Predicting Risk of Multiple Sclerosis Worsening

Notebook for the iDPP Lab on Intelligent Disease Progression Prediction at CLEF 2023

Marek Hanzl[1], Lukáš Picek[1]

[1]*University of West Bohemia, Faculty of Applied Sciences, Czech Republic*

Abstract

This paper describes our participation in the first two tasks of the iDPP@CLEF 2023 challenge focused on providing clinicians with AI-based methods for better prediction of Multiple Sclerosis progression. We evaluate several standard and transformer-based methods, e.g., Random Forest, Gradient Boosting, and SurfTRACE transformer, to address the risk and cumulative probability of Multiple Sclerosis worsening. The considerable performance increase was achieved by (i) hyper-parameter fine-tuning, (ii) validation procedure, and (iii) data pre-processing. The best method based on the Random Forest algorithm scored first place in Task 1 and 2 (sub-task A) with a C-Index of 0.834, and a mean AUROC score of 0.881, respectively, while reducing the runner-up's error by 16.2% and 2.3%, respectively. Our methods purely designed and optimized for sub-task A and submitted into sub-task B showed considerable robustness towards overfitting on a specific dataset as achieved third and second place and achieving 0.601 C-Index and second in Task 2, sub-task B of 0.607 mean AUROC score.

Keywords

Multiple Sclerosis, Artificial Intelligence, Survival Analysis, Gradient Boosting, Transformers

1. Introduction

Multiple Sclerosis (MS) is a chronic autoimmune disease characterised by progressive impairment of neurological functions, leading to a patient's gradual loss of motor, sensory, visual, or cognitive capabilities [1, 2]. It can lead to various physical and mental symptoms, while the progression and severity vary widely between individuals. The MS is widely spread and affects primarily young adults, especially women. Most people diagnosed are between 20 to 50 years old [3]. To provide context: Over 2.8 million people suffer from MS worldwide, and around 300 people are diagnosed with MS daily. The overall costs related to MS treatment were estimated in 2019 at $85.3 billion [4]. The heterogeneity of each patient's progression immensely increases the difficulty of selecting a proper medical treatment and motivates further clinical research. It urges a need for novel and reliable methods that should assist the clinical decision-making process and help to advance the development of effective treatments, leading to better care of patients.

The iDPP@CLEF 2023 [5, 6] lab aims to address this task by opening international challenges. Proposed challenges aim to provide an evaluation ground for developing new Artificial Intelligence (AI) and Machine Learning (ML) techniques which could detect patient complications

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

✉ hanzl@students.zcu.cz (M. Hanzl); picekl@kky.zcu.cz (L. Picek)

early, stratify individuals according to their risk levels, and predict disease progression over time. The 2023 edition of the iDPP competition offered three tasks:

1. *Predicting risk of disease worsening.*
2. *Predicting the probability of worsening at different time windows.*
3. *Impact of Exposition to Pollutants* – patients with Amyotrophic Lateral Sclerosis.

In this paper, we describe our submissions in the first two tasks. The first task requires ranking subjects based on the risk of worsening, while the second task specifies the predictions by explicitly assigning the cumulative probability of worsening at different time intervals. Both provide pre-computed data of occurrence and time of worsening. These tasks are divided into two sub-tasks: A and B, which differ in the definition of worsening based on the Expanded Disability Status Scale (EDSS) [7].

To address the risk and cumulative probability of Multiple Sclerosis worsening, we evaluate several AI-based standard survival analysis methods, which include Random Forest, Gradient Boosting models [8], and a recent SurfTRACE transformer model [9]. Furthermore, we provide an extensive overview of the hyper-parameter fine-tuning of these selected methods. In a few cases, we combine these models to create our ensemble model methods.

To allow robust evaluation of overall performance, we use several metrics selected by the competition organization, i.e. Harrell's Concordance Index (C-Index) [10] for Task 1, AUROC curve [11] and O/E ratio [12] for Task 2. We chose model runs with the highest validation score, i.e., the highest C-Index, to make the final prediction for the submission files.

2. Data

In this section, we analyze the given data for each respective task and address their specific issues. Subsequently, we provide a detailed description of the pre-processing steps taken to maximize the prediction precision of ML models.

2.1. Data characteristics

The datasets provided by competition organizers [5, 6] consist of static and dynamic data. They include the medical history of 1,192 patients from two clinical institutions in Italy located in Pavia and Turin. The dynamic data span over a period of 2.5 years and are split into different subsets. These comprise information on relapses, EDSS, Evoked Potentials (EP), MRIs, and the MS course. The ground-truth data, i. e. the outcomes, include the actual occurrence of worsening and the relative time of this occurrence. The provided datasets for both sub-tasks (A, B) were split into training and test sub-sets in approximately 80/20 ratio. Unlike the training datasets, the test datasets do not contain any information about the true outcomes. These remained unpublished until the time after the submission deadline.

The sub-task A provides complete 440 unique patients for training and 110 for testing. In the case of sub-task B, information on 510 unique patients is available for training and 128 for testing. Even though the number of patients is relatively high, only a fraction of them includes all types of medical records. For reference, there are only 103 (23.4%) unique patients in the training dataset A and 155 (30.4%) in the dataset B with medical records from all dataset sub-sets.

Table 1
Numbers of unique patients and medical records in data sub-sets which are included in the provided A and B datasets.

	Dataset	Static Vars.	Outcomes	EDSS	EP	Relapses	MRI	MS Type
Patients	A	440	440	439	153	259	279	210
Records		440	440	2,661	1,211	481	960	310
Patients	B	510	510	510	183	284	303	218
Records		510	510	3,069	1,522	553	966	325

Each medical record refers to a single entry (row) in the dataset. The number of unique patients and the number of their medical records for the dataset subsets are listed in Table 1.

It is apparent that the amount of data present differs widely between patients, as it is indicated in Table 1. This significant data imbalance poses an issue in the classification stage, as standard classifiers face challenges when dealing with the class imbalance and often prioritize the larger classes and disregard the smaller ones, reaching a sub-optimal solution [13]. Although, the imbalance is likely inherent to the problem. Furthermore, severe time gaps between clinical visits of many patients are present, leaving out important information about the time progression of the disease.

2.2. Data pre-processing

The success of ML in achieving optimal performance on a given task relies on various factors, with the representation and quality of the instance data being of critical importance. Effective knowledge discovery during training phases becomes arduous when irrelevant and redundant information or noisy and unreliable data are present. Data preparation and filtering steps, including data cleaning, normalization, transformation, feature extraction, and selection, are vital in achieving the best validation performance on the specific dataset [14].

In our case, the ruling factor for feature and pre-processing method selection, the C-Index validation performance on the dataset A was employed. The premise is that the data should be almost equivalent except for the EDSS feature definition, meaning the particular decision considering pre-processing should be generally valid for both cases.

First, all available information in the medical records is loaded from provided dataset files and subsequently grouped by unique patient id. To be precise, many patients have multiple medical records based on the number of repetitions of different medical examinations. In this manner, it is ensured that all the available information concerning each patient will be present.

The features are divided into several groups derived from their characteristics, e.g., static and time-dependent features, and categorical and numerical features. Static variables represent the time-invariant features. If these are likewise categorical, then the one-hot encoding is applied, which elegantly solves the issue of missing values. Currently, these features are the patient's sex, residence, ethnicity, centre, MS diagnosed in pediatric age, and the record of the presence of several symptoms derived from their physical location. Whereas, the *"time_since_onset"* and *"diagnostic_delay"* features generally resulted in poorer performance, and were, therefore, omitted.

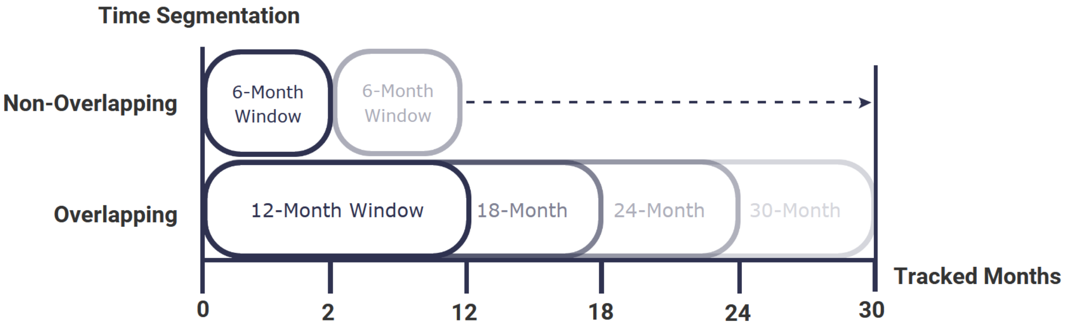

Figure 1: Visualisation of segmenting time-window features of the dynamic, time-dependent data. For each dynamic feature (e.g. EDSS score), five 6-month non-overlapping and four accumulative (overlapping) time windows were created.

In the case of time-dependent features, it is crucial to extract the temporal context of measured values. For this goal, a *"sliding time-window"* segmentation approach [2] is implemented, where a 6-month-long non-overlapping time window, together with a series of gradually extending, cumulative time windows (6, 12, 18, 24, and 30 months) are applied. This process is visualised in Figure 1. Upon the segmented features, we apply various statistical functions, mainly mean, one standard deviation, median, mode, and in some cases, the sum of all occurrences of a feature, are calculated. In the case of the EDSS and Relapses dataset sub-sets, the mean, one standard deviation, median, and mode of the EDSS score and the relapse occurrences are computed, respectively. Meanwhile, in the Evoked Potentials sub-set, we calculate a sum of all occurrences where the altered potential feature was positive and a sum of all occurrences where the altered potential feature was both positive and negative. In this way, we aim to capture information about the percentage of diagnosed altered potentials.

In the MS-type dataset section, generally, no more than two different recordings were present for each patient. These determine the diagnosed MS type and the time of the diagnosis. We have attempted to one-hot encode each MS-type record and added the time of diagnosis. However, this did not lead to any improvement in performance, and thus, the data were omitted. The application of the MRI dataset section mostly led to similar results, and with a single exception, the data were omitted too.

In the last step, the missing feature values were handled. The numerical features were normalized to reduce the scale of the data, leading to an improvement in the numerical qualities of the dataset. We decided to retain the maximum amount of data available. Therefore, all the missing values were filled via the use of *Fast.ai* [15] functions, namely *TabularPandas*. This function creates a new categorical feature for every feature with at least one missing value. In this new feature, there are two categories representing missing or not missing values. In the original feature, the missing values are filled with a median of the whole feature. Afterwards, they are normalized by subtracting the mean and dividing by one standard deviation, which are both derived from the newly filled feature. Additionally, we have attempted to use different filling methods of time-dependent values for each patient. However, this led to worse overall prediction performance.

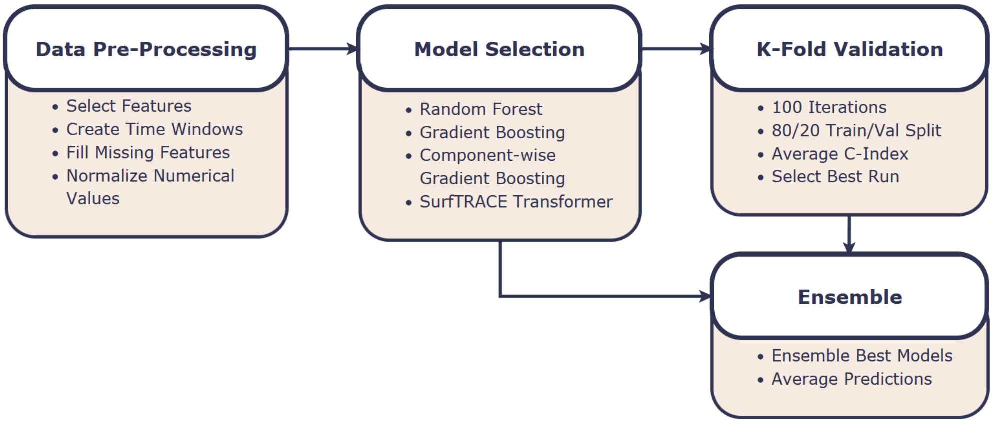

Figure 2: Machine-learning workflow. Stages of method training and validation. Methods consist of data pre-processing, selection of the specific model and k-fold validation based on C-Index performance in 100 iterations. Afterwards, additional methods are created by ensembling the models of all types.

3. Methodology

In this section, we focus on describing the steps of assembling our Machine Learning pipeline used to produce the predictions of the MS progression. To generate these predictions, it is essential to undergo a data pre-processing step, followed by selecting appropriate ML models capable of performing these predictions. Subsequently, we specify the methods employed for training, validating, and fine-tuning the models' hyper-parameters, which are crucial to achieving the best and yet explainable results.

We selected several survival analysis models, as the prediction of worsening can fit in the statistical branch of survival analysis, which focuses on studying the occurrence and time of a specific event. Survival analysis offers significant value by addressing two common challenges that conventional statistical methods often struggle with, namely censoring and time-dependent covariates [16, 17]. Our workflow is visualised in Figure 2. The data analysis, data pre-processing, and feature selection steps were already described in the previous section.

3.1. Models

To select the most viable method, several different approaches were tested, mainly various survival analysis models like linear models, ensembles, or survival support vector machines from the *scikit-learn* library – the *scikit-survival* [8]. From these, we have selected three ensemble models with the most positive and stable performance:

- *Random Survival Forest*
- *Gradient Boosting Survival Analysis*
- *Component-wise Gradient Boosting Survival Analysis* (CGBSA)

Furthermore, we used a recent transformer [18] model suited for survival analysis, the *SurfTRACE* [9] transformer. A large, correctly trained transformer model should be able to contain all the available information, i.e., data and their time context, in the features of the specified topic. In theory, this should lead to increased prediction capabilities.

Additionally, we created an ensemble model of our own by combining the best run of every model type and averaging their predictions. This should lead to a compensation of extreme predictions made by these models, improving their stability. Likewise, we attempted to use the maximal predicted values for prediction instead. However, this approach performed worse than the averaged predictions and was omitted.

3.2. Training and validation strategy

The training and validation procedure is described as follows: Each model was trained and validated in 100 iterations. In each iteration, the pre-processed dataset is randomly split into training and validation sets in an 80/20 ratio (i.e., the same ratio as in provided training and test datasets). The C-Index is measured for both training and validation sets. Afterwards, the best model is chosen based on the achieved average and one standard deviation of the C-Index calculated from all the iterations. Furthermore, we attempted to use the *MinVal* 95/5 split ratio for several runs. Meaning, models are provided with an even larger proportion of available data in the training stage. We perform multiple iterations to mitigate the effect of randomly split data, as it is likely that some splits are easier for model fitting and can lead to unreasonably high performance. Thus, it is better to evaluate model performance by averaging multiple runs, reducing the effect of randomness. The transformer was trained with Adam optimizer for 40 epochs with 10-epoch early-stopping.

Finally, we select the run (one from 100 iterations) with the highest C-Index value to make the final predictions, i.e., predictions for the submission. To remain unbiased, we used the same seed for every run.

3.3. Hyper-parameter fine-tuning

The selected models start with multiple available hyper-parameters, whose initial, general settings are presumably inadequate for the current task performance. Meaning, optimal model settings can be found through extensive iterative hyper-parameter search and lead to further maximization of validation performance.

The Weights & Biases framework [19] was used for this task. We mainly used the *"Sweeping tool"* with mixed random and grid search based on the C-Index performance on the dataset A. To explain, first, the random search over a wider range of parameters was applied to localize roughly optimal intervals. These intervals were then exhaustively grid-searched. The list of all pre-selected search hyper-parameters is in Table 2[1]. In total, approximately 3,500 separate runs were performed.

After selecting the optimal hyper-parameter values, we achieved marginal improvements for standard models, which likely indicates that the key part to achieving better performance lies in proper data pre-processing. On the other hand, the fine-tuning proved critical for the transformer's performance. Although, the transformer predictions remained still quite volatile. The complete results are available in Table 3.

[1] Default values were used for all non-mentioned Hyper-parameters. Please refer to Scikit-Survival documentation.

Table 2
Tested hyper-parameter values for selected methods. Evaluation of performance was based on achieved C-Index values on the dataset A.

ML Method	Hyper-parameters	Values	Search method
Random Forest	n_estimators	[100, 300, 500]	Grid search
	max_depth	[6, 8, 10]	
	min_samples_split	[8, 10, 15]	
	min_samples_leaf	[4, 6, 8]	
	min_weight_fraction_leaf	[0.0, 0.3]	
	max_features	[sqrt, log2]	
Gradient Boosting	n_estimators	[100, 200]	Grid search
	subsample	[0.2, 0.5, 1]	
	dropout_rate	[0, 0.2]	
	ccp_alpha	[0, 0.1, 1]	
	learning_rate	[0.1, 0.5]	
	max_depth	[3, 5, 7]	
	min_samples_split	[2, 4]	
	min_weight_fraction_leaf	[0.0, 0.3]	
	max_features	[sqrt, log2]	
CGBSA	loss	[coxph, squered, ipcwls]	Grid search
	n_estimators	[100, 200, 300, 500]	
	learning_rate	[0.1, 0.5, 1]	
	subsample	[0.2, 0.5, 1]	
	dropout_rate	[0, 0.2, 1]	
SurfTRACE	batch_size	(48–128)	Random search
	weight_decay	(5e^{-5}–1e^{-3})	
	learning_rate	(5e^{-4}–1e^{-2})	
	hidden_size	[16, 32, 64]	
	intermediate_size	[64, 128, 256, 512]	
	num_hidden_layers	[2, 4, 6]	
	num_attention_heads	[2, 4, 6]	
	hidden_dropout_prob	(0.2–0.4)	
	attention_probs_dropout_prob	(0.1–0.3)	

Table 3
Improvement of the methods due to the hyper-parameter search measured by the average validation C-Index performance on the dataset A.

	Random Forest	Random Forest MRI	Gradient Boosting	Component-wise Gradient Boosting	SurfTRACE
Before	0.731	0.738	0.741	0.719	0.561
After	0.741	0.747	0.750	0.725	0.698
Δ	+0.95%	+0.93%	+0.92%	+0.67%	+15.36%

3.4. Validation results

Here we discuss the validation results of the best-performing methods measured by the C-Index value on the validation set for both datasets (A, B). We used previously discussed models (Random Forrest, Gradient Boosting, Component-wise G. B., SurfTRACE transformer, and averaging Ensemble model) together with the *Random Forest MRI*. In this method, an extended version of pre-processed data was used, which included specific MRI data. The overall validation performance after 100 independent iterations is visible in Figure 3.

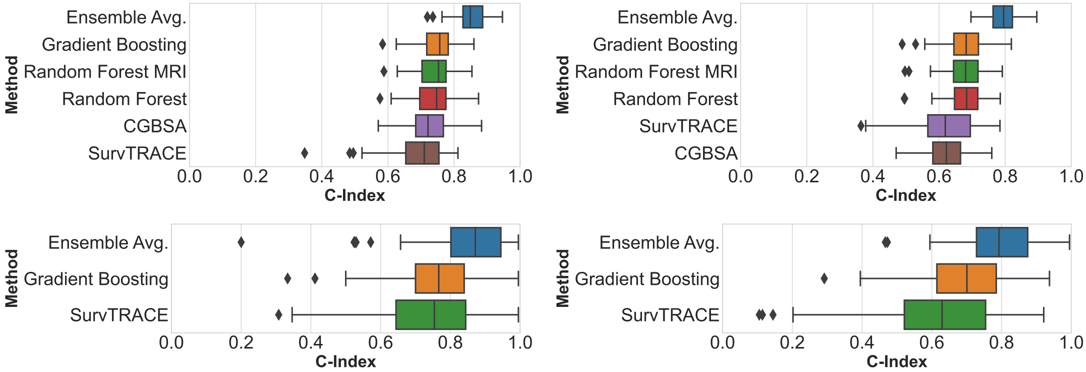

Figure 3: Results of validation C-Index performance of selected methods after 100 iterations. The dataset A (left) and B (right). Baseline (first row) and MinVal (second row) splits.

In the case of the dataset A, the averaging Ensemble model achieved the top performance. This outcome is explainable by the fact that the model already consists of the pre-trained, most well-performing models and is only tested for prediction accuracy. Compared to the others, the transformer suffers from substantial deviation, likely due to insufficient convergence and early stopping. The performance of the dataset B is generally worse than that of the dataset A. Partially due to the optimisation being solely done on the dataset A, but the largest impact is presumably caused by the definition of the EDSS score, or by the different data distribution.

For a few submissions, a *MinVal* strategy was tested, where the data were split into training and validation sets in a 95/5 ratio during the K-Fold training. The rationale is that with a larger size of the training set, more information is provided to the models in the training stage, possibly decreasing the likelihood of over-fitting. The results are available in Figure 3.

From the before-mentioned methods, the runs with the best overall validation C-Index were used to make predictions for final submissions for both tasks and their respective sub-tasks. In total, 18 files for Task 1 and 10 files for Task 2 were submitted.

4. Results

In this section, we provided official results of the submitted runs for Task 1 and Task 2 and their respective sub-tasks. The submissions were selected in part based on the best achieved C-Index values and to test all the previously mentioned models and different strategies. We discussed the achieved results and attempted to explain them. In the last part, we compared the best runs of the top 5 teams participating in the competition.

The names of the displayed methods correspond to the model names. Concerning the competition, the names correspond to the *freefield* part of the submission names described in the competition naming convention. More precisely, survRf is the Random Forest, survGB is the Gradient Boosting, and AvgEnsemble is Ensemble Avg. Rest remains more the same.

4.1. Task 1

Here we discuss the results of the prediction accuracies described by the C-Index values and their 95% confidence intervals of the submissions made to the Task 1, the sub-tasks (datasets) A and B. The results are displayed in Figure 4. Models are sorted descendingly, with the highest score of 0.834 *(0.741–0.927)* achieved by the Random Forest MRI model. In the second place, the Averaging Ensemble model scored 0.828 *(0.739–0.917)*. Other models performed similarly well too. In the first place for the dataset B comes the SurfTRACE transformer with a C-Index of 0.601 *(0.482–0.721)*. The worst performance in both cases was achieved by the SurfTRACE transformer. This is most likely due to a selection of badly converged model runs for the submission. The sub-optimal convergence is explainable by a model over-fitting over specific data divisions. Compared to the validation performance, the runs of the dataset B scored lower than expected. A possible explanation would be that the test dataset consisted of data differing in distribution from the training dataset.

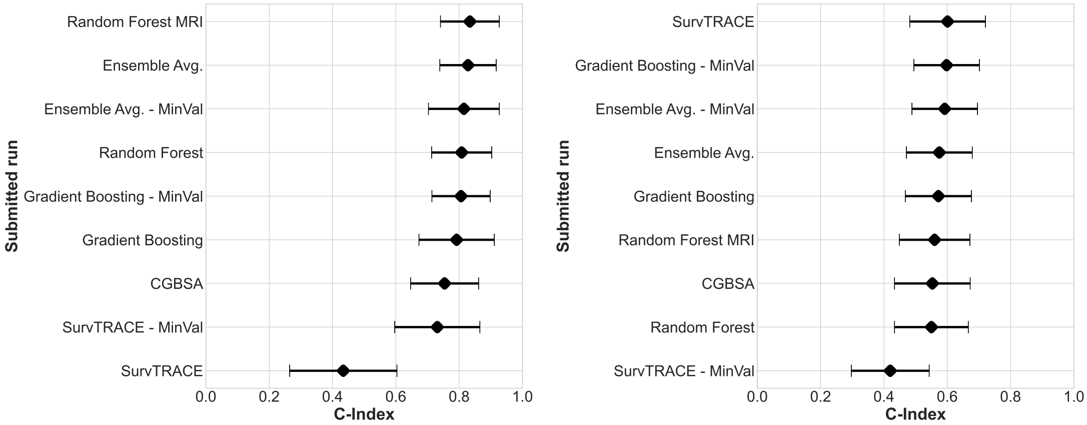

Figure 4: Achieved C-Index values and respective 95% confidence intervals for submitted methods on both sub-tasks. Left sub-task A (Task 1), right sub-task B (Task 1).

4.2. Task 2

In this section, we provide the results of the prediction accuracies of the submission files in the Task 2. The scores consist of cumulative AUROC values, O/E ratios, and their respective 95% confidence intervals of all submitted methods. These are separated into five columns for both metrics. The first describes achieved scores of prediction of the cumulative probability of worsening in the first 2 years, the second in the first 4 years, etc. Methods are sorted in descending order based on the average AUROC score of all predicted time intervals. In the tables, the following name enumeration is applied:

- **I** – Random Forest MRI
- **II** – Random Forest
- **III** – Gradient Boosting - MinVal
- **IV** – Gradient Boosting
- **V** – Component-wise Gradient Boosting

The Random Forest MRI method scored the highest, with an average AUROC of 0.881, while scoring the best in Task 1. The O/E ratio, which represents the ratio of observed to expected events at each time interval, is relatively high across all the predictions. This likely indicates a practical issue when there is a greater portion of observed patient events than was expected. All achieved scores in Task 2, sub-task A are displayed in Table 4.

Table 4
AUROC and O/E Ratio cumulative scores in each given interval for all submitted methods. Results of submissions to sub-task A (Task 2), with the ID specified in the preceding text.

ID	AUROC 0–2	AUROC 0–4	AUROC 0–6	AUROC 0–8	AUROC 0–10
I	0.924 *(0.800–1.000)*	0.907 *(0.816–0.998)*	0.896 *(0.801–0.991)*	0.838 *(0.713–0.964)*	0.839 *(0.699–0.979)*
II	0.914 *(0.784–1.000)*	0.893 *(0.798–0.989)*	0.898 *(0.808–0.989)*	0.828 *(0.702–0.954)*	0.820 *(0.672–0.968)*
III	0.894 (*0.787–1.000*)	0.898 *(0.810–0.985)*	0.901 *(0.800–1.000)*	0.818 *(0.677–0.959)*	0.808 *(0.648–0.967)*
IV	0.877 *(0.745–1.000)*	0.891 *(0.796–0.986)*	0.868 *(0.753–0.984)*	0.790 *(0.641–0.938)*	0.812 *(0.654–0.969*)
V	0.862 (*0.731–0.993*)	0.842 *(0.713–0.971)*	0.805 *(0.670–0.941)*	0.747 *(0.597–0.898)*	0.798 *(0.649–0.947)*
ID	**O/E Ratio 0–2**	**O/E Ratio 0–4**	**O/E Ratio 0–6**	**O/E Ratio 0–8**	**O/E Ratio 0–10**
I	1.889 *(0.937–2.842)*	2.339 *(1.391–3.287)*	1.797 *(1.068–2.525)*	1.731 *(1.065–2.396)*	1.447 *(0.875–2.019)*
II	1.811 *(0.879–2.744)*	2.283 *(1.347–3.220)*	1.796 *(1.067–2.524)*	1.732 *(1.066–2.398)*	1.458 *(0.884–2.032)*
III	0.946 *(0.272–1.620)*	1.759 *(0.937–2.581)*	1.768 *(1.045–2.490)*	1.926 *(1.224–2.628)*	1.658 *(1.046–2.270)*
IV	0.919 *(0.255–1.583)*	1.831 *(0.993–2.670)*	1.739 *(1.022–2.455)*	1.906 *(1.207–2.604)*	1.644 *(1.035–2.254)*
V	3.106 *(1.885–4.327)*	1.975 *(1.104–2.847)*	1.366 *(0.731–2.002)*	1.312 *(0.732–1.891)*	1.467 *(0.891–2.042)*

The results for Task 2, sub-task B are available in Table 5. The Gradient Boosting - MinVal method scored first with a mean AUROC of 0.607. There is observed a similar trend as in Task 1, sub-task B, where the scores worsen across all predictions. This is likely due to the same reason as previously discussed. These were: optimisation for dataset A, different definitions of the EDSS feature, and possibly different properties of the test sub-dataset.

4.3. Competition results

To evaluate the improvements provided by our work, we compared the achieved results with other participating teams. We selected the runs with the best performance from the top 5 running teams for each task and their respective sub-tasks. For the first task, the scores are sorted by the highest achieved C-Index in Table 6. Our methods managed to score first place in sub-task A and third in sub-task B. These are great results, meaning we can provide the best available performance for the optimized dataset. Moreover, the methods provide comparable performance even for not-optimized datasets, leaving substantial space for further improvement.

For Task 2, the achieved scores are displayed in Table 7. In sub-task A, our method achieved the best AUROC score with an average of 0.881, which is just slightly ahead of the next competitor. In sub-task B, we scored second place with an average AUROC score of 0.607, which is about 5.24% worse than the first competing team.

Table 5
AUROC and O/E Ratio cumulative scores for sub-task B (Task 2). Method ID is described in the text.

ID	AUROC 0–2	AUROC 0–4	AUROC 0–6	AUROC 0–8	AUROC 0–10
III	0.606 *(0.437–0.776)*	0.612 *(0.468–0.756)*	0.602 *(0.451–0.754)*	0.587 *(0.433–0.742)*	0.626 *(0.465–0.787)*
V	0.514 *(0.311–0.717)*	0.580 *(0.423–0.737)*	0.604 *(0.452–0.756)*	0.627 *(0.477–0.777)*	0.628 *(0.463–0.793)*
IV	0.569 *(0.392–0.747)*	0.597 *(0.454–0.741)*	0.589 *(0.440–0.737)*	0.580 *(0.427–0.733)*	0.594 *(0.430–0.758)*
I	0.596 *(0.421–0.770)*	0.561 *(0.407–0.715)*	0.559 *(0.407–0.711)*	0.525 *(0.369–0.681)*	0.491 *(0.324–0.658)*
II	0.590 *(0.410–0.769)*	0.552 *(0.401–0.704)*	0.549 *(0.398–0.700)*	0.522 *(0.367–0.678)*	0.506 *(0.340–0.672)*

ID	O/E Ratio 0–2	O/E Ratio 0–4	O/E Ratio 0–6	O/E Ratio 0–8	O/E Ratio 0–10
III	0.920 *(0.353–1.486)*	1.228 *(0.716–1.740)*	1.375 *(0.896–1.854)*	1.430 *(0.979–1.880)*	1.489 *(1.052–1.926)*
V	1.818 *(1.021–2.615)*	0.774 *(0.367–1.180)*	1.515 *(1.012–2.017)*	1.295 *(0.866–1.724)*	1.166 *(0.780–1.553)*
IV	1.045 *(0.441–1.649)*	1.259 *(0.741–1.778)*	1.363 *(0.886–1.840)*	1.404 *(0.957–1.850)*	1.454 *(1.022–1.885)*
I	2.257 *(1.370–3.145)*	1.525 *(0.955–2.096)*	1.364 *(0.886–1.841)*	1.302 *(0.872–1.732)*	1.316 *(0.905–1.726)*
II	2.292 *(1.398–3.187)*	1.523 *(0.953–2.093)*	1.351 *(0.876–1.826)*	1.304 *(0.873–1.734)*	1.313 *(0.903–1.724)*

Table 6
Official competition results – risk of worsening – for top 5 teams for Task 1, Sub-task A and Task 1, Sub-task B sorted by C-Index. Our results are in bold.

Rank	Submissions	C-Index
1.	**uwb_T1a_survRFmri**	**0.834**
2.	CBMUniTO_T1a_coxnet	0.802
3.	fcool_T1a_RandomSurvivalForest	0.801
4.	HULATUC3M_T1a_survcoxnet	0.774
5.	sisinflab-aibio_T1a_RF2	0.771

Rank	Submissions	C-Index
1.	fcool_T1b_FastKernelSurvivalSVM	0.690
2.	CBMUniTO_T1b_coxnet	0.634
3.	**uwb_T1b_SurvTRACE**	**0.601**
4.	uhu-etsi-1_T1b_s02	0.598
5.	sisinflab-aibio_T1b_GB2	0.587

Table 7
Official competition results – cumulative probability of worsening – for top 5 teams. Average AUROC score for Task 2, Sub-task A and Task 2, Sub-task B . Our results are in bold.

Rank	Submission	2 years	4 years	6 years	8 years	10 years
1.	**uwb_T2a_survRFmri**	**0.924**	**0.907**	**0.896**	**0.838**	**0.839**
2.	HULATUC3M_T2a_survcoxnet	0.864	0.898	0.938	0.859	0.831
3.	CBMUniTO_T2a_coxnet	0.890	0.900	0.856	0.787	0.796
4.	sisinflab-aibio_T2a_RF1	0.754	0.873	0.871	0.746	0.745
5.	uhu-etsi-1_T2a_05	0.774	0.740	0.774	0.703	0.722

Rank	Submission	2 years	4 years	6 years	8 years	10 years
1.	CBMUniTO_T2b_cwgbsa	0.632	0.626	0.655	0.673	0.709
2.	**uwb_T2b_survGB_minVal**	**0.606**	**0.612**	**0.602**	**0.587**	**0.626**
3.	sbb_T2b_Cox	0.642	0.567	0.601	0.594	0.622
4.	sisinflab-aibio_T2b_GB2	0.614	0.639	0.629	0.616	0.527
5.	uhu-etsi-1_T2b_s02	0.644	0.590	0.610	0.567	0.609

5. Conclusion

In this work, we described our participation in the iDPP 2023 challenge focused on providing clinicians with new ways of predicting the progression of Multiple Sclerosis. We provided an extensive overview of the methods and the results. We fully described our data analysis, preprocessing, and feature selection methods. Likewise, we described the steps taken to assemble the Machine Learning pipeline. We selected the best-performing survival-analysis-based models consisting of Random Forest and Gradient Boosting decision tree methods, together with the recent SurvTrace transformer. To further improve baseline performance, we conducted an extensive hyper-parameter search for selected methods and described the benefits of the fine-tuning, i.e., 16% improvement of transformer predictions and around 1% improvement of the others. The performance was measured by the achieved validation C-Index average after 100 separate iterations, which was a ruling factor for most of the decisions. The runs with the highest value were then used to make the final predictions.

We provide achieved scores for all of our submissions in Figure 4, and in Table 6, and Table 7. Furthermore, we compared our results with the top 5 teams. In terms of to C-Index and AUROC average, we scored first place in both A sub-tasks with a C-Index of 0.834 and a mean AUROC score of 0.881, respectively. In the case of sub-tasks B, we achieved results comparable with others. In Task 1 we scored third with a C-Index of 0.601, and we scored second in Task 2 based on the average AUROC score of 0.607. We showed considerable robustness towards overfitting on a specific dataset, as we achieved third and second place, even though our methods were purely designed and optimized for sub-task A. This likely implies the importance of the EDDS score definition as in the first case, the methods performed on average much better than in the case of the dataset B.

6. Acknowledgment

The work has been supported by the grant of the University of West Bohemia, project No. SGS-2022-017. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth, and Sports of the Czech Republic.

References

[1] M. M. Goldenberg, Multiple sclerosis review, P & T : a peer-reviewed journal for formulary management 37 (2012) 175–84.

[2] M. F. Pinto, H. Oliveira, S. Batista, L. Cruz, M. Pinto, I. Correia, P. Martins, C. Teixeira, Prediction of disease progression and outcomes in multiple sclerosis with machine learning, Scientific Reports 10 (2020).

[3] J. C. Brust, Current diagnosis & treatment neurology, McGraw Hill Professional, 2018.

[4] C. Walton, R. King, L. Rechtman, W. Kaye, E. Leray, R. A. Marrie, N. Robertson, N. L. Rocca, B. Uitdehaag, I. van der Mei, M. Wallin, A. Helme, C. A. Napier, N. Rijke, P. Baneke, Rising prevalence of multiple sclerosis worldwide: Insights from the atlas of ms, third edition, Multiple Sclerosis Journal 26 (2020) 1816–1821. PMID: 33174475.

[5] G. Faggioli, A. Guazzo, S. Marchesin, L. Menotti, I. Trescato, H. Aidos, R. Bergamaschi, G. Birolo, P. Cavalla, A. Chiò, A. Dagliati, M. de Carvalho, G. M. Di Nunzio, P. Fariselli, J. M. García Dominguez, M. Gromicho, E. Longato, S. C. Madeira, U. Manera, G. Silvello, E. Tavazzi, E. Tavazzi, M. Vettoretti, B. Di Camillo, N. Ferro, Intelligent Disease Progression Prediction: Overview of iDPP@CLEF 2023, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, A. Giachanou, A. Li, D. abd Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), Lecture Notes in Computer Science (LNCS), Springer, Heidelberg, Germany, 2023.

[6] G. Faggioli, A. Guazzo, S. Marchesin, L. Menotti, I. Trescato, H. Aidos, R. Bergamaschi, G. Birolo, P. Cavalla, A. Chiò, A. Dagliati, M. de Carvalho, G. M. Di Nunzio, P. Fariselli, J. M. García Dominguez, M. Gromicho, E. Longato, S. C. Madeira, U. Manera, G. Silvello, E. Tavazzi, E. Tavazzi, M. Vettoretti, B. Di Camillo, N. Ferro, Overview of iDPP@CLEF 2023: The Intelligent Disease Progression Prediction Challenge, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), CLEF 2023 Working Notes, CEUR Workshop Proceedings (CEUR-WS.org), ISSN 1613-0073., 2023.

[7] J. F. Kurtzke, On the evaluation of disability in multiple sclerosis, Neurology 11 (1961) 686–686.

[8] S. Pölsterl, scikit-survival: A library for time-to-event analysis built on top of scikit-learn, Journal of Machine Learning Research 21 (2020) 1–6.

[9] Z. Wang, J. Sun, Survtrace: Transformers for survival analysis with competing events, in: Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, 2022, pp. 1–9.

[10] J. Harrell, Frank E., R. M. Califf, D. B. Pryor, K. L. Lee, R. A. Rosati, Evaluating the Yield of Medical Tests, JAMA 247 (1982) 2543–2546.

[11] A. P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern recognition 30 (1997) 1145–1159.

[12] T. P. Debray, J. A. Damen, K. I. Snell, J. Ensor, L. Hooft, J. B. Reitsma, R. D. Riley, K. G. Moons, A guide to systematic review and meta-analysis of prediction model performance, bmj 356 (2017).

[13] N. V. Chawla, N. Japkowicz, A. Kotcz, Editorial: Special issue on learning from imbalanced data sets, SIGKDD Explor. Newsl. 6 (2004) 1–6.

[14] S. B. Kotsiantis, D. Kanellopoulos, P. E. Pintelas, Data preprocessing for supervised leaning, International journal of computer science 1 (2006) 111–117.

[15] J. Howard, S. Gugger, Fastai: A layered API for deep learning, Information 11 (2020) 108.

[16] P. D. Allison, Survival analysis using SAS: a practical guide, Sas Institute, 2010.

[17] K. Bogaerts, A. Komarek, E. Lesaffre, Survival analysis with interval-censored data: A practical approach with examples in R, SAS, and BUGS, CRC Press, 2017.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[19] L. Biewald, Experiment tracking with weights and biases, 2020. URL: https://www.wandb.com/, software available from wandb.com.