression mapping, meaning quality levels with equal value will be considered the same class. Table 6 clarifies which classes we merged together for the classification experiments.

Feature Subset Optimisation We trained 14 extra models for each previously listed algorithm, in each run varying the feature set and keeping all six classes. Table 7 lists those experiments, showing the different combinations of feature categories we tested. The combination IDs follow the naming convention previously defined in Table 2.

Table 7: Model Experimentation: Feature Subset Combinations (Transposed)

Feat	CSRHN	SRHN	CRHN	CSHN	CSRN	CSRH	C	S	R	H	N	CSR	CSH	CRH	CH
C	×		×	×	×	×	×			×		×	×	×	×
S	×	×		×	×	×		×				×	×		
R	×	×	×		×	×			×			×		×	
H		×	×	×		×				×			×	×	×
N	×	×	×	×	×						×				
#	145	112	73	135	123	137	33	72	10	22	8	115	127	65	55

Multi-language Performance We constructed four new datasets, each respective to a different language. They all distinguish six different levels of quality, but

each Wikipedia version has a different naming convention for its quality levels. To avoid confusion, we normalized the classes into letter grades, ranging from A+ to F. Table 8 displays the mapping between the classes and their respective letters.

Table 8: Model Experimentation: Quality Levels per Language

Standard	**A+**	**A**	**B**	**C**	**D**	**F**
English [27]	FA	GA	B	C	Start	Stub
French [26]	AdQ	BA	A	B	BD	Ébauche
Portuguese [23]	6	5	4	3	2	1
Russian [24]	ИС	ХС	I	II	III	IV

We did not find any Wikipedia version that provides a list of articles for every quality level, aside from the English one. We instead used MediaWiki's API to obtain random articles, determining their quality from the discussion page. Although that approach produces an unbalanced dataset, it summarizes the quality distribution within each Wikipedia. We discarded some titles while collecting the articles, as some items had no quality assigned. Therefore, the dataset sizes slightly differ between them, with English having 11,096 articles, French 11,195, Portuguese 10,525, and Russian 10,341.

Besides considering diverse languages, we experimented with three different feature sets for each one: **CSRH**, **CRH**, and **CH**. We always excluded network features because we do not have access to WebGraph datasets for other Wikipedia versions. We also tested the model without style and readability features because those are mostly inaccurate outside the English domain, especially the readability ones [8]. We selected the *forest_r* models for these experiments as that algorithm performed best on the standard dataset.

4 Results

This section discusses the main findings of our machine-learning experiments. Apart from the ones listed in Section 4.4, where we analyze the performance of three top-performing regression models (*CSRH6/forest_r*, *CRH6/forest_r*, and *CH6/forest_r*) on a multi-language scenario, all the results relate to experiments on the English Wikipedia using the 36,000 articles dataset. Since our result data is extensive, we summarize the essential information using tables and graphs, but all the collected results are available in a research data repository [16].

4.1 Algorithm Performances

We collected multiple performance metrics from our experimentations. Figure 1 shows the best-performing model was the Support Vector Classifier, achieving 64% accuracy. Most of the other algorithms reached similar values, ranging from

60% to 62%. When applied to the training set, these models often achieved 95 to 100% accuracy. These high accuracies suggest our approach is vulnerable to overfitting, so we will only consider results from the testing set for the rest of this section.

Then, we decided to analyze algorithm performance per class, because we suspected the models were performing worse on certain quality levels. Figure 2 suggests all algorithms have more difficulty predicting the quality of mid-tier articles than the extreme ones, likely because B and C levels are very similar quality-wise.

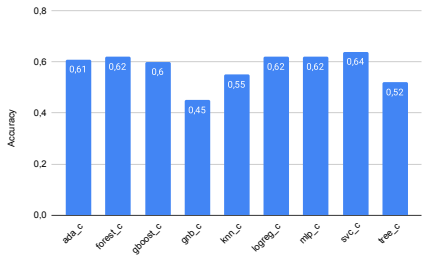

Fig. 1: Accuracy of classification models trained with all features and classes

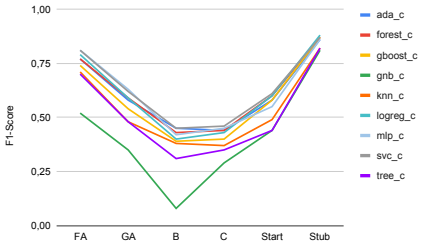

Fig. 2: F1-score of classification models trained with all features and classes, per article quality

We have also compared the performance of the multiple regression algorithms. We show the Mean Absolute Error (MAE) of the predictions in Figure 3, as it is considered an appropriate indicator of the average error [30]. In this case, Random Forest has shown the best results (0.09), but almost all others display MAE values around 0.1. We also performed a per-class analysis of the regression results, showing the comparison in Figure 4 and the following findings:

- Gradient Boosting shows a unique pattern compared to the other algorithms, achieving the lowest error rate for mid-tier articles.
- Most algorithms peak the MAE on the Start articles, probably because its quality value (0.4) is more distant from its adjacent levels (Stub - 0.0, C - 0.6). Conversely, the error bottoms at featured articles because its only adjacent level (GA) is only 0.1 points lower.
- Otherwise, the error rates usually do not vary significantly throughout the multiple classes.

4.2 Analysis of Number of Classes

It is expected that model accuracy increases as the number of classes decreases, but we decided to assess which algorithms would perform better with each number of classes.

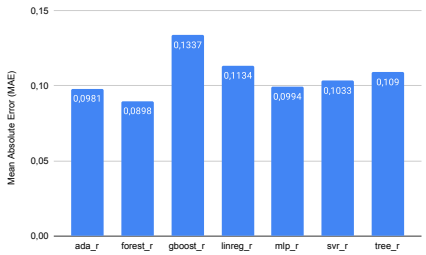

Fig. 3: MAE of regression models trained with all features and classes

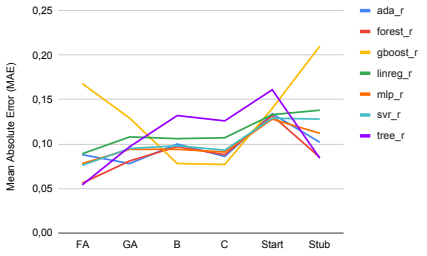

Fig. 4: MAE of regression models trained with all features and classes, per article quality

Comparing classification results is straightforward, as we can directly compare the obtained accuracies. However, for the regression algorithms, directly comparing the mean error does not give great insight into performance improvement, as fewer classes lead to more spaced-out levels of quality and may lead to a larger error. For that reason, we compared the values of $Avg_{Dist} = \frac{MAE}{Avg_{Spacing}}$, where $Avg_{Spacing} = \frac{1}{NumClasses-1}$. In other words, $Avg_{Spacing}$ is the average distance between two levels of quality, and Avg_{Dist} represents how many classes away will the model prediction be from the correct one.

Table 9 summarizes these results for both classification and regression algorithms, listing the most effective algorithms for each number of quality levels, and shows how well performance improves when the number of labels decreases. We can determine that Support Vector Classifiers (svc_c) and Random Forest Regressors ($forest\_r$) are consistently good-performing models, independently of the used classes.

Table 9: Number of Classes: Most effective algorithms for each experiment

# **Classes**	**Best Models**	**Accuracy**	**MAE**	Avg_{Dist}
6	svc_c	64%	-	-
	$forest\_r$	-	0.0898	0.4490
5	svc_c	73%	-	-
	$forest\_r$	-	0.0878	0.3512
3	ada_c, $forest\_c$, svc_c	81%	-	-
	$forest\_r$	-	0.1333	0.2666
2	$forest\_c$, mlp_c, svc_c	91%	-	-
	$forest\_r$	-	0.1479	0.1479

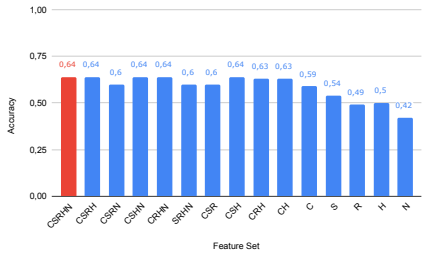

Fig. 5: **Support Vector Classifier**: Model accuracy

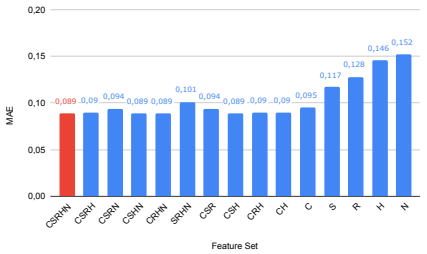

Fig. 6: **Random Forest Regressor**: Model MAE

4.3 Feature Category Importance

We observed that if a particular algorithm performs worse with a specific feature set, other algorithms will follow suit. As a result, we decided to focus this analysis solely on the *svc_ c* and *forest_ r* models, which are the best-performing classifier and regressor in every feature set.

Figures 5 and 6 show how the model performs when trained with different feature sets. It is unsurprising that using all features does not worsen performance, but we can quickly notice that not all categories are equally important. We can observe that content features have the best impact on the results, but keeping history features out also negatively affects performance. The charts also indicate that network features are not too crucial for accurately predicting an article's quality, which is fortunate considering their long computation times.

4.4 Multi-language Model Performance

As mentioned before, we constructed four new datasets, each respective to a different language. We then assessed the performances of three regression models: *CSRH6/forest_ r*, *CRH6/forest_ r*, and *CH6/forest_ r*. This experiment aims to simulate a real-time scenario, hence the usage of imbalanced datasets and regression models, which are better suited for predicting continuous values. Figure 7 shows that we can maintain the prediction error at an acceptable rate, even in other languages. We do not assess the MAE because these datasets are not balanced, so we want to give more weight to larger errors.

Additionally, we noticed that, on average, the model fares better against Portuguese articles. The relatively lower abundance of mid-tier Portuguese articles may explain that phenomenon, as it is more trivial to distinguish between articles that are much more distinct quality-wise. We can also observe that using fewer features significantly decreases model performance on non-English articles, which is surprising, considering that style and readability features are supposedly more accurate for English texts.

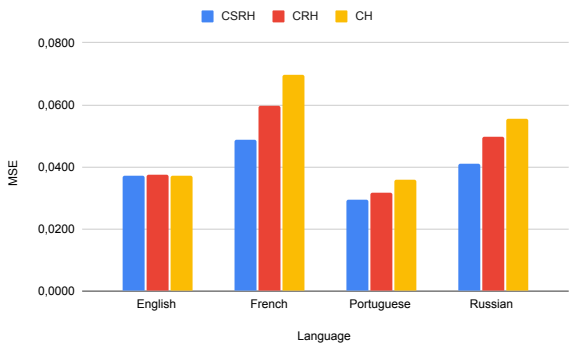

Fig. 7: Comparison between Mean Squared Error (MSE) of model predictions across different dataset languages

5 Discussion and implications of results

Wikipedia is a vast encyclopedia with a significant quality disparity between the millions of its articles, so it is difficult for users to determine the quality of the article they are reading efficiently. Envisioning the construction of a model that automatically predicts the quality of Wikipedia articles, we compared several algorithms with different sets of features in English articles and also in a multi-language scenario with English, French, Portuguese, and Russian articles.

We found that the best-performing algorithms for classification and regression were Support Vector Machines (SVM) and Random Forest, respectively (RQ1). The first achieved an accuracy of 64% and the second a Mean Absolute Error of 0.09. Our best-performing algorithms match or beat most state-of-the-art approaches, comparing our results with those we collected during the literature review.

As expected, decreasing the number of classes improved the model's accuracy (RQ2), but we considered the 5-class experiments to produce the most significant relative improvement, with an accuracy of 73%. Regardless, there does not seem to be an actual benefit to using less than six classes, as prediction misses only tend to occur on adjacent quality levels.

Regarding feature importance (RQ3), we found content features to be much more effective than the rest, but mixing them with the other categories led to a significant performance boost, especially history features. Additionally, we have shown that keeping the style and readability features improves performance for predicting non-English articles. We found that history features increase prediction time but considering them in a real-time scenario may still be justified by the performance gains. On the other hand, determining network features requires substantial computational effort for the benefit it generates, making them less suitable for real-time usage.

We also tested our model in randomly generated datasets of non-English Wikipedia articles (RQ4). All datasets showed promising results, though, on average, we can expect this solution to perform slightly worse in non-English Wikipedias. We have deposited the datasets used in the experiments and our results in a public repository. Finally, we provide this project's source code in a GitHub repository.

6 Conclusions

Wikipedia is a vast and evolving encyclopedia maintained by volunteers. We have successfully trained well-performing models for automatically predicting Wikipedia quality, designing a solution as good or better than state-of-the-art approaches.

Although this work has various benefits, it also comes with a few limitations. The main one, which is a common issue in many similar approaches, is that article features are simply not enough to measure quality. Downscaling entire articles into a few numbers guarantees the loss of critical information: The article's context, the integrity of the text, and the dimension of its theme are elements crucial to determining the quality and are virtually impossible to summarize for an approach like ours. It may be for this reason that these approaches never appear to reach perfect accuracy, seeming to peak around the 60%'s.

Moreover, even if we could make a perfect model of Wikipedia's quality scale, nothing guarantees that it would actually be a *good* measure of quality. That scale is maintained by many different editors across the most distinct WikiProjects, so it is heavily subject to human error. Editors are likely to unintentionally apply their biases to the evaluations, effectively leading to varying standards for the same level of quality. Additionally, we assume that the quality assessments are up-to-date with the latest changes to the article, which is not guaranteed.

We consider this study has potential for future research on a few paths. The set of features could be further tuned by assessing their importance individually instead of grouping them in categories. For instance, some style features are probably much more relevant than others, so upon further experiment, that category could see a severe reduction without significantly affecting model performance. Such an analysis could also help mitigate the model overfit we encountered. Authors may also wish to experiment with additional features or algorithms (e.g., deep learning), to try to further improve our results.

Acknowledgments. This work is funded by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., under the support UID/50014/2023 (https://doi.org/10.54499/UID/50014/2023).

References

1. Bellomi, F., Bonato, R.: Network Analysis for Wikipedia. In: Proceedings of Wikimania 2005. Frankfurt, Germany (2005)

2. Bergstra, J., Bengio, Y.: Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research **13**, 281–305 (2012), https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf
3. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly Media, Inc., 1st edn. (2009)
4. Bird, S., Klein, E., Loper, E.: NLTK: Natural Language Toolkit. https://www.nltk.org/ (2025), accessed: 2025-06-26
5. Boldi, P., Vigna, S.: The webgraph framework i: compression techniques. In: The 2004 World Wide Web Conference (WWW04). p. 602. Association for Computing Machinery (ACM), New York, United States (2004). https://doi.org/10.1145/988672.988752
6. Boldi, P., Vigna, S.: Laboratory for Web Algorithmics - Datasets. https://law.di.unimi.it/datasets.php (2023), accessed: 2023-12-12
7. Coleman, M., Liau, T.L.: A computer readability formula designed for machine scoring. Journal of Applied Psychology **60**, 283 (4 1975). https://doi.org/10.1037/H0076540
8. Contreras, A., García-Alonso, R., Echenique, M., Daye-Contreras, F.: The SOL Formulas for Converting SMOG Readability Scores Between Health Education Materials Written in Spanish, English, and French. Journal of Health Communication **4**, 21–29 (2010). https://doi.org/10.1080/108107399127066
9. Dale, E., Chall, J.S.: A Formula for Predicting Readability: Instructions. Educational Research Bulletin **27**, 37–54 (1948)
10. Jo, J.M.: Effectiveness of Normalization Pre-Processing of Big Data to the Machine Learning Performance. Journal of the KIECS **14**, 547–552 (2019). https://doi.org/10.13067/JKIECS.2019.14.3.547
11. Kincaid, J.P., Fishburne, R.P., Rogers, R.L., Chissom, B.S.: Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel. Tech. rep., Institute for Simulation and Training, University of Central Florida (1975)
12. scikit learn: Scikit-Learn: Machine Learning in Python. https://scikit-learn.org/stable/ (2025), accessed: 2025-06-26
13. Lee, Y.W., Strong, D.M., Kahn, B.K., Wang, R.Y.: AIMQ: a methodology for information quality assessment. Information & Management **40**, 133–146 (2002), https://www.sciencedirect.com/science/article/abs/pii/S0378720602000435
14. Medar, R., Rajpurohit, V.S., Rashmi, B.: Impact of training and testing data splits on accuracy of time series forecasting in machine learning. In: 2017 International Conference on Computing, Communication, Control and Automation, ICCUBEA 2017. pp. 1–6. Institute of Electrical and Electronics Engineers Inc., Pune, India (9 2018). https://doi.org/10.1109/ICCUBEA.2017.8463779
15. Moás, P.M., Lopes, C.T.: Automatic Quality Assessment of Wikipedia Articles — A Systematic Literature Review. ACM Comput. Surv. **56**(4) (nov 2023). https://doi.org/10.1145/3625286
16. Moás, P.M., Lopes, C.T.: Data of real-time prediction of Wikipedia articles' quality [dataset]. INESC TEC (2022). https://doi.org/10.25747/2rdd-rc08
17. Roy, D., Bhatia, S., Jain, P.: Information asymmetry in Wikipedia across different languages: A statistical analysis. Journal of the Association for Information Science and Technology **73**, 347–361 (3 2022). https://doi.org/10.1002/ASI.24553
18. Senter, R., Smith, E.A.: Automated readability index. Tech. rep., Cincinnati Univ OH (1967)

19. Shalabi, L.A., Shaaban, Z., Kasasbeh, B.: Data mining: A preprocessing engine. Journal of Computer Science **2**, 735–739 (9 2006). https://doi.org/10.3844/JCSSP.2006.735.739
20. Singh, V., Pencina, M., Einstein, A.J., Liang, J.X., Berman, D.S., Slomka, P.: Impact of train/test sample regimen on performance estimate stability of machine learning in cardiovascular imaging. Scientific Reports 2021 11:1 **11**, 1–8 (7 2021). https://doi.org/10.1038/s41598-021-93651-5
21. Teblunthuis, N.: Measuring wikipedia article quality in one dimension by extending ores with ordinal regression. In: OpenSym 2021: 17th International Symposium on Open Collaboration. pp. 1–10. Association for Computing Machinery (ACM), Online, Spain (9 2021). https://doi.org/10.1145/3479986.3479991
22. Wikimedia: Wikistats - statistics for wikimedia projects. https://stats.wikimedia.org (2025), accessed: 2025-06-26
23. Wikipedia: Wikipédia:versão 1.0/avaliação. https://pt.wikipedia.org/wiki/Wikip%C3%A9dia:Vers%C3%A3o_1.0/Avalia%C3%A7%C3%A3o (2018), accessed: 2025-06-26
24. Wikipedia: Rossija/ocenki - vikipedija. https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%A1%D0%BE%D0%B2%D0%B5%D1%82_%D0%B2%D0%B8%D0%BA%D0%B8-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%BE%D0%B2/%D0%A7%D0%B0%D0%92%D0%9E_%D0%BF%D0%BE_%D0%BE%D1%86%D0%B5%D0%BD%D0%BA%D0%B5_%D1%81%D1%82%D0%B0%D1%82%D0%B5%D0%B9 (2024), accessed: 2025-06-26
25. Wikipedia: List of wikipedias. https://meta.wikimedia.org/wiki/List_of_Wikipedias (2025), accessed: 2025-06-26
26. Wikipedia: Projet:Évaluation. https://fr.wikipedia.org/wiki/Projet:%C3%89valuation (2025), accessed: 2025-06-26
27. Wikipedia: Wikipedia: Content assessment. https://en.wikipedia.org/wiki/Wikipedia:Content_assessment (2025), accessed: 2025-06-26
28. Wikipedia: Wikipedia: Pages with the most revisions. https://en.wikipedia.org/wiki/Wikipedia:Database_reports/Pages_with_the_most_revisions (2025), accessed: 2025-06-26
29. Wikipedia: Wikipedia: Size of wikipedia. https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia (2025), accessed: 2025-06-26
30. Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. Climate Research **30**, 79–82 (2005). https://doi.org/10.2307/24869236