Cultural Classification: Agnostic, Representative, Exclusive

Christian Bianchi and Kevin Giannandrea and Leonardo Mariut
Sapienza University of Rome
{bianchi.1984056,giannandrea.2202212,mariut.1986191}@studenti.uniroma1.it

1 Introduction

We address the classification of cultural items into three categories: agnostic, representative, and exclusive. Our approach combines a transformer-based model enriched with external knowledge from Wikipedia and Wikidata, and a non-language-model baseline. This report describes our methodology, design decisions, and evaluation strategy.

2 Methodology

This section outlines the features employed, the extraction process, and the proposed classification architecture. We employ both statistical and categorical features extracted from the dataset.

2.1 Dataset Exploration

We begin our analysis by exploring the dataset consisting of approximately 6 000 samples, each annotated with one of three class labels: cultural agnostic, cultural representative, and cultural exclusive. We observed a significant class imbalance, as illustrated in fig. 1. Notably, the culturally exclusive category is overrepresented compared to the others.

We then examine the feature type, which, as shown in fig. 2, provide strong discriminative power for distinguishing between the agnostic and exclusive labels. Additionally, the feature category also exhibits meaningful signal in differentiating between the classes (fig. 3).

2.2 Feature Extraction

Using the Wikidata API (Wikimedia Foundation), we explored the features available for each item and analyzed their distributions with respect to the target labels. We focused on several indicators, such as the number of sitelinks (fig. 4), the total number of claims associated with an item (fig. 5), and the presence of specific claims, such as property P17 "Country" (fig 6). These features exhibit varying degrees of correlation with the classification labels.

To enrich the feature set, we computed additional derived metrics. One of the derived features is the translation entropy of sitelinks titles (fig. 7), which measures the diversity of titles used for a given item across different language editions of Wikipedia. A low entropy indicates that most language versions use the same or very similar title, suggesting limited cultural reinterpretation or localization, which can be typical of culturally exclusive items. In contrast, an higher entropy suggests greater variation in how the concept is titled across languages, potentially indicating more culturally agnostic content. Another derived feature is the sum of culturally relevant claims, computed as the total number of claims linked to properties we deemed culture-specific (e.g., country, language, ethnic group). This aggregated value aims to capture the item's cultural specificity. We further enriched our feature set by scraping the corresponding Wikipedia pages using the `beautifulsoup4` library (Richardson, 2023). From HTML content, we extracted additional attributes such as page length - the total number of characters in the page text (fig. 8) - and number links - the count of internal Wikipedia hyperlinks (fig. 9). These features capture the depth and connectivity of an article: longer pages and those with more internal references may reflect broader context correlating with the cultural generality or specificity of the item. Sitelink-based statistical features (mean, median, and standard deviation) were extracted by first identifying the English Wikipedia page associated with a given Wikidata item. Wikidata itself (Vrandečić and Krötzsch, 2014) provides structured knowledge that enables rich cultural metadata extraction. From that page, all inline links were collected, and each link was resolved to its corresponding Wikidata item. The statistics were then computed over the set of linked Wikidata items sitelinks.

2.3 Model Architecture

Initially, we explored a graph-based approach, where graphs were constructed from stopwords, and their embeddings were passed to a neural network, achieving approximately 65% accuracy on the development set.

Next, we used a stacking approach combining XGBoost and a neural network with three hidden layers. In this model, XGBoost was trained on the features `category`, `translation_entropy`, `type`, and `sum_cultural_claims` to generate class probabilities. These probabilities were concatenated with `page_length` and `num_links` and passed as input to the neural network, resulting in 76% accuracy on the development set.

The third approach involved using only XGBoost, trained on all available features, which achieved 77% accuracy on the development set. This model was chosen as the final architecture due to its superior performance and relative simplicity.

Hyperparameter tuning was performed via grid search. The selected parameters were:

- `colsample_bytree` = 0.8
- `learning_rate` = 0.03
- `max_depth` = 6
- `min_child_weight` = 1
- `n_estimators` = 270
- `subsample` = 0.8

3 Experiments

Our non-LLM model began with two concept-graph strategies: a single global graph or individual per-item graphs. We chose the latter for simplicity. The global-graph idea promised rich, cross-item queries (e.g. measuring cultural or geographic ties) but was ruled out due to its probably huge memory footprint, slow queries, and increased maintenance overhead. With per-item graphs, our first "naive" approach parsed English Wikipedia pages: we tokenized section titles and descriptions, linked words sequentially, and merged repeated tokens nodes across sections. The preprocessing takes under an hour for the full dataset, but using raw tokens meant not merging similar nodes. Graph embeddings (via *graph2vec* (Narayanan et al., 2017)) and topological features reached only 58% accuracy, so this approach has been discarded. In the second approach, we refined concept extraction by leveraging Wikidata links within Wikipedia pages. Instead of raw words, we used Wikidata QIDs—unique identifiers for concepts, ensuring language-agnostic representation (with the help of QID). Each graph began with the item's english title, followed by section nodes and linked QIDs (weighted by sitelink count). This method was promising, particularly in capturing cultural specificity. For instance, the Italian Wikipedia page for pasta alla gricia referenced concepts like "cucina laziale" (Roman cuisine) and "primo piatto" (first-course dish), which appeared in few other languages, thus hinting at strong cultural exclusivity. However, scalability became an issue. Initially, we planned to build graphs using all available language versions of a Wikipedia page. This was unfeasible, as a compromise, we restricted graph construction just to english Wikipedia pages. Unfortunately this issue persisted even with this limitation and the final graph-based approach—using multi-language or english only sitelinks and recursively expanding linked items—could not be completed in time due to the sheer volume of requests, API calls and processing required. Meanwhile, we tested simpler feature augmentation, mimicking some possible graph features (e.g., English page length, linked-item counts, and sitelink statistics) to boost the final model performance. This data has been already explained above. Data plotting revealed interesting results (Fig.4). Finally, neither graph-based method made the final model: the first was too simplistic, the second too time-consuming, and only the augmented data was helpful.

4 Results

Table 2 reports the comparative performance of all evaluated models. The hyperparameter-tuned XGBoost classifier achieves the highest accuracy in the dev set (fig. 10), surpassing both the stacking-based approach and the graph-based method. We also trained several transformer-based classifiers, fine-tuning BERT, RoBERTa, and DeBERTa-v3 on the item name and description, enriched with metadata (category and type). The best-performing model was BERT-base, which achieved 77.3% accuracy and 75.7% macro F1 on the dev set. DeBERTa-v3 performed slightly better on culturally exclusive items, while RoBERTa offered a small boost in macro F1. Overall, LM-based classifiers matched or slightly exceeded the performance of feature-based models, particularly on agnostic items with rich textual descriptions.

References

Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. 2017. graph2vec: Learning distributed representations of graphs. In *Proceedings of the Workshop on Mining and Learning with Graphs (MLG)*.

Leonard Richardson. 2023. Beautiful soup documentation. Version 4.12, accessed 2025-05.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57(10):78–85.

Wikimedia Foundation. Wikidata api. Accessed 2025-05.

A Tables

Config	USE_GRAPH	USE_EMBED	USE_SITELINK	Accuracy ↑	Precision ↑	Recall ↑	F1-score ↑
Sitelinks only	False	False	True	**0.6522**	**0.6524**	**0.6462**	**0.6444**
Graph only	True	False	False	0.5819	0.5816	0.5819	0.5779
Embeddings only	False	True	False	0.5351	0.5367	0.5439	0.5397
Graph + Embeddings	True	True	False	0.5853	0.5863	0.5858	0.5811
Graph + Sitelinks	True	False	True	0.5786	0.5790	0.5818	0.5765

Table 1: Performance of a 3-layer DNN trained on different combinations of experimental features. `USE_GRAPH` includes mesoscale and topological properties, `USE_EMBED` uses Graph2Vec embeddings (Narayanan et al., 2017), and `USE_SITELINK` includes sitelink count and translation entropy features. Arrows indicate that higher values are better.

Model	Accuracy ↑	F1-score ↑	Precision ↑
XGBoost (Final)	**0.77**	**0.76**	**0.76**
Stacking (XGB + NN)	0.76	0.76	0.76
DeBERTa-v3 (LM-based)	0.77	0.755	0.76
RoBERTa (LM-based)	0.77	0.758	0.76
BERT-base (LM-based)	0.773	0.757	0.77
Graph-based (DNN)	0.58	0.58	0.58

Table 2: Accuracy and macro-averaged F1 and precision across all models. Transformer-based classifiers (e.g., DeBERTa-v3, RoBERTa, BERT) perform on par with XGBoost, with BERT slightly outperforming others in accuracy. Graph-based models underperform due to scalability limitations and limited semantic abstraction.

B Images

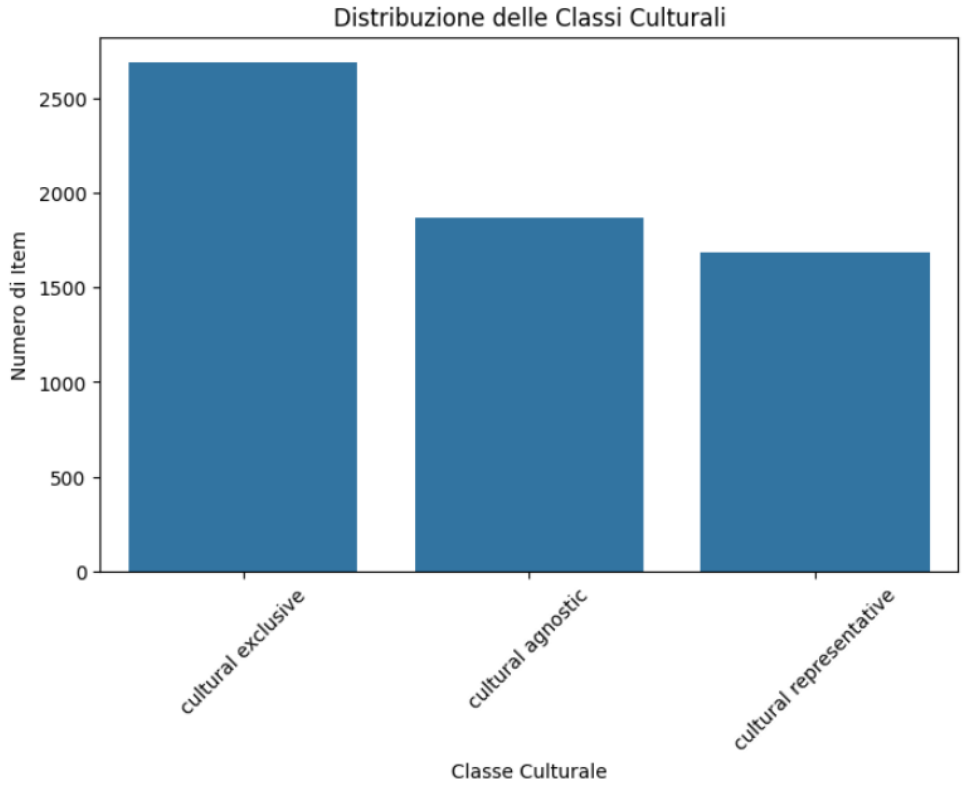

Figure 1: Class distribution across the dataset.

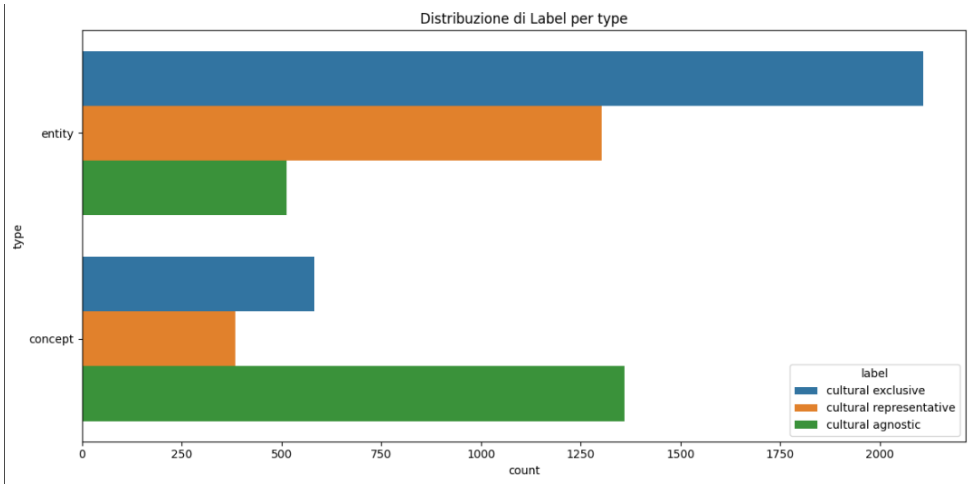

Figure 2: Distribution of the `type` feature across class labels.

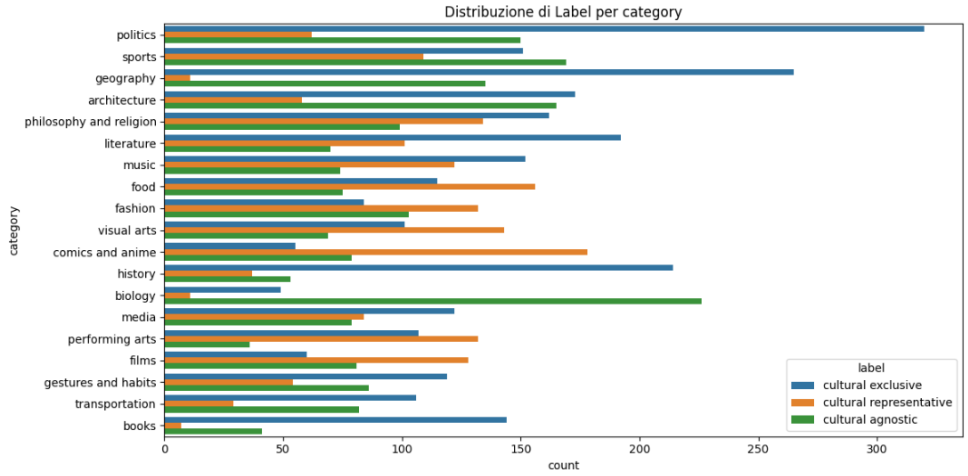

Figure 3: Distribution of the `category` feature across class labels.

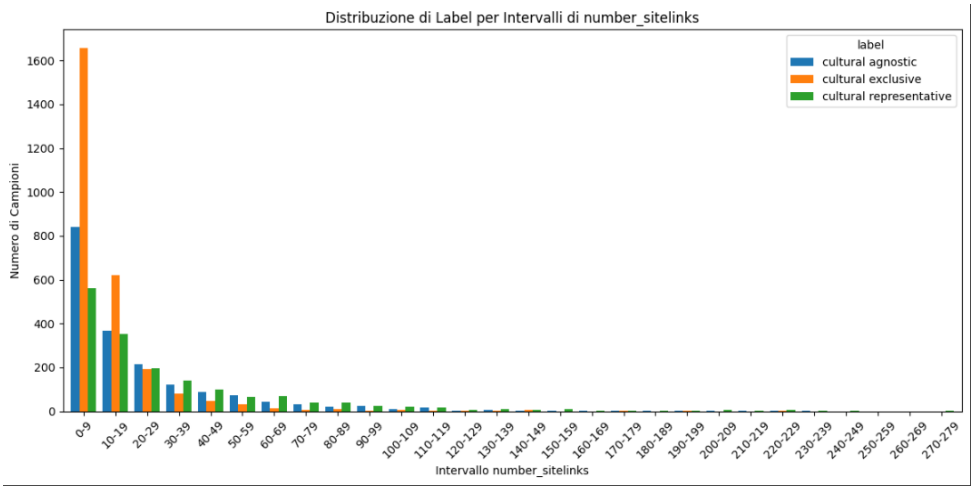

Figure 4: Distribution of the `number sitelinks` feature across class labels.

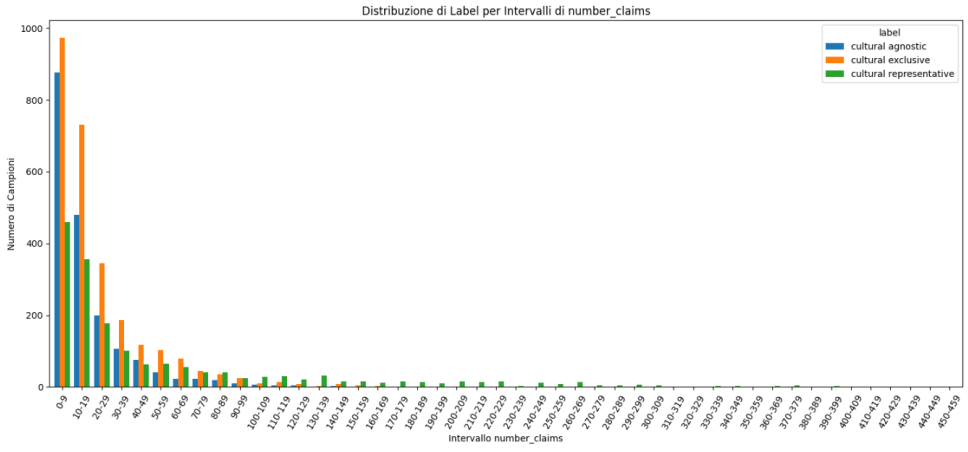

Figure 5: Distribution of the `number claims` feature across class labels.

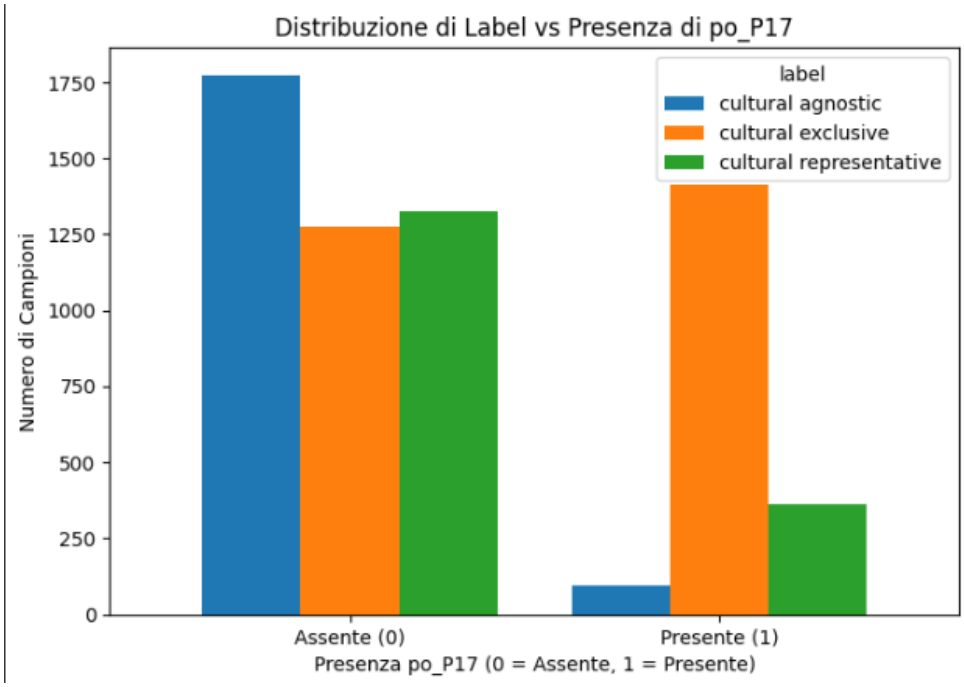

Figure 6: Distribution of the `presence of P17` feature across class labels.

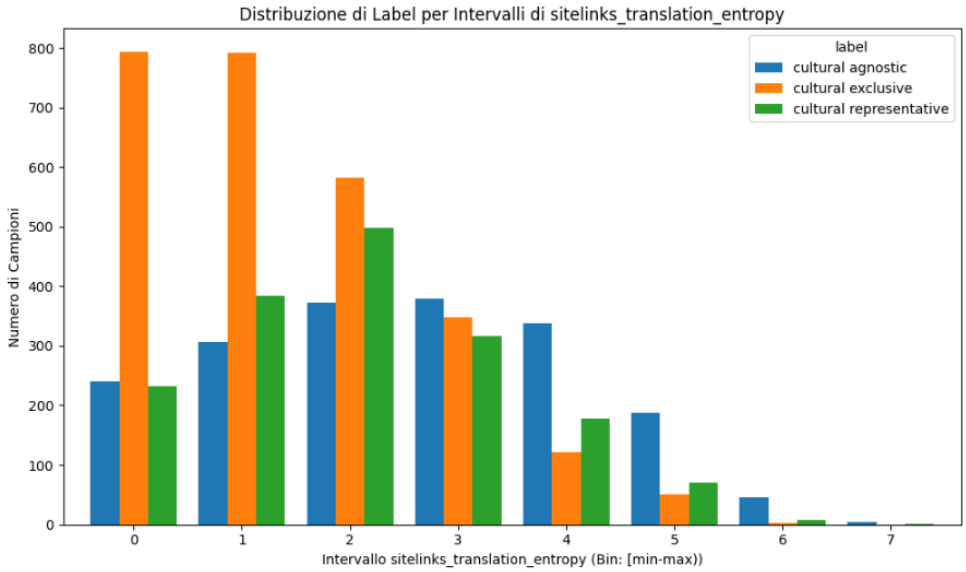

Figure 7: Distribution of the `sitelinks title translation entropy.` feature across class labels

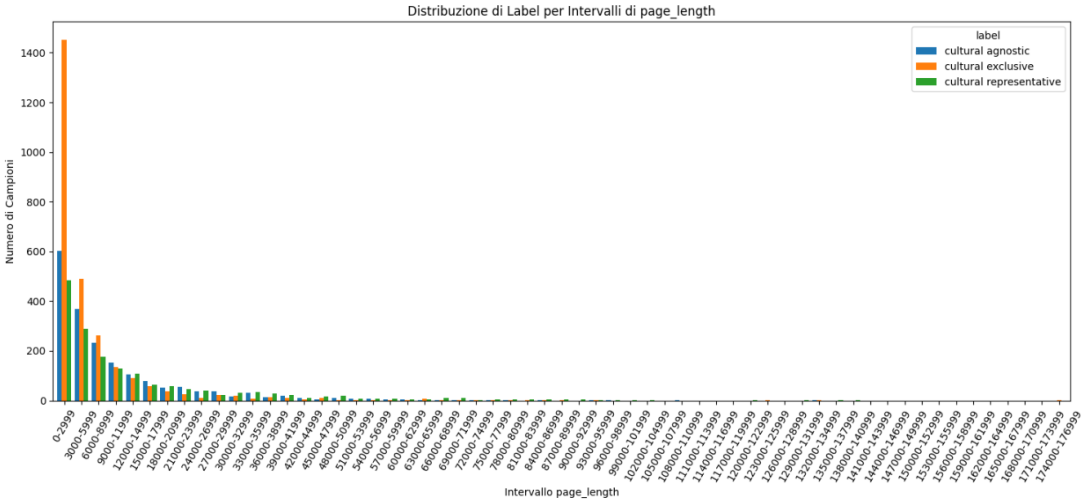

Figure 8: Distribution of the `page length` feature across class labels.

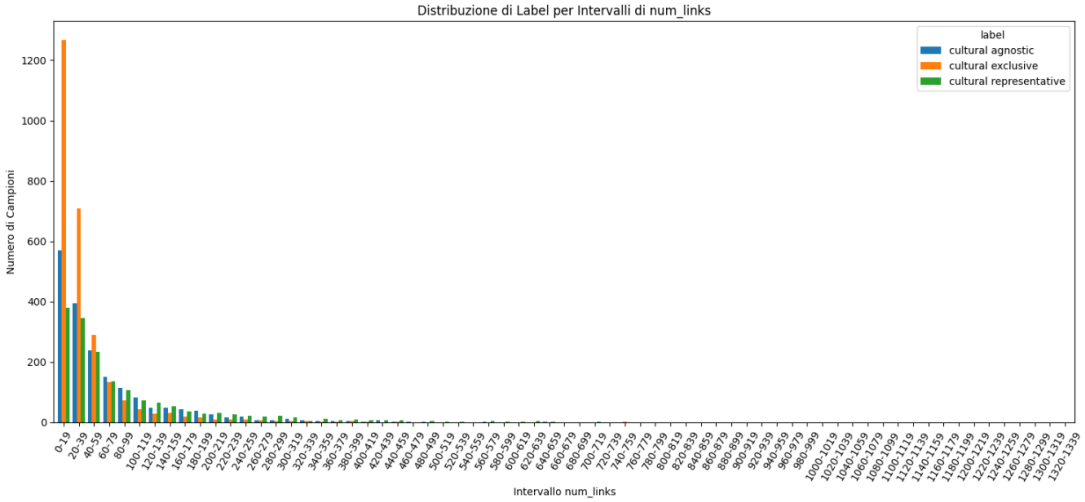

Figure 9: Distribution of the `number links` feature across class labels.

	precision	recall	f1-score	support
cultural agnostic	0.78	0.92	0.85	117
cultural representative	0.79	0.62	0.69	107
cultural exclusive	0.73	0.75	0.74	76
accuracy			0.77	300
macro avg	0.77	0.76	0.76	300
weighted avg	0.77	0.77	0.76	300

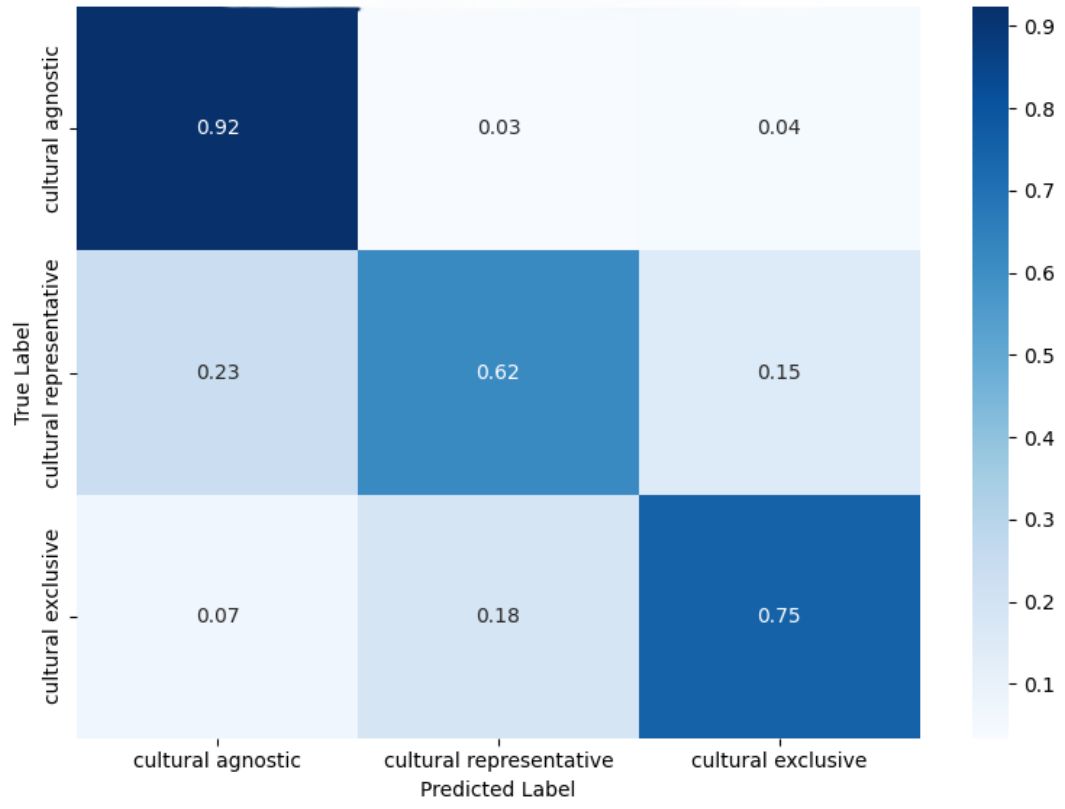

Figure 10: Performance of the `XGBoost` model computed in the dev set.