big data and cognitive computing

MDPI

Article

Application of Natural Language Processing and Genetic Algorithm to Fine-Tune Hyperparameters of Classifiers for Economic Activities Analysis

Ivan Malashin *, Igor Masich, Vadim Tynchenko *, Vladimir Nelyub, Aleksei Borodulin and Andrei Gantimurov

Artificial Intelligence Technology Scientific and Education Center, Bauman Moscow State Technical University, Moscow 105005, Russia; is.masich@gmail.com (I.M.)

* Correspondence: i.malashin@emtc.ru (I.M.); vadimond@mail.ru or vtynchenko@emtc.ru (V.T.); Tel.: +7-926-875-7128 (I.M.)

Abstract: This study proposes a method for classifying economic activity descriptors to match Nomenclature of Economic Activities (NACE) codes, employing a blend of machine learning techniques and expert evaluation. By leveraging natural language processing (NLP) methods to vectorize activity descriptors and utilizing genetic algorithm (GA) optimization to fine-tune hyperparameters in multi-class classifiers like Naive Bayes, Decision Trees, Random Forests, and Multilayer Perceptrons, our aim is to boost the accuracy and reliability of an economic classification system. This system faces challenges due to the absence of precise target labels in the dataset. Hence, it is essential to initially check the accuracy of utilized methods based on expert evaluations using a small dataset before generalizing to a larger one.

Keywords: NLP; multiclass classification; subgroups; NACE; economic activities; big data; expert evaluation

Citation: Malashin, I.; Masich, I.; Tynchenko, V.; Nelyub, V.; Borodulin, A.; Gantimurov, A. Application of Natural Language Processing and Genetic Algorithm to Fine-Tune Hyperparameters of Classifiers for Economic Activities Analysis. *Big Data Cogn. Comput.* **2024**, *8*, 68. https://doi.org/10.3390/bdcc8060068

Academic Editors: Sadok Ben Yahia, Amnir Hadachi and Jenq-Shiou Leu

Received: 28 April 2024
Revised: 1 June 2024
Accepted: 11 June 2024
Published: 13 June 2024

1. Introduction

Nomenclature of Economic Activities (NACE) is a European standard classification system used to categorize economic activities [1]. NACE has become instrumental in standardizing economic data collection and analysis across European Member States since its inception in the 1970s. It serves as the foundation for gathering and processing data on economic activities, enabling cross-national and pan-European comparisons. It serves as a common framework for statistical analysis and reporting within the European Union (EU) and its member states. Each economic activity is assigned a unique NACE code, facilitating consistency and comparability across different industries and regions [2]. The accurate classification of economic activities is crucial for various purposes, including economic policy formulation [3], resource allocation [4], market analysis [5], and international trade [6].

Given the significance of NACE codes in understanding economic trends and facilitating decision-making processes, ensuring the accuracy of these codes is paramount. However, discrepancies between the assigned NACE codes and the actual nature of economic activities can occur, leading to misclassification and potential distortions in data analysis and policy development.

The application of ML techniques in classifying economic activities is essential, as evidenced by its widespread adoption in the literature.

The EU's monthly harmonized Short-term Business Statistics (STS) is pivotal for assessing the European economy. The timeliness of STS is crucial for policymakers to respond effectively to economic shifts. The paper [7] evaluates different machine learning algorithms to enhance the timeliness and granularity of Austrian STS data via early estimations of missing survey data. While a multivariate time series model is currently used for total

industry and construction, adaptations could extend its applicability to NACE divisions, except for divisions with small populations, suggesting the need for alternative methods in such cases.

The study [8] introduces a method to classify European Cleantech companies using supervised machine learning on business descriptions. By training the model on labeled data, it learns to identify Cleantech firms based on terms like "sustainable" and "waste management", improving NACE-based classification accuracy.

Analyzing web job vacancies offers advantages over traditional survey-based methods, facilitating faster decision-making based on factual data. The paper [9] presents an automated approach for classifying millions of web job vacancies into standard occupational categories using machine learning techniques, thereby enhancing the understanding of labor market demands across different countries, with insights aligned with the NACE taxonomy.

Paper [10] elucidates the Land Use/Cover Area frame Survey (LUCAS) methodology, which is crucial for gathering comprehensive in situ data on land cover and use in the EU, aligning with NACE classification standards. The resulting harmonized database facilitates robust geo-spatial and statistical analyses of land surface changes, with potential applications in multi-temporal assessments and deep learning advancements

Work [11] addresses the impact of Industry 4.0 on the labor market, particularly within the Facility Service (FS) industry, and emphasizes the importance of assessing FS employment trends aligned with the European Norm for Facility Management and NACE classifications. Despite challenges in defining FS activities, the study provides insights into FS employment in major European economies, informing future strategies amidst ongoing transformations in the labor market.

The research addresses the challenge of classifying texts of descriptions of economic activities in coherence with the NACE classification. Economic activity information includes not only textual descriptions but also prices, completion deadlines, and the already proposed NACE product classification codes, which are designed to categorize activities by economic types. However, discrepancies between the proposed codes and the predicted NACE code may arise due to user incompetence or intentional misrepresentation. It posing a challenge in accurately classifying activities.

Handling this problem involves using text preprocessing algorithms through natural language processing (NLP) techniques to vectorize texts, fine-tuning hyperparameters of various classifiers algorithms [12–14] with Genetic Algorithms (GA), subgroup discovery algorithms [15], and statistical analysis methods to evaluate the effectiveness (accuracy) of the developed models. The complexity lies in the absence of accurate target labels in the training data, which are essential for classification methods and model training and tuning.

We aim to develop an approach capable of accurately classifying activities based on their textual descriptions, thereby enhancing the reliability and utility of NACE-coded data for economic analysis and decision-making.

2. Materials and Methods

2.1. Dataset Description

To address the research objectives, we leverage a dataset comprising 20 million records of economic activities. These records encapsulate information such as activity descriptions, prices, completion deadlines, and NACE product classification codes. The utilization of NACE codes enables the categorization of activities based on economic activity types, facilitating an analysis of the government procurement landscape.

The visualization of the distribution of activities across the 88 divisions, identified by two-digit numerical codes (01 to 99) within the NACE classification system, is shown in Figure 1. This diagram shows the prevalence of economic activities across various economic sectors, shedding light on the dynamics of procurement activities within each division. Table A1 provide descriptions of the NACE codes corresponding to the statistics shown in Figure 1.

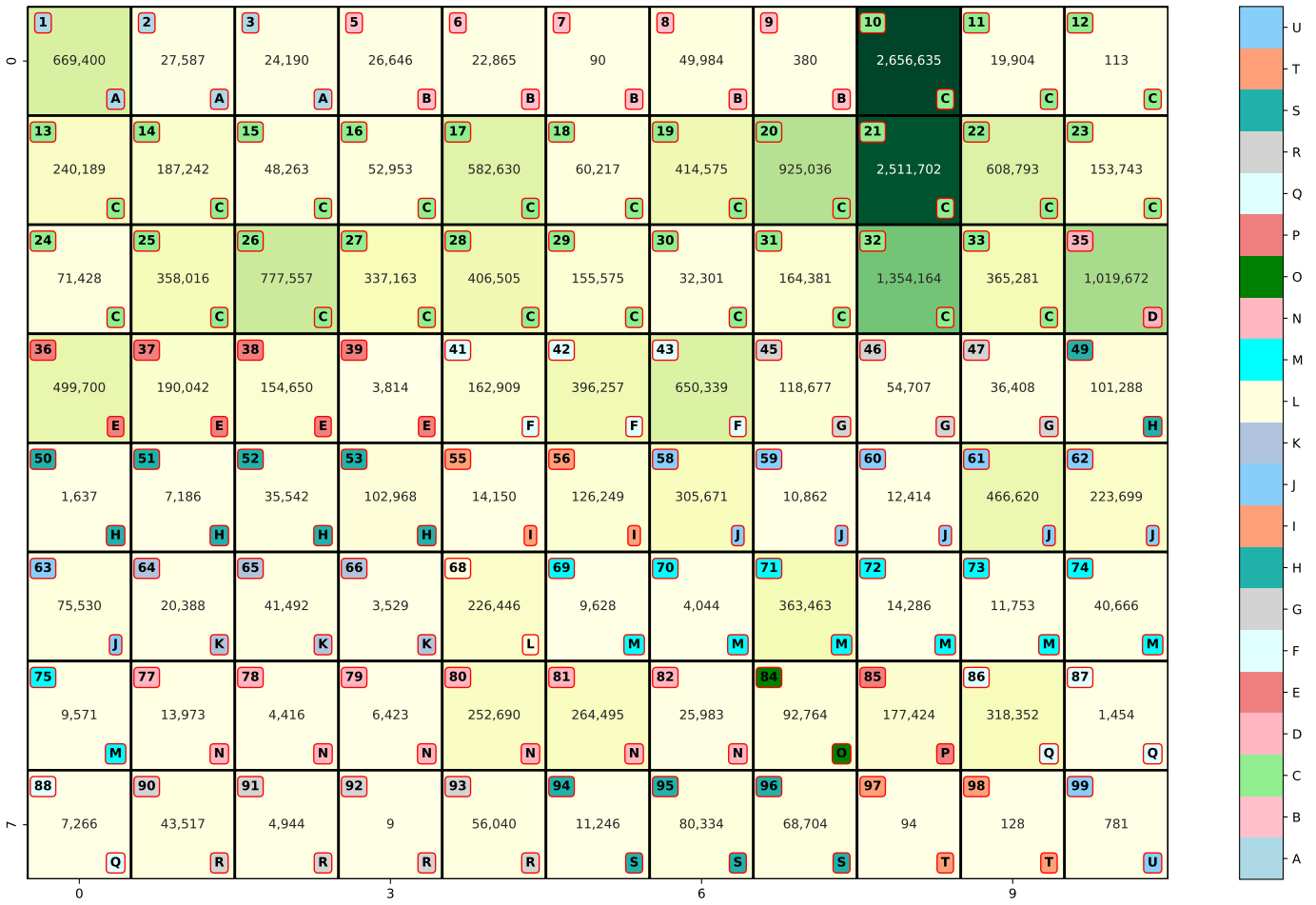

Figure 1. Visual The representation of economic activities' distribution according to NACE codes: each cell of the diagram represents one of 88 divisions of NACE codes (1–99), grouped into 21 sections, identified by alphabetical letters from A to U, with the number of corresponding activities depicted in the center of each cell.

For the efficient processing and analysis of text descriptors at scale, we employed Apache Spark, a high-performance framework for distributed data processing [16–19]. The approach involved vectorizing descriptor texts using the Term Frequency–Inverse Document Frequency (TF-IDF) method, which assesses the importance of each word in the context of the entire text corpus.

The vectorization process includes the loading of the initial data into Apache Spark in DataFrame format. Then, each text was tokenized into individual words using a tokenizer. To convert words into vectors, we utilized the CountVectorizer and TfidfVectorized methods, which assigns a unique numerical identifier to each word and converts it into a fixed-length vector. TF-IDF vectors were computed for each text using Inverse Document Frequency (IDF). This method considers the frequency of a word's occurrence in a document and its inverse frequency across the entire corpus. The obtained TF-IDF vectors were utilized for the analysis and classification of text data, enabling the identification of important patterns and trends within the data.

Figure 2 illustrates of the experimental pipeline of the article, featuring the processing of large text data using Apache Spark, text vectorization via NLP, and classification of vectorized texts using various multi-class classifiers. Hyperparameter tuning of the classifier was performed using a genetic algorithm, with the objective function being classification accuracy. Expert assessment of a portion of the data was utilized to gauge accuracy, which may be inaccurate due to the absence of correct labels. Ultimately, we obtained a model for the improved classification of descriptors of economic activities.

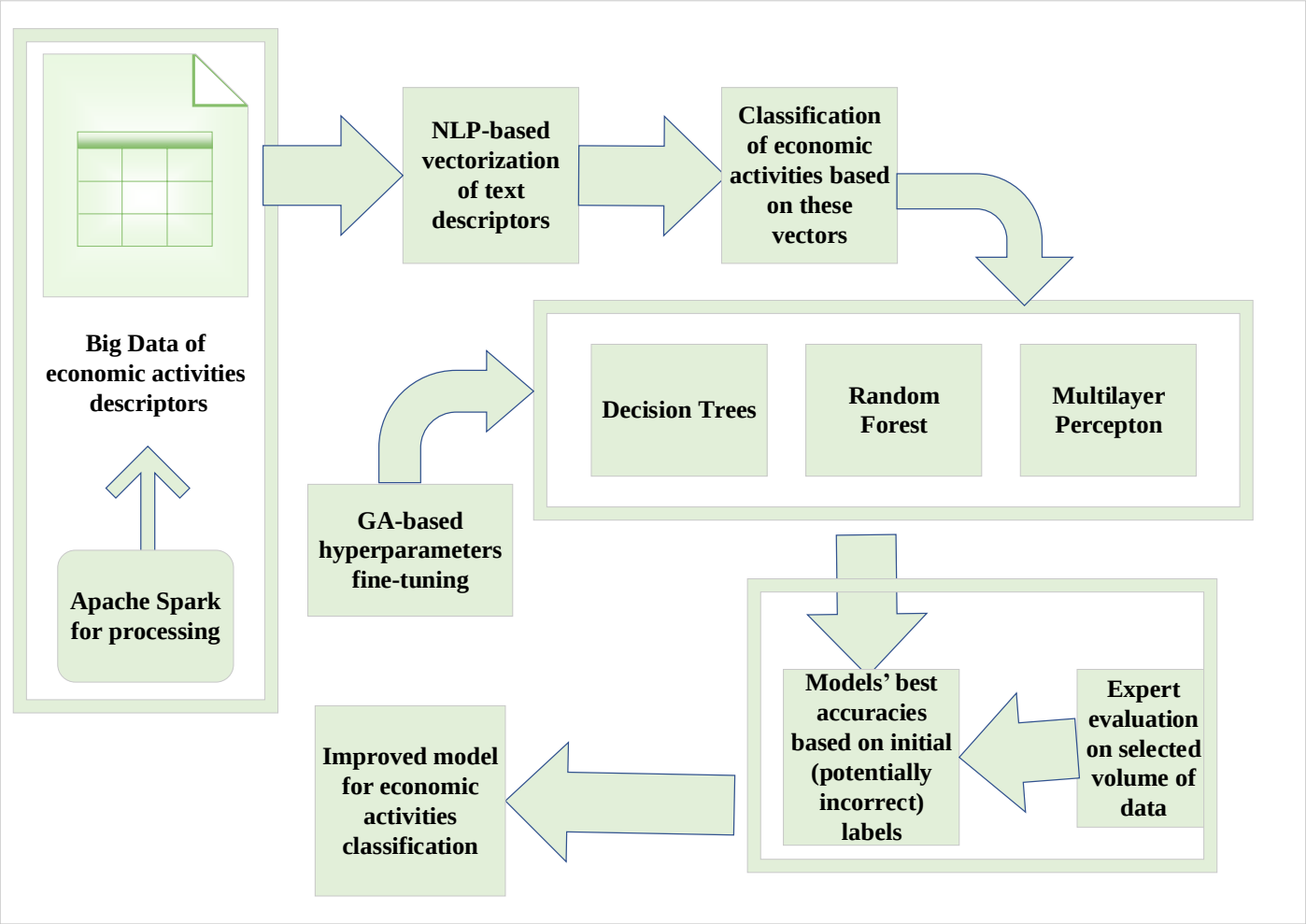

Figure 2. Experimental pipeline for economic activity descriptors' classification.

2.2. Natural Language Processing

NLP [20,21] is a broad field of artificial intelligence and computational linguistics that studies the problems of the computer analysis and synthesis of natural languages. It deals with converting human language text into a format suitable for further computational processing [22]. Text preprocessing [23] is essential for machine learning algorithms to work with natural language text.

Typically, the first step in text processing is normalization [24]. This operation involves converting texts to a desired case, removing punctuation marks [25] (usually achieved by deleting specified characters from the text), removing numbers [26] (or converting them to a different format), and eliminating whitespace characters [27]. Normalization is necessary to standardize text processing methods. Basic transformations were performed on contract descriptions at this stage, including removing special characters, single literals, multiple spaces, geographical names, and abbreviations. Subsequently, all characters were converted to lowercase for text standardization [28] and ease of further processing. Tokenization [29], which involves splitting long strings into shorter ones, is the next step. This process allows for complex textual data to be transformed into a simpler and more manageable form.

The next step is stemming [30]. The variety of correct word forms, which have similar meanings but different spellings, with suffixes, prefixes, endings, etc., complicates the creation of dictionaries and further processing. Stemming allows words to be reduced to their basic form. The essence of this approach lies in finding the base of the word, achieved by sequentially removing its parts from the end and beginning. Stemming rules are predefined, often represented by regular expressions. Lemmatization [31] is

an alternative to stemming, aiming to bring words to their dictionary form (lemma). Lemmatization considers the morphological characteristics of words, unlike stemming, which operates without context knowledge and does not differentiate between words with different meanings based on their parts of speech. However, stemming has advantages: it is easier to implement and faster [32].

Lemmatization and stemming are often used together for text processing [33]. Lemmatization reduces words to their canonical form, taking into account their morphological features, while stemming removes word endings, bringing them to the root form. This can improve text processing by reducing the number of word forms and simplifying analysis. WordNetLemmatizer [34] and Stemmer from the nltk.stem module of the NLTK library were used for lemmatization and stemming. Stemmer is specifically designed for processing text and can effectively shorten words to their root (stem), which is useful for reducing the morphological diversity of the text. This additionally reduced the volume of textual data and simplified the analysis.

The final step is vectorization [35]. Most mathematical models operate in high-dimensional vector spaces, so it is necessary to map text into a vector space. For example, the `bag-of-words` model allows for a a document to be represented as a multidimensional vector of words and their weights in the document [36]. In other words, each document is a vector in a multidimensional space, where the coordinates correspond to the word numbers, and the values of the coordinates correspond to the values of the weights. Another popular vectorization model is Word2vec [37], which represents each word as a vector containing information about contextually associated words. Another common vectorization model is TF-IDF [38,39].

The computational complexity of various classification methods depends directly on the dimensionality of the feature space. Therefore, to effectively train classifiers, it is often necessary to reduce the number of features (terms) used [40]. By reducing the dimensionality of the term space, it is possible to reduce the overfitting effect—a phenomenon where the classifier relies on random or erroneous characteristics of the training data rather than important and significant ones [41]. An overfitted classifier performs well on the instances it was trained on but much worse on test data. To avoid overfitting, the number of training examples should be proportional to the number of terms used. In some cases, reducing the dimensionality of the feature space by a factor of 10 (or even 100) may result in only a slight degradation of classifier performance.

There are several ways to determine the weight of document features. The most common method is to compute the TF-IDF function [42]. Its main idea is to assign greater weight to words with a high frequency within a specific document and a low frequency across other documents. The term frequency (TF) is calculated as the ratio of the number of occurrences of a word in a document to the total number of words in the document. The inverse document frequency (IDF) is the inverse of the frequency with which a word occurs in the collection of documents. IDF reduces the weight of common words. The final weight of the term in the document relative to the entire document collection is calculated by TF * IDF.

It should be noted that the formula evaluates the significance of a term solely based on its frequency of occurrence in the document, without considering the order of terms in the document and their lexical collocations.

To transform text into numerical vectors, two main vectorization methods were selected and compared: CountVectorizer [43] and TfidfVectorizer [44]. CountVectorizer employs the bag-of-words method, relying on the frequency of word occurrences in the document. TfidfVectorizer is used to assess the importance of a word in the context of the entire corpus of texts based on the TF-IDF approach. The parameters used for both models are shown in Table 1.

Table 1. Parameters for CountVectorizer and TfidfVectorizer.

Parameter	Description	CountVectorizer	TfidfVectorizer
`max_features`	Max. number of most frequently occurring words in vectorization	1500	1500
`min_df`	Min. document frequency for a word to be included in analysis	5	5
`max_df`	Max. document frequency for a word to be excluded from analysis	0.7	0.7

2.3. Optimizing Classifier Parameters Using a Genetic Algorithm

The primary criterion for evaluating the effectiveness of models was classification accuracy. The analysis revealed that the accuracy of the models showed no significant differences when using different text vectorization methods. This indicates that both approaches to text vectorization were effective for the task and adequately transformed textual data for machine learning [45].

Despite the similarity in terms of accuracy, one of the key factors in choosing the vectorization method was the data processing speed. Vectorization using TF-IDF demonstrated a higher speed of text transformation into vectors compared to the bag-of-words method.

Based on the comparison, TF-IDF was chosen as the primary method of text vectorization in the project. This choice was motivated not only by the similarity in accuracy between the two methods but also by considering the efficiency of data processing. The application of TF-IDF not only effectively represents text as vectors for subsequent model training but also reduces the time required for data preparation, which is a significant advantage when working with large text corpora.

A series of experiments was undertaken utilizing various machine learning techniques to finetune the parameters of classifiers for the effective classification of economic activities in accordance with NACE codes.

For parameter optimization, we leveraged the genetic algorithm from the DEAP library. The optimization process spanned 10 generations. In each iteration, a new generation of individuals was generated by combining the current population using crossover probability as 0.5 and mutation probability as 0.1. Subsequently, each individual in the new generation was evaluated, and fitness values were assigned accordingly. The top-performing individuals were then selected to form the subsequent generation. The optimization efforts targeted parameters across multi-classifiers, including Naive Bayes, Decision Tree, Random Forest, and Multilayer Perceptron. For each candidate in the population, the value of the fitness function was computed. In this case, this corresponded to the accuracy metric of classification quality.

Naive Bayes classifier [46] demonstrated an accuracy of 66%, which is a satisfactory result. It is worth noting that the Naive Bayes classifier is known for its ability to handle large volumes of data and provide a satisfactory classification quality in a relatively short training time.

The Decision Tree model, as described by Song et al. [47], employed specific hyperparameters, which were determined after optimizing the classifier's hyperparameters using a genetic algorithm. These hyperparameters included Gini as the criterion [48], Best as the splitter [49], a maximum tree depth [50] of 20, two samples for split [51], a minimum of one sample for split, a minimum weight fraction [52] for split set to 0, and equal class weights. This approach provided an accuracy of 62.7%, which was lower than that of the Naive Bayes classifier. Although decision trees can capture the relationships between features, there is a possibility that an overly complex tree was built in this experiment, leading to overfitting and a deterioration in the model's ability to generalize information to new data.

Random Forest [53] model achieved the best results after the GA-based fine-tuning optimization process, with the following parameters: 48 trees, tree depth of 93, minimum number of samples required to split an internal node of 94, and minimum number of samples required to form a leaf node of 3. The combination of these parameters yielded the

highest accuracy of 0.71 for the classification of construction and installation works within the given dataset.

The Multilayer Perceptron [54] (MLP) model achieved the highest classification accuracy of 71%, with an F1-score of 0.7. The settings for the artificial neural network obtained after genetic algorithm finetuning optimization included a single hidden layer with 100 neurons, ReLU activation function [55], Adam optimizer [56] for weight tuning, an α coefficient of 0.0001 for L_2 regularization, a batch size of 200 for training, 200 epochs, momentum set to 0.9, and beta values of 0.9 and 0.999 for β_1 and β_2, respectively.

Figure 3 illustrates the evolution of hyperparameters and the accuracy of Decision Tree, Random Forest, and MLP multi-class classifiers across generations of Genetic Algorithm optimization. It is evident from the plot that there is a trend towards the simplification of model architectures with a slight increase in accuracy over generations. This observation underscores the significance of model refinement in handling large volumes of data efficiently.

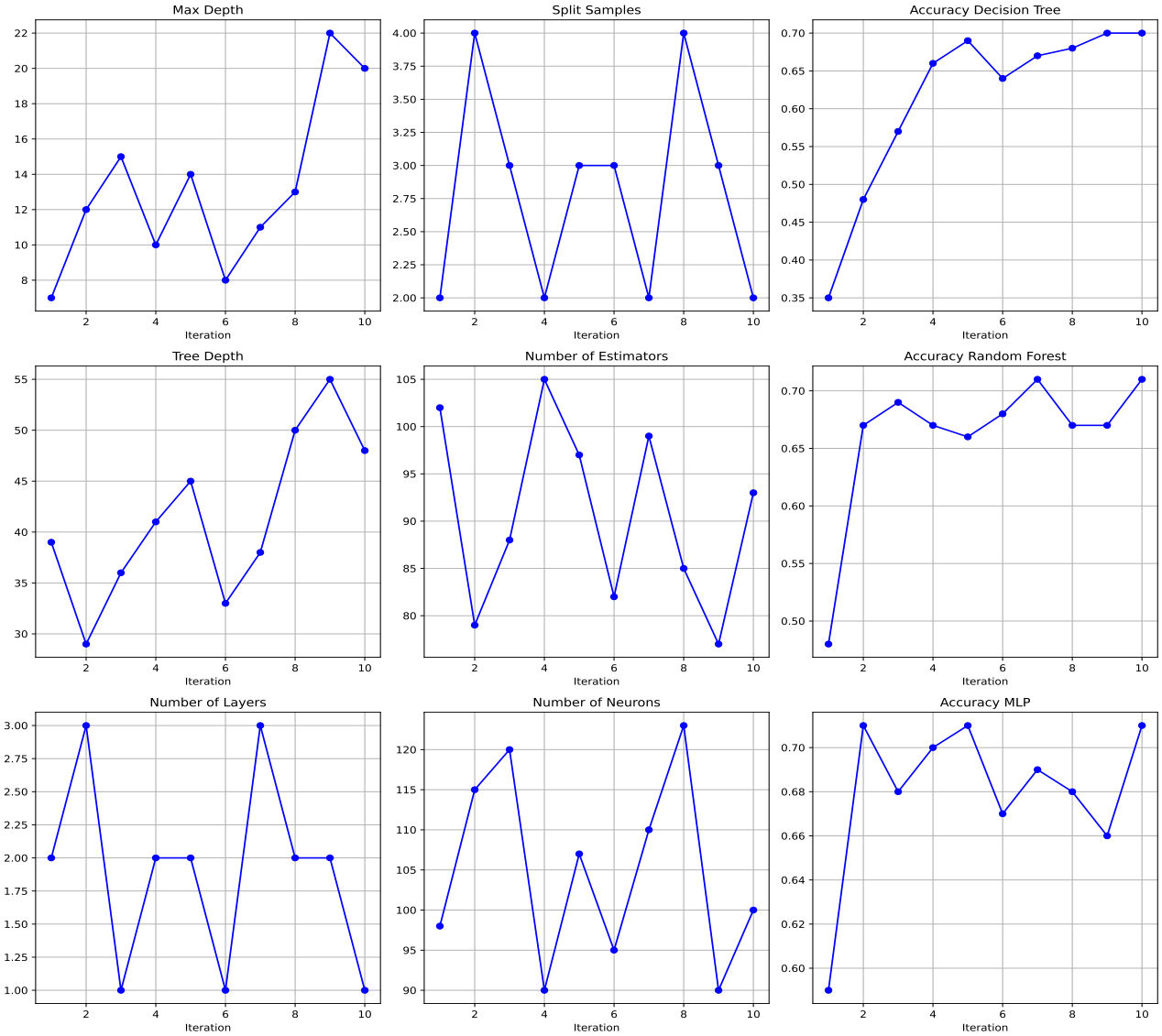

Figure 3. Evolution of hyperparameters and accuracy in Decision Tree (**upper row**), Random Forest (**mid row**), and MLP (**bottom row**) multi-class classifiers across generations of GA hyperparameters during finetuning optimization.

Table 2 presents accuracy and F1-score metrics for each of the considered models based on the conducted experiments.

MLP demonstrated the best performance among the tested methods, achieving a classification accuracy of 71% and an F1-score of 0.73. These metrics indicate the model's high ability to distinguish between classes and its balance between precision and recall.

It is worth mentioning that this model has several advantages, including flexibility in tuning and the ability to identify complex dependencies in the data, which are critical for multi-class classification tasks.

Additionally, the Naive Bayes classifier, while showing slightly inferior results, did not require the finetuning of hyperparameters, highlighting its potential as a reliable baseline solution for initial classification system deployment.

The decision tree, although showing slightly lower results compared to other models, has the advantage of interpretability and ease of visualization, which can be useful when analyzing classification errors and understanding the decision-making logic of the model.

However, for the purpose of achieving the maximum efficiency and accuracy of the system, Multilayer Perceptron will be used as the primary tool for classifying economic activities. Further experimentation is proposed to finetune the MLP hyperparameters to improve classification results and avoid overfitting.

Table 2. Comparison of Classification Methods.

Classification Method	Accuracy	F1-Score
Naive Bayes	66%	0.71
Decision Tree	62.7%	0.70
Random Forest	71%	0.70
Multilayer Perceptron	71%	0.73

3. Results

3.1. Subgroups

During the development of the classification model, a problem of a large number of incorrectly labeled [57] activities in the provided data was identified. The main reasons for the incorrect assignment of codes were identified as follows:

- Random error in code selection by the author.
- Intentional selection of the wrong code.
- Selection of the wrong code due to a lack of information about which code should be assigned.

Table 3, containing examples of economic activity descriptors with the initially labeled NACE codes and the NACE codes predicted by the best classifier, along with possible reasons for discrepancies in the data. For instance, a descriptor like 'Fire depot in the village' lacks information regarding the nature of the activity—whether it involves the repair or procurement of the fire depot. Similarly, the 'Execution of works on the manufacture and installation of fire doors' illustrates an error in code selection, as the predicted code proved to be more accurate than the initially labeled one. This further emphasizes the importance of classification in large datasets, where more accurate classification can be achieved through machine learning than when using human judgment.

The first group of erroneously labeled activities is small and will not be processed separately. For the second group, it makes sense to identify patterns related to price and duration (it is likely that the author's selection of the wrong code for personal benefit is associated with these parameters). For the third group, it was assumed that the author selected the code based on the most suitable description.

Special detection is required for the most numerous third group, as a large amount of incorrect data can affect training and, consequently, classification accuracy.

Table 3. Examples of incorrectly labeled economic activities' descriptions.

Economic Activity Description	Initial NACE	Description of Initial NACE Code	Predicted NACE	Description of Predicted NACE Code	The Type of Discrepancy
Execution of facade repair works	41.2	Building and construction works	43.3	Finishing works in buildings and structures	Intentional selection of the wrong code.
Fire depot in the village	42.1	Road construction	26.3	Communication equipment	Lack of description
Execution of works on the manufacture and installation of fire doors	43.2	Electrical and other types of installation works	43.3	Finishing and finishing works in buildings and structures	Error in NACE code selection
The implementation of works on the device of an inclined lift for low-mobilitygroups of the population	43.2	Electrical and other types of installation works	43.9	Other specialized construction works	Error in NACE code selection
Emergency maintenance in 2019	43.2	Electrical and other types of installation works	33.1	Repair services for metal products, machinery, and equipment	Lack of description
Rent of special equipment	43.9	Other specialized construction works	68.2	Rental services for own or leased real estate.	Intentional selection of the wrong code.

As a result, we proposed to identify within individual groups, each of which represents an NACE code, subgroups/clusters that raise questions.

To solve this problem, the pysubgroups library was used [58]—a tool for identifying lexical patterns among contract descriptions. These subgroups are easy to analyze; instead of labeling a large amount of data, it is necessary to analyze patterns (rules representing words/phrases present in code descriptions) and make a decision based on how well the rule corresponds with the contract.

The pysubgroups library is an instrument used to find subgroups in a dataset. A subgroup is a subset of data that can be distinguished by certain characteristics or patterns, resulting in subgroup members demonstrating behavior that is different from the general population [59]. For example, in a dataset of activities, a subgroup may represent construction activities with unique characteristics, such as seasonality or type of work.

Using pysubgroups to divide activities into subgroups allows for a more detailed analysis and understanding of the data, thus improving the quality of classification. For instance, if we have a general category "Architectural, engineering, and related technical consultancy services", subgroup analysis can reveal subcategories such as "Technical consultancy services", "Construction supervision", and "Architectural and engineering services", each with unique attributes, described by their own lexical rules.

In addition to in-depth analysis, subgroup discovery can contribute to improving the accuracy of predictive models. Models trained on data divided by subgroups can more accurately reflect the characteristics of each subgroup, reducing overall classification error [60]. As a result, the model can more accurately identify anomalies and incorrectly classified examples, which is critically important for cleaning the training dataset before model training.

Finally, the use of subgroups can significantly assist in identifying and correcting misclassified groups [61]. By defining the key features that characterize each subgroup, it is possible to develop a system of rules or modify existing classification models to better recognize and assign activities to their true categories. This can be particularly useful when processing new activities, where the likelihood of error is higher due to the lack of previous examples for training. As a result of subgroup analysis by experts, the existing dataset was cleaned.

3.2. Expert Evaluation

Group **F** and division 71, associated with construction and building works, were considered by experts. Figure 4 shows the distribution of descriptors across these codes.

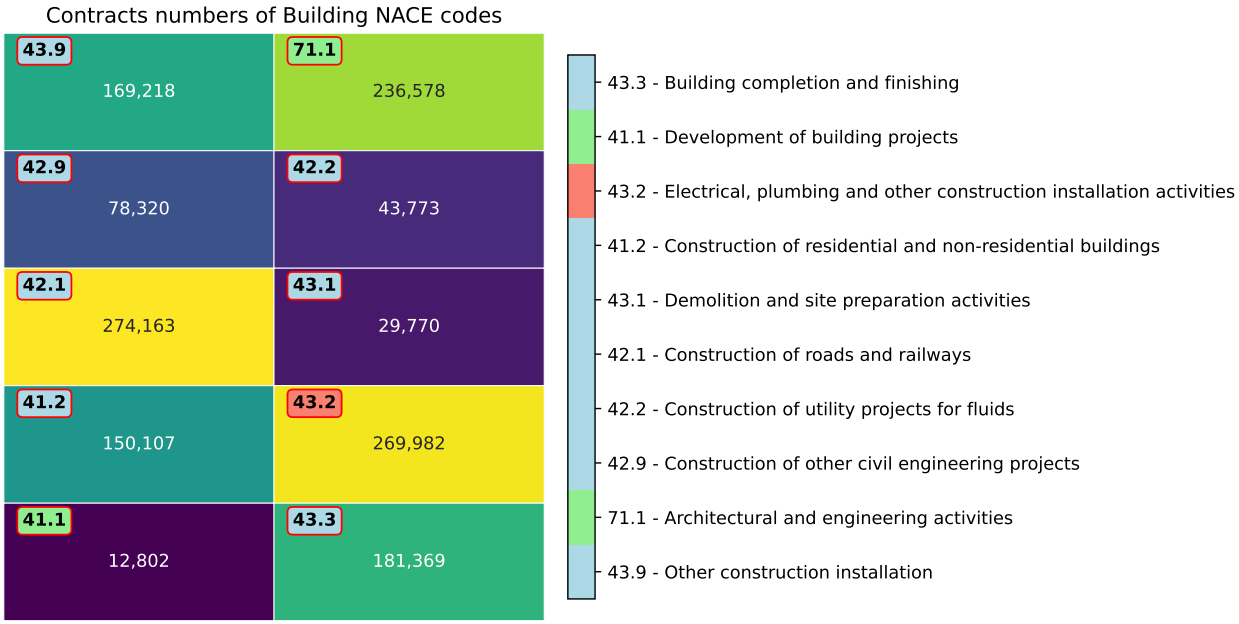

Figure 4. Visual representation of the distribution of economic activities of the considered NACE codes: Construction and Installation Works (CIW)—41.2, 42.1, 42.2, 42.9, 43.1, 43.3, 43.9 (lightblue); Engineering and Surveying Works (ESW)—41.1, 71.1 (lightgreen); Communication Infrastructure Installation (CII)—43.2 (salmon).

Experts evaluated how well each activity's descriptor matched the expected meaning of the NACE code. The expert assessments impact the final results by removing outliers from the data, such as economic activities with a duration exceeding 100 years or costing below EUR 1. The aforementioned contract groups are proposed to be treated as outliers and removed. With a large sample of 20 million records, we can eliminate these few anomalies without harming the data. On the contrary, removing or correcting outliers can improve model quality, reduce overfitting, and enhance its generalization ability.

The following features were discovered when reviewing these activities:

- NACE code 42.1: groups containing phrases indicating road maintenance (including the words "maintenance" and "road AND maintenance") were identified, which are not suitable for this group.
- NACE code 41.1L groups related to construction supervision (including words "control" and "supervision") were identified. These contract groups were marked as not corresponding to their NACE code.
- NACE code 41.2: groups of activities related to procurement rather than construction were identified. The groups based on the key rule "procurement" were designated as miscellaneous.
- NACE code 42.2: activities were divided into two main groups: repair and construction. Subgroups based on the rules "repair" or "repair" AND "capital" were identified, with the remaining activities containing construction objects. A few activities containing the words "technological" AND "connection" were also identified. This group should belong to NACE 43.2.
- NACE code 43.1: subgroups with the word "landscaping" were assigned to NACE 42.9. Additionally, subgroups based on the rules "wood", "wood AND territory", "emergency AND wood", and "supply" were excluded and classified as miscellaneous.
- NACE code 43.2: a separate group of activities based on the rule "alarm" was identified. This group is not relevant to the analyzed code and was classified as miscellaneous.

A visual representation of the concordance between the classifications provided by our predictive model and those determined by domain experts for specific NACE codes is shown in Figure 5.

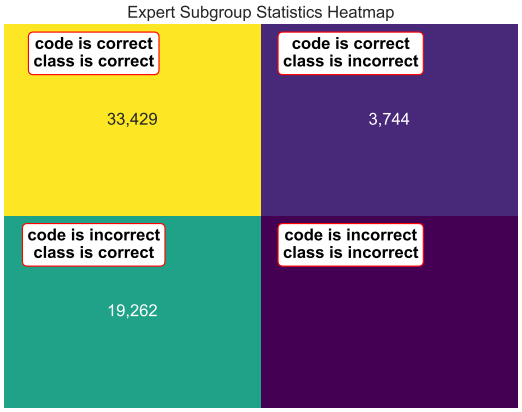

Figure 5. Expert Subgroup Statistics Heatmap for subgroup of 56,435 activities.

The heatmap illustrates three distinct scenarios: (1) instances where both the model and experts correctly identified the code (33,429), (2) cases where the initial code was incorrect, but the model predicted it accurately (19,262), and (3) situations where the code was initially correct, but the model misclassified it (3744). Notably, instances where both the model and experts erred were not considered further, as they could not be discerned with a single expert assessment. Overall, for the selected subgroup, the model's accuracy in predicting NACE codes reached 93%.

Figure 6 illustrates the confusion matrix, providing an in-depth analysis of the multi-class classification MLP model's performance across both balanced and imbalanced subgroups of the original dataset. It offers a comprehensive overview of the predicted versus actual class labels across all categories, achieved through the multilayer perceptron classifier.

The expert evaluation process allowed for the identification and correction of inconsistencies [62] in the classification of activities, ensuring the accuracy and reliability of subsequent analyses and models.

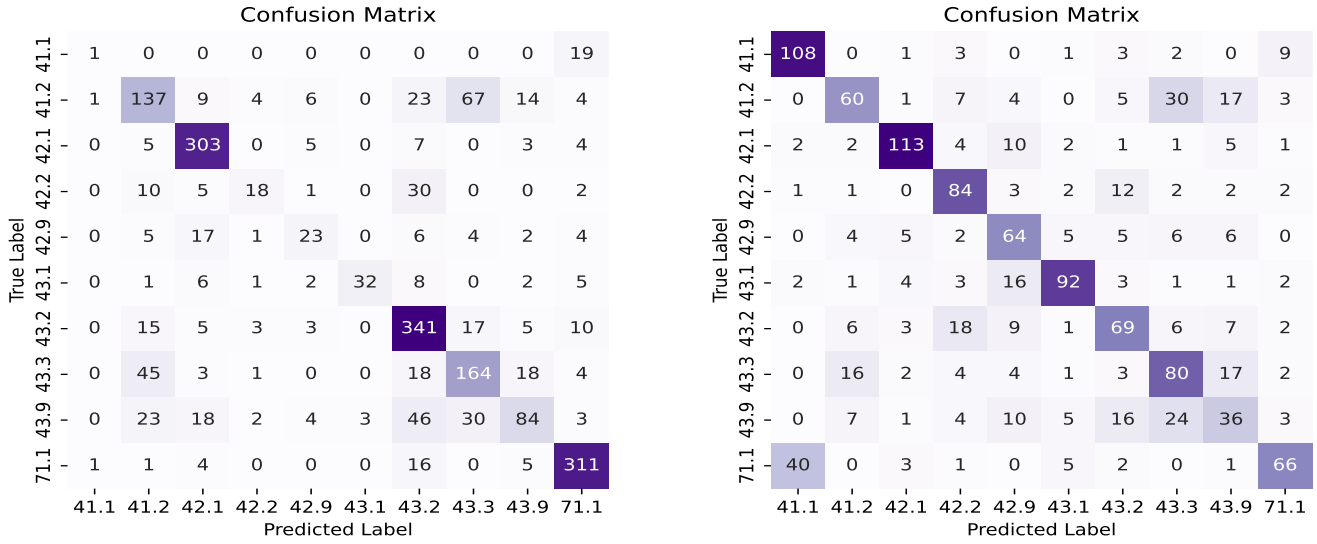

Figure 6. *Cont.*

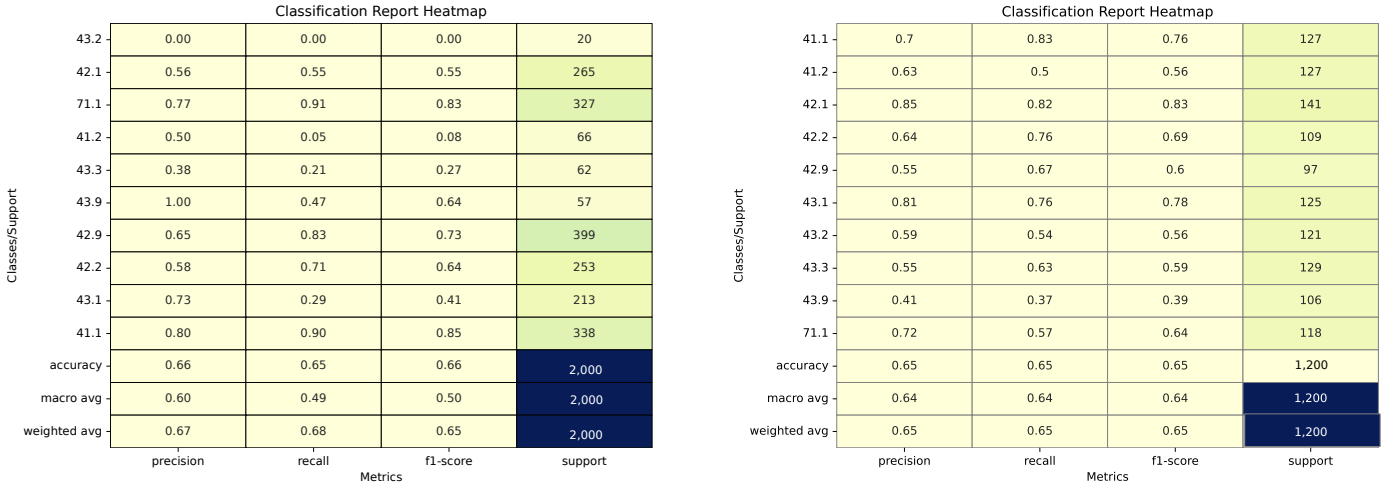

Figure 6. Confusion Matrices and classification reports illustrating MLP multi-class classification results for imbalanced (**left** picture) and balanced (**right**) datasets of economic activity descriptors.

4. Discussion

Based on the analysis conducted and the results obtained, we identified several directions for further work that will help improve the classification system for economic activities and enhance its effectiveness. For instance, additional experiments are proposed to finetune the hyperparameters of the current model [63]. Furthermore, exploring the application of more sophisticated machine learning models, including ensemble methods and deep learning, is recommended to increase classification accuracy. To gain a deeper understanding of the reasons behind misclassifications, a detailed analysis of cases of incorrect classification is proposed [64]. This will allow for weaknesses in the model to be identified and the training process to be adjusted to eliminate systematic errors. To improve the model's generalization ability, it is advisable to meticulously label the training dataset, especially for classes with a high number of erroneous predictions. [65]. These measures may help increase the accuracy and reliability of the classification system, while also ensuring its adaptability and scalability for future tasks.

The need for text classification is widely addressed in various scientific sources. Text classification in NLP is rapidly evolving, driven by transformer-based models like LLMs. Fields et al. [66] surveys text classification techniques across diverse applications, proposing an expanded taxonomy to include multimodal classification. It evaluates model accuracy, discusses the ethical implications, and emphasizes the importance of a nuanced understanding and holistic deployment in real-world scenarios.

The increasing complexity of legal documents demands advanced automated text classification techniques. Xie et al. [67] proposed a method that utilizes Convolutional Neural Networks [68] (Conv1D), which is adept at capturing hierarchical features in sequential data. Max-pooling aids in extracting significant features, while softmax activation handles multi-class legal citation categorization. Addressing previous limitations, their model aims to enhance legal citation text classification, offering a robust solution for automated categorization in the legal domain. A performance evaluation of RandomForest, SVM [69], and MLP models reveals Conv1D's superior outcomes, with a weighted F1-Score of 0.57. Its notable precision and F1-Score underscore its accuracy and suitability for multi-class categorization in analyses of legal text.

Large Language Models [70] (LLMs) excel in various AI and NLP tasks but can lead to issues like fake news if misused. Detecting AI-generated text is crucial for responsible LLM use. Abburi et al. [71] explores how to distinguish AI from human-generated text and attribute text to specific language models. They propose an ensemble neural model using the probabilities from pre-trained LLMs as features for a Traditional Machine Learning (TML) classifier. For English, their F_1 score is 0.733 for AI-generated text classification.

Zhao et al. [72] introduce BERT-Augmented Prompt Engineering in Large Language Models (BAP-LLM), a novel method combining BERT's precision with Foundation Models' extensive training foundations. When applied to the New York Times news corpus, the BAP-LLM approach outperformed existing models, showcasing its potential for automated news categorization and paving the way for hybrid model research. The accuracies of BERT [73], GPT-4 [74], GPT-3 [75], and BAP-LLM are 0.74, 0.76, 0.62, and 0.79, respectively, against a baseline random classifier of 0.19. BAP-LLM surpasses all individual models across metrics, highlighting its efficacy in leveraging BERT's insights and the generative capabilities of Foundation Models.

As a neural network model based on transformers, BERT leverages deep learning and is pre-trained on a large corpus of texts. BERT generates context-dependent vector representations (embeddings) for words and sentences, which can be utilized in various NLP problems.

TF-IDF is a simpler and computationally cheaper method that can be effective in basic text analysis tasks. It is a statistical technique for evaluating the importance of a word in a document within a collection of documents. It generates weight coefficients for words in a document, which are used to represent documents as fixed-length vectors.

One feature of the problem of classifying economic activity based on textual descriptions is the presence of a training dataset with labels, many of which are incorrect (up to 30% of all examples). Thus, the training dataset for classification contains a significant number of anomalies.

Vectorizing text using the relatively simple TF-IDF method is sufficient. The main problem is in constructing a classifier using vectorized data, which shifts the focus from LLMs to multi-class classifiers like Naive Bayes, Decision Trees, Random Forests, and Multilayer Perceptrons for large datasets requiring finetuning. This problem is addressed by applying a genetic algorithm to fine-tune the classifier's hyperparameters.

5. Conclusions

In conclusion, the study focused on improving the classification system for economic activities' descriptions using a combination of machine learning techniques and expert evaluation. Through the analysis, we identified key areas for enhancement and proposed strategies to address them. Optimizing the hyperparameters of a multi-class classifier using GA offers the advantage of automatically finetuning model settings for improved performance and accuracy. Expertly assessing labels on a subset of data can facilitate the construction of an enhanced model to classify economic activities according to NACE codes.

After optimizing the hyperparameters of the multi-class classifiers, the following best configurations were identified: for Random Forest, the number of trees is 48 and the maximum depth of the trees is 93. For MLP, the number of hidden layers is 1, the number of neurons in the hidden layer is 100, and Adam is used as an optimizer. Both achieved an accuracy of 71% and an F1 score of 0.7 and 0.73, respectively for Random Forest and MLP. Expert assessment of a subset of activities, specifically for the construction of NACE codes, revealed a significant portion of incorrect labels in the original dataset (up to 30%), and that the MLP model actually has an accuracy of 93% on the selected data subset.

Overall, these efforts are expected to lead to a more accurate and reliable classification system for economic activities, with increased adaptability and scalability for future tasks in this domain.

Author Contributions: Conceptualization, I.M. (Ivan Malashin), I.M. (Igor Masich), V.T. and A.G.; data curation, I.M. (Igor Masich) and A.B.; formal analysis, I.M. (Ivan Malashin), V.N. and A.B.; funding acquisition, A.B. and A.G.; methodology, A.G.; project administration, A.B. and A.G.; resources, I.M. (Ivan Malashin); software, I.M. (Ivan Malashin) and I.M. (Igor Masich); supervision, A.B. and A.G.; validation, I.M. (Igor Masich), V.T., V.N. and A.B.; visualization, I.M. (Ivan Malashin); writing—original draft, I.M. (Ivan Malashin) and I.M. (Igor Masich); writing—review and editing, V.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article material, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1 provide descriptions of the NACE codes corresponding to the statistics shown in Figure 1. Each row in the table represents a specific NACE code along with its corresponding title, providing a comprehensive reference for understanding the depicted statistics. Descriptions were taken from [76].

These tables serve as a key reference for interpreting the data presented in the corresponding figure, facilitating a better understanding of the statistical insights depicted in the research.

Table A1. NACE codes descriptions.

Division	Title
A.1	Crop and animal production, hunting, and related service activities
A.2	Forestry and logging
A.3	Fishing and aquaculture
B.5	Mining of coal and lignite
B.6	Extraction of crude petroleum and natural gas
B.7	Mining of metal ores
B.8	Other mining and quarrying
B.9	Mining support service activities
C.10	Manufacture of food products
C.11	Manufacture of beverages
C.12	Manufacture of tobacco products
C.13	Manufacture of textiles
C.14	Manufacture of wearing apparel
C.15	Manufacture of leather and related products
C.16	Manufacture of wood and of products of wood and cork, except furniture; manufacture of articles of straw and plaiting materials
C.17	Manufacture of paper and paper products
C.18	Printing and reproduction of recorded media
C.19	Manufacture of coke and refined petroleum products
C.20	Manufacture of chemicals and chemical products
C.21	Manufacture of basic pharmaceutical products and pharmaceutical preparations
C.22	Manufacture of rubber and plastic products
C.23	Manufacture of other non-metallic mineral products
C.24	Manufacture of basic metals
C.25	Manufacture of fabricated metal products, except machinery and equipment
C.26	Manufacture of computer, electronic, and optical products
C.27	Manufacture of electrical equipment
C.28	Manufacture of machinery and equipment n.e.c.
C.29	Manufacture of motor vehicles, trailers, and semi-trailers
C.30	Manufacture of other transport equipment
C.31	Manufacture of furniture
C.32	Other manufacturing
C.33	Repair and installation of machinery and equipment
D.35	Electricity, gas, steam, and air conditioning supply
E.36	Water collection, treatment, and supply
E.37	Sewerage
E.38	Waste collection, treatment, and disposal activities; materials recovery
E.39	Remediation activities and other waste management services
F.41	Construction of buildings
F.42	Civil engineering

Table A1. *Cont.*

Division	Title
F.43	Specialised construction activities
G.45	Wholesale and retail trade and repair of motor vehicles and motorcycles
G.46	Wholesale trade, except of motor vehicles and motorcycles
G.47	Retail trade, except of motor vehicles and motorcycles
H.49	Land transport and transport via pipelines
H.50	Water transport
H.51	Air transport
H.52	Warehousing and support activities for transportation
H.53	Postal and courier activities
I.55	Accommodation
I.56	Food and beverage service activities
J.58	Publishing activities
J.59	Motion picture, video and television programme production, sound recording, and music publishing activities
J.60	Programming and broadcasting activities
J.61	Telecommunications
J.62	Computer programming, consultancy, and related activities
J.63	Information service activities
K.64	Financial service activities, except insurance, and pension funding
K.65	Insurance, reinsurance, and pension funding, except compulsory social security
K.66	Activities auxiliary to financial services and insurance activities
L.68	Real estate activities
M.69	Legal and accounting activities
M.70	Activities of head offices; management consultancy activities
M.71	Architectural and engineering activities; technical testing and analysis
M.72	Scientific research and development
M.73	Advertising and market research
M.74	Other professional, scientific, and technical activities
M.75	Veterinary activities
N.77	Rental and leasing activities
N.78	Employment activities
N.79	Travel agency, tour operator reservation service, and related activities
N.80	Security and investigation activities
N.81	Services to buildings and landscape activities
N.82	Office administrative, office support, and other business support activities
O.84	Public administration and defence; compulsory social security
P.85	Education
Q.86	Human health activities
Q.87	Residential care activities
Q.88	Social work activities without accommodation
R.90	Creative, arts, and entertainment activities
R.91	Libraries, archives, museums, and other cultural activities
R.92	Gambling and betting activities
R.93	Sports activities and amusement and recreation activities
S.94	Activities of membership organisations
S.95	Repair of computers and personal and household goods
S.96	Other personal service activities
T.97	Activities of households as employers of domestic personnel
T.98	Undifferentiated goods-and services-producing activities of private households for own use
U.99	Activities of extraterritorial organisations and bodies

References

1. Schnabl, E.; Zenker, A. *Statistical Classification of Knowledge-Intensive Business Services (KIBS) with NACE Rev. 2*; Fraunhofer ISI: Karlsruhe, Germany, 2013; Volume 25.
2. Nijhowne, S. Defining and classifying statistical units. In *Business Survey Methods*; Wiley Online Library: New York, NY, USA, 1995; pp. 49–64.
3. Barrier, E.B. The concept of sustainable economic development. In *The Economics of Sustainability*; Routledge: London, UK, 2017; pp. 87–96.
4. Graiet, M.; Mammar, A.; Boubaker, S.; Gaaloul, W. Towards correct cloud resource allocation in business processes. *IEEE Trans. Serv. Comput.* **2016**, *10*, 23–36. [CrossRef]
5. Ievdokymov, V.; Ostapchuk, T.; Lehenchuk, S.; Grytsyshen, D.; Marchuk, G. *Analysis of the Impact of Intangible Assets on the Companies' Market Value*; Natsional'nyi Hirnychyi Universytet. Naukovyi Visnyk: Dnipropetrovsk Oblast, Ukraine, 2020; pp. 164–170.
6. Békés, G.; Muraközy, B.; Harasztosi, P. Firms and products in international trade: Evidence from Hungary. *Econ. Syst.* **2011**, *35*, 4–24. [CrossRef]

7. Fröhlich, M. Nowcasting short-term indicators with machine learning methods. *Stat. J. IAOS* **2022**, *38*, 1411–1436. [CrossRef]
8. Ambrois, M.; Butticè, V.; Caviggioli, F.; Cerulli, G.; Croce, A.; De Marco, A.; Giordano, A.; Resce, G.; Toschi, L.; Ughetto, E.; et al. *Using Machine Learning to Map the European Cleantech Sector*; Technical report, EIF Working Paper; European Investment Fund (EIF): Luxembourg, 2023.
9. Boselli, R.; Cesarini, M.; Mercorio, F.; Mezzanzanica, M. Using machine learning for labour market intelligence. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, 18–22 September 2017; Proceedings, Part III 10; Springer: Cham, Switzerland, 2017; pp. 330–342.
10. d'Andrimont, R.; Yordanov, M.; Martinez-Sanchez, L.; Eiselt, B.; Palmieri, A.; Dominici, P.; Gallego, J.; Reuter, H.I.; Joebges, C.; Lemoine, G.; et al. Harmonised LUCAS in-situ land cover and use database for field surveys from 2006 to 2018 in the European Union. *Sci. Data* **2020**, *7*, 352. [CrossRef] [PubMed]
11. Redlein, A.; Stopajnik, E. Current labour market situation and upcoming trends in the European Facility Service Industry. *J. Facil. Manag. Educ. Res.* **2017** *1*, 1.
12. Gite, S.; Patil, S.; Dharrao, D.; Yadav, M.; Basak, S.; Rajendran, A.; Kotecha, K. Textual feature extraction using ant colony optimization for hate speech classification. *Big Data Cogn. Comput.* **2023**, *7*, 45. [CrossRef]
13. Jasmir, J.; Nurmaini, S.; Tutuko, B. Fine-grained algorithm for improving knn computational performance on clinical trials text classification. *Big Data Cogn. Comput.* **2021**, *5*, 60. [CrossRef]
14. Hawalah, A. Semantic ontology-based approach to enhance Arabic text classification. *Big Data Cogn. Comput.* **2019**, *3*, 53. [CrossRef]
15. Masich, I.; Rezova, N.; Shkaberina, G.; Mironov, S.; Bartosh, M.; Kazakovtsev, L. Subgroup Discovery in Machine Learning Problems with Formal Concepts Analysis and Test Theory Algorithms. *Algorithms* **2023**, *16*, 246. [CrossRef]
16. Salloum, S.; Dautov, R.; Chen, X.; Peng, P.X.; Huang, J.Z. Big data analytics on Apache Spark. *Int. J. Data Sci. Anal.* **2016**, *1*, 145–164.
17. Ahmed, N.; Barczak, A.; Rashid, M. An enhanced parallelisation model for performance prediction of Apache Spark on a multinode Hadoop cluster. *Big Data Cogn. Comput.* **2021**, *5*, 65. [CrossRef]
18. Ayazbayev, D.; Bogdanchikov, A.; Orynbekova, K. Defining Semantically Close Words of Kazakh Language with Distributed System Apache Spark. *Big Data Cogn. Comput.* **2023**, *7*, 160. [CrossRef]
19. Kroß, J.; Krcmar, H. Pertract: Model extraction and specification of big data systems for performance prediction by the example of Apache Spark and Hadoop. *Big Data Cogn. Comput.* **2019**, *3*, 47. [CrossRef]
20. Chowdhary, K.; Chowdhary, K. Natural language processing. In *Fundamentals of Artificial Intelligence*; Springer: New Delhi, India, 2020; pp. 603–649.
21. Musleh, D.A.; Alkhwaja, I.; Alkhwaja, A.; Alghamdi, M.; Abahussain, H.; Alfawaz, F.; Min-Allah, N.; Abdulqader, M.M. Arabic Sentiment Analysis of YouTube Comments: NLP-Based Machine Learning Approaches for Content Evaluation. *Big Data Cogn. Comput.* **2023**, *7*, 127. [CrossRef]
22. Clark, A. Magic words: How language augments human computation. In *Language and Meaning in Cognitive Science*; Routledge: London, UK, 2012; pp. 21–39.
23. Anandarajan, M.; Hill, C.; Nolan, T.; Anandarajan, M.; Hill, C.; Nolan, T. Text preprocessing. In *Practical Text Analytics: Maximizing the Value of Text Data*; Springer: Cham, Switzerland, 2019; pp. 45–59.
24. Clark, E.; Araki, K. Text normalization in social media: Progress, problems and applications for a pre-processing system of casual English. *Procedia-Soc. Behav. Sci.* **2011**, *27*, 2–11. [CrossRef]
25. Büschken, J.; Allenby, G.M. Improving text analysis using sentence conjunctions and punctuation. *Mark. Sci.* **2020**, *39*, 727–742. [CrossRef]
26. Zhao, J.; Gui, X. Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access* **2017**, *5*, 2870–2879.
27. Agnihotri, D.; Verma, K.; Tripathi, P. Pattern and cluster mining on text data. In Proceedings of the 2014 Fourth International Conference on Communication Systems and Network Technologies, Bhopal, India, 7–9 April 2014; pp. 428–432.
28. Kaufmann, M.; Kalita, J. Syntactic normalization of twitter messages. In Proceedings of the International Conference on Natural Language Processing, Kharagpur, India, 8–11 December 2010; Volume 16.
29. Vijayarani, S.; Janani, R. Text mining: Open source tokenization tools-an analysis. *Adv. Comput. Intell. Int. J. (ACII)* **2016**, *3*, 37–47.
30. Singh, J.; Gupta, V. Text stemming: Approaches, applications, and challenges. *ACM Comput. Surv. (CSUR)* **2016**, *49*, 1–46. [CrossRef]
31. Khyani, D.; Siddhartha, B.; Niveditha, N.; Divya, B. An interpretation of lemmatization and stemming in natural language processing. *J. Univ. Shanghai Sci. Technol.* **2021**, *22*, 350–357.
32. Korenius, T.; Laurikkala, J.; Järvelin, K.; Juhola, M. Stemming and lemmatization in the clustering of finnish text documents. In Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, Washington, DC, USA, 8–13 November 2004; pp. 625–633.
33. Balakrishnan, V.; Lloyd-Yemoh, E. Stemming and lemmatization: A comparison of retrieval performances. In Proceedings of the SCEI Seoul Conferences, Seoul, Republic of Korea, 10–11 April 2014.
34. Saranya, S.; Usha, G. A Machine Learning-Based Technique with IntelligentWordNet Lemmatize for Twitter Sentiment Analysis. *Intell. Autom. Soft Comput.* **2023**, *36*, 339–352. [CrossRef]

35. Yang, X.; Yang, K.; Cui, T.; Chen, M.; He, L. A Study of Text Vectorization Method Combining Topic Model and Transfer Learning. *Processes* **2022**, *10*, 350. [CrossRef]
36. Qiu, D.; Jiang, H.; Chen, S. Fuzzy information retrieval based on continuous bag-of-words model. *Symmetry* **2020**, *12*, 225. [CrossRef]
37. Jang, B.; Kim, M.; Harerimana, G.; Kang, S.U.; Kim, J.W. Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism. *Appl. Sci.* **2020**, *10*, 5841. [CrossRef]
38. Roshan, R.; Bhacho, I.A.; Zai, S. Comparative Analysis of TF–IDF and Hashing Vectorizer for Fake News Detection in Sindhi: A Machine Learning and Deep Learning Approach. *Eng. Proc.* **2023**, *46*, 5. [CrossRef]
39. Abubakar, H.D.; Umar, M.; Bakale, M.A. Sentiment classification: Review of text vectorization methods: Bag of words, Tf-Idf, Word2vec and Doc2vec. *SLU J. Sci. Technol.* **2022**, *4*, 27–33. [CrossRef]
40. Diao, R.; Chao, F.; Peng, T.; Snooke, N.; Shen, Q. Feature selection inspired classifier ensemble reduction. *IEEE Trans. Cybern.* **2013**, *44*, 1259–1268. [CrossRef] [PubMed]
41. Dos Santos, E.M.; Sabourin, R.; Maupin, P. Overfitting cautious selection of classifier ensembles with genetic algorithms. *Inf. Fusion* **2009**, *10*, 150–162. [CrossRef]
42. Wang, N.; Wang, P.; Zhang, B. An improved TF-IDF weights function based on information theory. In Proceedings of the 2010 International Conference on Computer and Communication Technologies in Agriculture Engineering, Chengdu, China, 12–13 June 2010; Volume 3, pp. 439–441.
43. Turki, T.; Roy, S.S. Novel hate speech detection using word cloud visualization and ensemble learning coupled with count vectorizer. *Appl. Sci.* **2022**, *12*, 6611. [CrossRef]
44. Kumar, V.; Subba, B. A TfidfVectorizer and SVM based sentiment analysis framework for text data corpus. In Proceedings of the 2020 National Conference on Communications (NCC), Kharagpur, India, 21–23 February 2020; pp. 1–6.
45. Egger, R. Text Representations and Word Embeddings: Vectorizing Textual Data. In *Applied Data Science in Tourism: Interdisciplinary Approaches, Methodologies, and Applications*; Springer: Cham, Switzerland, 2022; pp. 335–361.
46. Leung, K.M. Naive bayesian classifier. *Polytech. Univ. Dep. Comput. Sci. Risk Eng.* **2007**, *2007*, 123–156.
47. Song, Y.Y.; Ying, L. Decision tree methods: Applications for classification and prediction. *Shanghai Arch. Psychiatry* **2015**, *27*, 130. [PubMed]
48. Tangirala, S. Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 612–619. [CrossRef]
49. Anmala, J.; Turuganti, V. Comparison of the performance of decision tree (DT) algorithms and extreme learning machine (ELM) model in the prediction of water quality of the Upper Green River watershed. *Water Environ. Res.* **2021**, *93*, 2360–2373. [CrossRef] [PubMed]
50. Silva, S.; Almeida, J. Dynamic maximum tree depth: A simple technique for avoiding bloat in tree-based gp. In Proceedings of the Genetic and Evolutionary Computation—GECCO 2003: Genetic and Evolutionary Computation Conference, Chicago, IL, USA, 12–16 July 2003; Proceedings, Part II; Springer: Cham, Switzerland, 2003; pp. 1776–1787.
51. Buntine, W.; Niblett, T. A further comparison of splitting rules for decision-tree induction. *Mach. Learn.* **1992**, *8*, 75–85. [CrossRef]
52. Chan, T.M.; Zheng, D.W. Hopcroft's problem, log-star shaving, 2D fractional cascading, and decision trees. *ACM Trans. Algorithms* **2022**. [CrossRef]
53. Algehyne, E.A.; Jibril, M.L.; Algehainy, N.A.; Alamri, O.A.; Alzahrani, A.K. Fuzzy neural network expert system with an improved Gini index random forest-based feature importance measure algorithm for early diagnosis of breast cancer in Saudi Arabia. *Big Data Cogn. Comput.* **2022**, *6*, 13. [CrossRef]
54. Taud, H.; Mas, J. Multilayer perceptron (MLP). In *Geomatic Approaches for Modeling Land Change Scenarios*; Springer: Cham, Switzerland, 2018; pp. 451–455.
55. Banerjee, C.; Mukherjee, T.; Pasiliao, E., Jr. An empirical study on generalizations of the ReLU activation function. In Proceedings of the 2019 ACM Southeast Conference, Kennesaw, GA, USA, 18–20 April 2019; pp. 164–167.
56. Zhang, Z. Improved adam optimizer for deep neural networks. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–2.
57. Brodley, C.E.; Friedl, M.A. Identifying and eliminating mislabeled training instances. In Proceedings of the National Conference on Artificial Intelligence, Portland, OR, USA, 4–8 August 1996; pp. 799–805.
58. Lemmerich, F.; Becker, M. pysubgroup: Easy-to-use subgroup discovery in python. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, 10–14 September 2018; Proceedings, Part III 18; Springer: Cham, Switzerland, 2019; pp. 658–662.
59. Atzmueller, M. Subgroup discovery. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2015**, *5*, 35–49. [CrossRef]
60. Kim, M.P.; Ghorbani, A.; Zou, J. Multiaccuracy: Black-box post-processing for fairness in classification. In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, Honolulu, HI, USA, 27–28 January 2019; pp. 247–254.
61. De Lusignan, S.; Khunti, K.; Belsey, J.; Hattersley, A.; Van Vlymen, J.; Gallagher, H.; Millett, C.; Hague, N.; Tomson, C.; Harris, K.; et al. A method of identifying and correcting miscoding, misclassification and misdiagnosis in diabetes: A pilot and validation study of routinely collected data. *Diabet. Med.* **2010**, *27*, 203–209. [CrossRef] [PubMed]

62. Oishi, S.M.; Morton, S.C.; Moore, A.A.; Beck, J.C.; Hays, R.D.; Spritzer, K.L.; Partridge, J.M.; Fink, A. Using data to enhance the expert panel process: Rating indications of alcohol-related problems in older adults. *Int. J. Technol. Assess. Health Care* **2001**, *17*, 125–136. [CrossRef] [PubMed]
63. Li, H.; Chaudhari, P.; Yang, H.; Lam, M.; Ravichandran, A.; Bhotika, R.; Soatto, S. Rethinking the hyperparameters for fine-tuning. *arXiv* **2020**, arXiv:2002.11770.
64. Fielding, A.H.; Bell, J.F. A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environ. Conserv.* **1997**, *24*, 38–49. [CrossRef]
65. Brodley, C.E.; Friedl, M.A. Identifying mislabeled training data. *J. Artif. Intell. Res.* **1999**, *11*, 131–167. [CrossRef]
66. Fields, J.; Chovanec, K.; Madiraju, P. A Survey of Text Classification with Transformers: How wide? How large? How long? How accurate? How expensive? How safe? *IEEE Access* **2024**, *12*, 6518–6531. [CrossRef]
67. Xie, Y.; Li, Z.; Yin, Y.; Wei, Z.; Xu, G.; Luo, Y. Advancing Legal Citation Text Classification A Conv1D-Based Approach for Multi-Class Classification. *J. Theory Pract. Eng. Sci.* **2024**, *4*, 15–22. [CrossRef] [PubMed]
68. Phiphitphatphaisit, S.; Surinta, O. Deep feature extraction technique based on Conv1D and LSTM network for food image recognition. *Eng. Appl. Sci. Res.* **2021**, *48*, 581–592.
69. Zub, K.; Zhezhnych, P.; Strauss, C. Two-Stage PNN–SVM Ensemble for Higher Education Admission Prediction. *Big Data Cogn. Comput.* **2023**, *7*, 83. [CrossRef]
70. Kasneci, E.; Seßler, K.; Küchemann, S.; Bannert, M.; Dementieva, D.; Fischer, F.; Gasser, U.; Groh, G.; Günnemann, S.; Hüllermeier, E.; et al. ChatGPT for good? On opportunities and challenges of large language models for education. *Learn. Individ. Differ.* **2023**, *103*, 102274. [CrossRef]
71. Abburi, H.; Suesserman, M.; Pudota, N.; Veeramani, B.; Bowen, E.; Bhattacharya, S. Generative ai text classification using ensemble llm approaches. *arXiv* **2023**, arXiv:2309.07755.
72. Zhao, F.; Yu, F. Enhancing Multi-Class News Classification through Bert-Augmented Prompt Engineering in Large Language Models: A Novel Approach. In Proceedings of the 10th International Scientific and Practical Conference "Problems and Prospects of Modern Science and Education", Stockholm, Sweden, 12–15 March 2024; International Science Group: New York, NY, USA, 2024; 381p.
73. Prottasha, N.J.; Sami, A.A.; Kowsher, M.; Murad, S.A.; Bairagi, A.K.; Masud, M.; Baz, M. Transfer learning for sentiment analysis using BERT based supervised fine-tuning. *Sensors* **2022**, *22*, 4157. [CrossRef] [PubMed]
74. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
75. Floridi, L.; Chiriatti, M. GPT-3: Its nature, scope, limits, and consequences. *Minds Mach.* **2020**, *30*, 681–694. [CrossRef]
76. NACE: Statistical Classification of Economic Activities in the European Community. Available online: https://ec.europa.eu/eurostat/web/nace/overview (accessed on 10 June 2024).