One-Stop-Shop Artificial Intelligence and Machine Learning for Official Statistics

Project 101146355 – AIML4OS

Workpackage 11

USE CASE - ML for enterprise network modelling

Deliverable	D11.2 – Report describing the model specifications
Month due	31 May 2026
Type	R – Document, report
Prepared by	Przemysław Skrzypczak (Stats Poland), Grzegorz Bujwid (Stats Poland), Edwin de Jonge (CBS), Gert Buiten (CBS)

Workpackage Leader:

Gert Buiten and Peter Nooteboom (CBS)
g.buiten@cbs.nl and pd.nooteboom@cbs.nl

Deliverable description

This report is produced by Workpackage 11 of the AIML4OS project. The main outcome of this WP are AIML-based models and software that each EU National Statistical Institute (NSI) could use to derive a firm-level supply chain network for their national economy, allowing for a range of economic and policy analyses. As such, the WP consists of tasks on

- describing the firm-level dataset needed for an NSI to derive such a network,
- developing and training the Machine Learning models on an actual national firm-level network dataset to be used by a NSI,
- and the task to derive a national firm-level network dataset that can be used as training set for the Machine Learning (ML) models.

This report is Deliverable D11.2 of WP11 and documents the successful completion of Task 11.2 "Use ML to develop models for deriving binary networks from firm-level data".

In this task, the following steps were undertaken:

1. The task validated the models using ML quality indicators based on the test sets. It will check the best model by applying it to firm-data and will compare the estimated/derived network with ground truth network datasets for one or more countries

Originally, an extension was envisaged to develop a method that nowcasts the network structure for year *t+1* based on a model for year *t* combined with firm-level data on *t+1*. This method could not be tested since the firm-level network dataset for training and testing could only be made for one year. Therefore, it was not possible to reconstruct a network for a second year and compare it with the 'real' data for the second year. However, it is nevertheless possible to use a model trained on year for reconstructing year t+1. This will be briefly discussed in a separate paragraph below.

Abstract

This report is being submitted in accordance with deliverable 11.2 within Workpackage 11. The purpose of this deliverable is to document the process of training and benchmarking multiple ML classifiers on a training set derived from the Portuguese firm-to-firm network (constructed in Task 11.1 of WP11). Model performance was systematically quantified to identify the optimal classifier in terms of predictive quality and computational efficiency. The selected model will subsequently be operationalised within a pipeline, which will be the main result of WP11. The report also describes how the selected model will be further improved in the next round of training, in combination with three other top performing models. This combined approach helps to increase explainability, prevention of bias and improved feature engineering. It also describes how we deal with the 'rare event' (or class imbalance) problem arising from the fact that only less than one percent of potential firm to firm links exists.

Introduction

The main aim of WP11 is to create a software pipeline that EU NSI's can use to reconstruct firm-level supply chain networks using only regular data that is already available at all NSI's – such as the statistical business register, firm addresses, NACE-codes and VAT-data on turnover. A crucial part (the engine so to speak) of this pipeline is described in this report: the reconstruction model and the way it has been trained on a Portuguese network dataset based on exhaustive administrative data. That dataset was developed in an earlier task of WP11 and is described in a separate report[1].

The current report describes the modelling process. This process is rooted in recent academic literature on firm-level network reconstruction and on the potential use of Machine Learning methods (ML) in that process [2].

An important insight from the literature is that a limited number of countries have very rich datasets on domestic firm-to-firm transactions, usually based on national legislation on extensions to regular VAT-declarations or the mandatory use of e-Invoicing systems for reporting all interfirm transactions in order to prevent VAT fraud. These datasets are available for a small number of countries, e.g. Belgium, Hungary, Ecuador and – with the help of the AIML4OS project – for Portugal. The majority of countries do not have similar administrative datasets. Fortunately, the literature also suggests that the firm-level networks show similar basic structural characteristics for all the countries with rich datasets.

This creates the possibility of using ML to train models on data from one country to reconstruct a firm-level network for another country. This approach was successfully developed and tested by Mungo et al. (2023; 2024) [3][4]. In WP11, we replicate that approach based on Portuguese network data and finetune it to use by NSI's in (at least) the EU to create predicted supply chain networks.

Objective and experimental setup

From an ML perspective, reconstructing firm-level networks is fundamentally a class imbalance problem. Supply chain networks are extremely sparse: only less than one percent of buyer/supplier links actually exist. This means that positive instances (observed links) are vastly outnumbered by negative ones (absent links). Training a classifier on the full dataset (with all negative examples) would result in models that predict all links as absent, simply because that is the overwhelming majority class. Addressing this requires sampling strategies to construct a balanced training set, one that contains a representative proportion of both existing and non-existing links.

The objective of this experiment was to evaluate and compare the performance of a broad set of supervised machine learning models for a binary classification task developed within Work Package 11 (WP11) of the AIML4OS project.

---

[1] Quaresma, S., Cunha, A., Poças, J., Skrzypcza, P., Bujwid, G., Anholzer, M., de Jonge, E., Buiten, G., One-Stop-Shop Artificial Intelligence and Machine Learning for Official Statistics, 2025, D11.1 – Report describing the training and test sets Rev 1.

[2] Bacilieri, A., Borsos, A., Astudillo-Estevez, P., Hoefer, M., & Lafond, F. Firm-level production networks: What do we (really) know?, 2026, Journal of Economic Dynamics and Control, 105313.

[3] Mungo, L., Lafond, F., Astudillo-Estévez, P., and Doyne Farmer, J., Reconstructing production networks using machine learning, 2023, J.Econ. Dyn. Control, 148, 104607.

[4] Mungo, L., Brintrup, A., Garlaschelli, D. and Lafond, F., Reconstructing supply networks, 2024 ,J. Phys. Complex. 5

The task consists in predicting the target class based on firm-level features derived from the constructed dataset.

Description of Classes and Balancing Strategy

Classes and Training Set Balancing

Description:
The training dataset consists of two classes representing supplier–buyer links:

- Positive class (LINKED = 1): contains all observed and verified links between suppliers and buyers.
- Negative class (LINKED = 0): initially contains no explicit links and is generated through a negative sampling procedure.

Balancing Procedure:
To ensure a balanced dataset for model training, non-existing links (negative examples) are added to the training set using a controlled sampling strategy:

- Candidate negative pairs are drawn from suppliers and buyers within the same NACE sections to maintain economic plausibility.
- Randomly selected non-existing links that do not appear in the positive dataset are added to form a negative class.
- Oversampling of negative examples or undersampling of positive examples is applied to achieve a 50/50 balance between classes.

Rationale:
This approach improves the model's ability to prevent bias toward the majority class, and ensures that predictions are both accurate and economically interpretable.

All models were trained and evaluated on a large-scale dataset comprising approximately 80 million observations.

The training dataset is artificially balanced, with two classes of approximately equal size. However, in real-world firm-level networks, less than 1% of all potential links actually exist, highlighting the highly imbalanced nature of the underlying prediction problem. In the specific example described in this document, the Portuguese dataset originally contained 40.189.160 positive transaction records. Hence, these records were balanced with 40.189.160 negative records, summing up to 80.378.320 records in total.

Model performance was assessed using standard classification metrics, including precision, recall, F1-score (per class, macro and weighted averages), and overall accuracy. . These indicators evaluate the quality of link prediction by comparing predicted links with the observed ground truth links in the original dataset.

Given the balanced nature of the dataset, particular emphasis was placed on the macro-averaged F1-score, which is the unweighted average of the F1 score calculated independently for each class. Hence, the macro-averaged F1 score provides a symmetric evaluation of model performance across both classes.

Table 1: Overview of the models included in the evaluation, along with the rationale for their selection. The objective is to assess predictive performance and compare model behaviour across linear, non-linear, and ensemble approaches.

Model	***Reason for selecting it for evaluation***
Logistic Regression	Used as a baseline linear classifier due to its simplicity, interpretability, and established performance in binary classification tasks. Provides a reference for evaluating more complex models.
Naive Bayes	Included as a probabilistic baseline model based on strong feature independence assumptions. Its low computational complexity makes it suitable for benchmarking.
Quadratic Discriminant Analysis	A generative model capturing non-linear decision boundaries through class-specific covariance estimation, enabling comparison with linear and non-linear classifiers.
K-Nearest Neighbours	Non-parametric, instance-based method included to evaluate performance without assumptions on data distribution; captures local data structures.
MLP Multilayer Perceptron	Feedforward neural network included to assess the benefits of deep, non-linear representations in modelling complex data relationships.
AdaBoost	Ensemble boosting method focusing on misclassified observations, enabling construction of a strong classifier from weak learners; included to evaluate boosting approaches.
Extra Trees	Extra Trees – Ensemble method based on randomized trees; representative of bagging-based ensemble approaches.
XGBoost	Gradient boosting framework with regularization and efficient optimization; included due to strong empirical performance on structured data.
LightGBM	Gradient boosting framework optimized for efficiency and scalability, suitable for large datasets; included to evaluate performance-speed trade-offs.
Histogram-Based Gradient Boosting	Scalable gradient boosting using histogram binning to improve computational efficiency, particularly for large datasets.
Random Forest	Widely used ensemble method based on bagging and feature randomness; included for robustness, stability, and strong baseline performance.

The models mentioned above were trained using 2022 Portuguese firm-level network dataset that was created as part of the AIML4OS WP11 activities. Due to the sensitive character of these microdata, work had to be done on-site in Portugal on a special stand-alone computer.

Overall performance comparison

The evaluated models can be grouped into three broad categories according to their performance.

Models with limited predictive performance
Logistic Regression, Naive Bayes, and Quadratic Discriminant Analysis exhibit relatively low predictive accuracy (approximately 50–60%) and low macro-averaged F1-scores (between 0.34 and 0.52; Figure 1).

These models likely exhibit asymmetric classification behaviour, as suggested by the discrepancy between precision and overall accuracy. However, this should be further verified using class-specific metrics such as confusion matrices or per-class recall.
This indicates that their underlying assumptions (linearity, conditional independence, or Gaussian class distributions) are not compatible with the complex and non-linear structure of the data.

Consequently, these approaches are not considered competitive for further development within WP11 given their relatively low predictive performance.

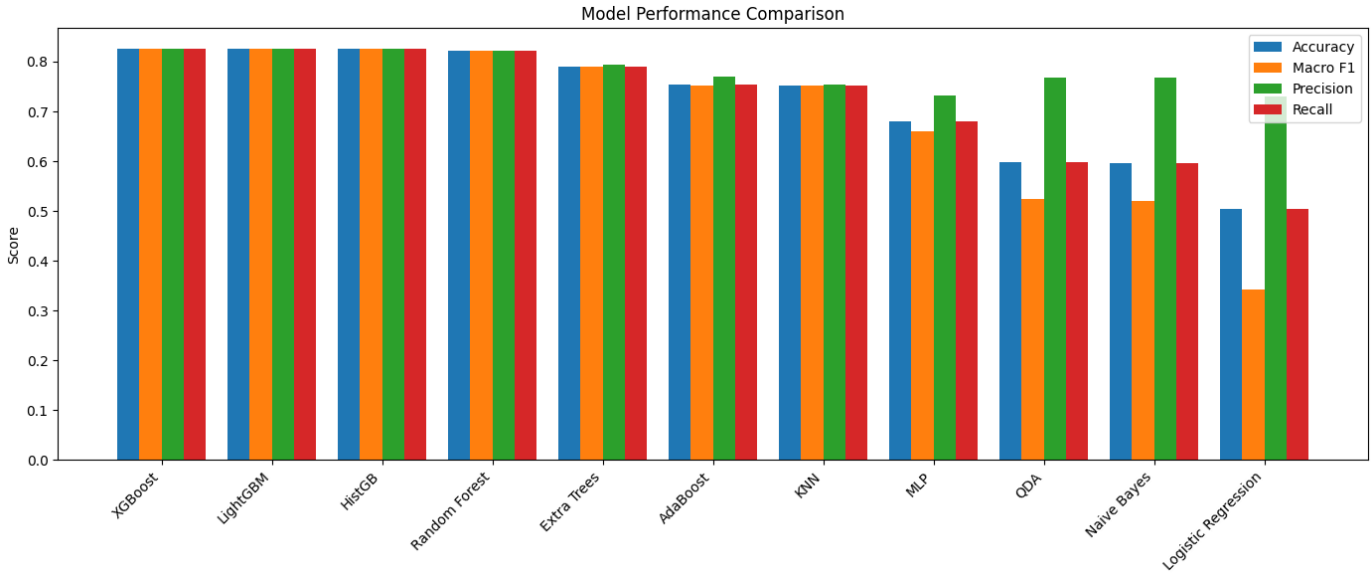

Figure 1: Chart of the performance indicators (score) per ML model. See the appendix for the exact numbers and a more extensive explanation on how these indicators are calculated and how hyperparameters were optimized. The Macro F1 score corresponds to the macro-averaged F1-score .

Intermediate-performance models
K-Nearest Neighbours, MLP Multilayer Perceptron, AdaBoost, and Extra Trees achieve moderate performance, accuracy values ranging from approximately 0.68 to 0.79, with the best-performing model in this group (Extra Trees) reaching approximately 0.79. While these models show improved balance between precision and recall, they are consistently outperformed by more advanced tree-based boosting methods.

These algorithms may serve as reference baselines but do not offer a competitive trade-off between predictive accuracy, scalability, and robustness.

High-performance models

The strongest performance is achieved by tree-based ensemble methods, including gradient boosting approaches (XGBoost, LightGBM, Histogram-Based Gradient Boosting) as well as bagging-based methods such as Random Forest

These models demonstrate:

- consistently high accuracy,
- balanced precision and recall for both classes,
- negligible differences between macro-averaged and weighted metrics, indicating stable and unbiased predictions.

Selected Algorithms for Further Development

Based on empirical evaluation, four algorithms have been selected for further analysis within WP11.

XGBoost

Rationale: Highest predictive performance with top accuracy and macro F1-score.
Strengths: State-of-the-art accuracy; models non-linear relationships; robust to noise; supports explainability (e.g., SHAP).
Limitations: Higher computational cost; complex implementation and tuning.
Role in WP11: Reference high-performance model for benchmarking and in-depth analysis.

LightGBM

Rationale: Comparable performance to XGBoost with superior computational efficiency for large datasets.
Strengths: Fast training and inference; low memory usage; scalable and suitable for production.
Limitations: Sensitive to feature binning and preprocessing; slightly less robust under high noise.
Role in WP11: Primary candidate for scalable, production-ready implementations.

Histogram-Based Gradient Boosting

Rationale: Balances performance, simplicity, and scikit-learn integration.
Strengths: Competitive accuracy; fast training via histogram discretization; compatible with Python ML pipelines; reproducible and transparent.
Limitations: Fewer tuning options; limited support for advanced explainability.
Role in WP11: Robust and transparent baseline model for reproducible statistical workflows.

Random Forest

Rationale: Robust to overfitting, handles diverse data, provides feature importance for interpretability.
Strengths: Resistant to overfitting; effective on noisy data; built-in feature importance; complements other models.
Limitations: Longer training on very large datasets; lower interpretability vs single trees.
Role in WP11: Complementary model providing stable predictions and additional variable insights.

Current Specifications of the Main Model

Based on the results, we consider LightGBM to be the main candidate for operational and large-scale deployment. Its predictive performance is comparable to that of XGBoost and it has superior computational efficiency and scalability.

The main advantages of LightGBM in this context are its computational efficiency, relatively low memory usage, suitability for large-scale operational deployment, and strong predictive performance.

Three other models will also be used in the next training phase, but in an auxiliary role.

Main Model Overview

The main predictive model used in this work package is LightGBM, a gradient boosting framework optimized for efficiency, scalability, and performance on structured datasets. This model was selected due to its strong empirical performance, interpretability relative to other ensemble methods, and capability to capture non-linear relationships in the data.

Hyperparameter Configuration

The current specifications of the LightGBM model, determined through Grid Search and Randomized Search cross-validation, are as follows:

Parameter	Value	Description
learning_rate	0.1	Controls the contribution of each tree to the final model; balances convergence speed with generalization.
max_depth	10	Limits the depth of individual trees to prevent overfitting while maintaining model expressiveness.
n_estimators	200	Number of boosting iterations; sufficient to ensure stable learning and improved predictive performance.
num_leaves	50	Maximum number of leaves per tree; balances model flexibility and stability.
random_state	42	Ensures reproducibility of experimental results.

Hyperparameter tuning was performed as part of the model evaluation stage and applied consistently across all evaluated models, using Grid Search and Randomized Search CV.

Performance Summary

The current configuration achieves high predictive performance on validation data, with an accuracy of 0.8255 and a macro F1-score of 0.8255, demonstrating the model's ability to capture the complex patterns of supplier–buyer relationships while maintaining generalization.

Approach for fine-tuning the main model and validating the results

As mentioned above, LightGBM is the main model that ultimately will provide the network reconstruction model that will be used for reconstructing firm-level network datasets for countries that do not have sufficiently rich datasets on buyer/supplier relationships.

XGBoost, Histogram-Based gradient Boosting and Random Forest will be used for helping to prepare the training data as well as the set-up of LightGBM in such a way that the results are optimal. XGBoost is retained as the reference model with the highest predictive performance and is primarily used for benchmarking and in-depth methodological analysis. Histogram-Based gradient Boosting and Random Forest are applied in a complementary manner. First of all to inform us better about the structure of the dataset and patterns in the data while also helping us understand the importance of variables and relations between them. This also supports improved feature engineering and parameter setting. Second, the other models can help in optimizing hyperparameter setting and robustness checks and finally help to assess potential overfitting and bias.

Preview: further fine-tuning in the network reconstruction phase

The final selected ML model will be subject to additional fine tuning to further improve the accuracy of the resulting network. As mentioned earlier, supply chain networks are sparse: only less than one percent of possible links actually exist. Hence, even if the accuracy of a trained model is high, only a limited number of predicted links will be true positives[5].

To fine tune the models, optimisation techniques can be performed based on the type of analysis a user performs. For most economic analyses, the trained model can be optimized with several measures. The current idea is to first calculate the raw link probabilities. Then we can calibrate them with the expected degree distribution, so with the distribution of incoming and outgoing links per firm, related to characteristics like firm size and/or NACE code. We will use the distribution from the Portuguese network dataset for this so-called 'probability calibration'. Subsequently, we run an edge sampling procedure to select the 'existing' links. This gives a reconstructed network that is consistent with several overall constraints and thus suited for e.g. studying the macro-economic effects of shocks, for instance related to a tariff rate.

For micro economic oriented analysis, a higher precision at the firm level is required. That can be done by optimizing the reconstruction model for firm-to-firm links, but that comes at the cost of losing some representativity of the structural characteristics of the network dataset as a whole. The gain, however, is that the user can study e.g. the characteristics of the 20 most vulnerable large firms.

Currently, we are experimenting with these more or less general ideas to see how we can best implement them in the reconstruction pipeline. The next deliverable will include the results of that work. The report on that task will contain a more specific elaboration and more concrete examples.

Nowcasting structure t+1 based on t

Originally, it was envisaged to also develop a method for nowcasting the network structure for year t+1 based on a model for year t combined with firm-level data on t+1. This method could not be tested since the firm-level network dataset for training and testing could only be made for one year. Therefore, it is not possible to reconstruct a network for a second year and compare it with the 'real' data for the second year. And thus we cannot measure how big the reconstruction errors are in such an approach and in which parts of the network their effects are most important.

However, there is a 'second best' solution, that can be tried once the reconstruction pipeline is fully operational. We can apply the reconstruction pipeline on t+1 input data and reconstruct the Portuguese network for t+1. We can aggregate that to the classification of the input/output table and calculate the ratios between the values of the IO table and the aggregated network. We already have these ratios for year t (2022). One might expect these ratios to be more or less stable in two subsequent years. Under that assumption, a comparison of the ratios between t and t+1 should provide an indication of the extent to which the reconstruction model can be safely applied to a next

[5] Mungo L, Lafond F, Astudillo-Estévez P and Doyne Farmer J 2023 Reconstructing production networks using machine learning J.Econ. Dyn. Control 148 104607, Esp. Appendix B

year. It also should indicate whether e.g. specific industry x industry combinations require additional information or measures to improve the nowcasting quality.

Link with other Workpackages

During the execution of this task, no active collaboration was needed with other Workpackages of the AIML4OS project. However, the discussion of the type of ML task and of the validation of the results touches upon the work of WP5 on the Total Machine Learning Error (TMLE) model. The WP11 approach seems to be an interesting candidate for applying the TMLE model and testing it in practice, but only once the reconstruction pipeline and the validation of the results are fully elaborated. As mentioned above, the reconstruction process contains some crucial steps for dealing with the complexity of this type of ML problem (i.e. the prediction of very rare events and the difference between the training and the target population). Therefore, such a test can only be done when the next two tasks are also completed.

Summary

In this report, we described the context, goals, challenges and approach for training ML models for reconstructing firm-level supply chain networks. The training set was created using a high quality Portuguese firm-level network dataset, the resulting model will be used as the 'engine' of a network reconstruction software pipeline that will also be applied in other countries and NSI's than Portugal.

A challenge of the ML training task applied here, is that existing firm to firm links are 'rare events'. We tackled that by creating a balanced training set in which a number of non-existing links is not taken into account.

We created a training software pipeline that allowed to train a series of models in an efficient way and used that on training and test sets derived from the Portuguese firm to firm network for 2022 that was created in an earlier task of WP11.

In total, we trained eleven different models and compared the results using several criteria. Based on the results, we consider LightGBM to be the main candidate for operational and large-scale deployment. Its predictive performance is comparable to that of XGBoost and it has superior computational efficiency and scalability. The main advantages of LightGBM in our case are its speed and relatively low memory use, suitability for operational deployment and of course it's high performance.

With this outcome, we arrive at the same conclusion as Mungo et al. (2023; 2024), that LightGBM is the best choice for this kind of prediction problems.

Three other models will also be used in the next training phase, but in an auxiliary role. XGBoost, Histogram-Based gradient Boosting and Random Forest will be used for helping to prepare the training data as well as the set-up of LightGBM in such a way that the results are optimal. XGBoost is retained as the reference model with the highest predictive performance and is primarily used for benchmarking and in-depth methodological analysis. Histogram-Based gradient Boosting and Random Forest are applied first of all to inform us better about the structure of the dataset and patterns in the data while also helping us understand the importance of variables and relations between them. That also helps to improve feature engineering and parameter setting. In addition, they can help in optimizing hyperparameter setting and robustness checks and finally help to assess potential overfitting and bias.

Based on the work of Mungo et al. (2023; 2024), we are aware that even with high accuracy, not all of the reconstructed existing links will actually be true – due to the 'rare event' problem. In the network reconstruction pipeline additional measures will be taken to make sure that the resulting networks are good enough for various types of analysis.

Appendix

Calculation of ML model performance indicators

In order to calculate performance indicators, the classification threshold is not explicitly defined in the implementation. Instead, the default threshold of 0.5 is implicitly applied through the predict() method of scikit-learn classifiers, which converts predicted probabilities into class labels using a standard decision boundary at 0.5.

The Area Under the Receiver Operating Characteristic Curve (AUROC) has been implemented in the current modelling pipeline. It is computed using predicted probability scores through the standard approach, where the false positive rate (FPR) and true positive rate (TPR) are derived via the roc_curve function, and the final AUROC value is obtained using the auc function. This implementation confirms that the metric is correctly calculated on probability estimates rather than class labels, and that it remains independent of the decision threshold. Consequently, AUROC is fully implemented and operational within the evaluation framework.

Table 2: Table with the performance indicators of Figure 1, per ML model.

Performance	***Model***	***Accuracy***	***Macro F1***	***Precision***	***Recall***
High	*XGBoost Classifier*	*0.8258*	*0.8258*	*0.8259*	*0.8258*
	LightGBM	*0.8255*	*0.8255*	*0.8256*	*0.8255*
	Histogram-Based Gradient Boosting	*0.8253*	*0.8253*	*0.8254*	*0.8253*
	Random Forest	*0.8227*	*0.8227*	*0.8229*	*0.8227*
Intermediate	*Extra Trees*	*0.7903*	*0.7895*	*0.7948*	*0.7903*
	AdaBoost	*0.7544*	*0.7510*	*0.7691*	*0.7544*
	K-Nearest Neighbours	*0.7529*	*0.7527*	*0.7538*	*0.7529*
	MLP Multilayer Perceptron	*0.6796*	*0.6602*	*0.7326*	*0.6796*
Low	*Quadratic Discriminant Analysis*	*0.5988*	*0.5236*	*0.7687*	*0.5988*
	Naïve Bayes	*0.5969*	*0.5204*	*0.7680*	*0.5969*
	Logistic Regression	*0.5038*	*0.3419*	*0.7304*	*0.5037*

Hyperparameter optimization

Hyperparameter optimization was conducted using a combined search strategy involving both GridSearchCV and RandomizedSearchCV. Both methods were executed independently but in parallel over identical model-specific hyperparameter spaces. The final selection of hyperparameters for each model was based on the configuration that achieved the highest mean cross-validation performance (Table 3), regardless of the search method from which it originated. This approach allowed for a balance between exhaustive search (GridSearchCV) and computational efficiency and broader exploration of the parameter space (RandomizedSearchCV).

Table 3: Overview of the used ML model hyperparameters.

Model	**Hyperparameter values**
K-Nearest Neighbors	Number of neighbours: n_neighbors=9
	Distance metric (e.g. Euclidean or Manhattan): metric=Manhattan
	Weighting strategy (e.g. uniform or distance): weight=uniform
Decision Tree	Maxiumum tree depth: max_depth=3

	Minimum samples required to split a node: min_samples_leaf=1
	Minimum samples per leaf: min_samples_split=2
Extra Trees	Number of estimators: n_estimators=50
	Maximum depth: max_depth=10
Logistic Regression	Regularization type: C=0.01
	Solver type: solver=liblinear
	Maximum number of iterations: max_iter=1000
MLP Classifier	Hidden layer classifiers: hidden_layer_sizes=(50,)
	Activation functions: activation=relu
	L2 regularization parameter: alpha=0.0001
Quadratic Discriminant Analysis	regularization parameter: reg_param = not provided
Histogram-Based Gradient Boosting	Learning rate: learning_rate=0.1
	Maximum iterations: max_iter=200
XGBoost Classifier	Learning rate: learning_rate=0.2
	Maximum depth: max_depth=7
	Number of estimators: n_estimators=100
	Subsampling ratio: subsample=0.7
LightGBM	Learning rate: learning_rate=0.1
	Maximum depth: max_depth=10
	Number of leaves: num_leaves=50
AdaBoost	Learning rate: learning_rate=0.01
	Number of estimators: n_estimators=50
Random Forest	Number of estimators: n_estimators=50
	Maximum depth: max_depth=5