Informatics in Medicine Unlocked 37 (2023) 101161

Contents lists available at ScienceDirect

Informatics in Medicine Unlocked

journal homepage: www.elsevier.com/locate/imu

High-performance prediction models for prostate cancer radiomics

Lars Johannes Isaksson [a,b,*], Marco Repetto [c,d,e], Paul Eugene Summers [a], Matteo Pepa [a], Mattia Zaffaroni [a,**], Maria Giulia Vincini [a], Giulia Corrao [a], Giovanni Carlo Mazzola [a,b], Marco Rotondi [a,b], Federica Bellerba [f], Sara Raimondi [f], Zaharudin Haron [g], Sarah Alessi [h], Paula Pricolo [h], Francesco Alessandro Mistretta [i], Stefano Luzzago [i], Federica Cattani [j], Gennaro Musi [b,i], Ottavio De Cobelli [b,i], Marta Cremonesi [k], Roberto Orecchia [l], Davide La Torre [b,e], Giulia Marvaso [a,b], Giuseppe Petralia [b,m], Barbara Alicja Jereczek-Fossa [a,b]

[a] *Division of Radiation Oncology, IEO European Institute of Oncology IRCCS, Milan, Italy*
[b] *Department of Oncology and Hemato-Oncology, University of Milan, Milan, Italy*
[c] *Digital Industries, Siemens Italy, Italy*
[d] *Department of Economics, Management and Statistics, University of Milan-Bicocca, Italy*
[e] *SKEMA Business School, Université Côte d'Azur, Sophia Antipolis Campus, France*
[f] *Department of Experimental Oncology, IEO European Institute of Oncology IRCCS, Milan, Italy*
[g] *Radiology Department, National Cancer Institute, Putrajaya, Malaysia*
[h] *Division of Radiology, IEO European Institute of Oncology IRCCS, Milan, Italy*
[i] *Division of Urology, IEO European Institute of Oncology IRCCS, Milan, Italy*
[j] *Medical Physics Unit, IEO European Institute of Oncology IRCCS, Milan, Italy*
[k] *Radiation Research Unit, IEO European Institute of Oncology IRCCS, Milan, Italy*
[l] *Scientific Directorate, IEO European Institute of Oncology IRCCS, Milan, Italy*
[m] *Precision Imaging and Research Unit, Department of Medical Imaging and Radiation Sciences, IEO European Institute of Oncology IRCCS, Milan, Italy*

ARTICLE INFO

Keywords:
Radiomics
Prostate cancer
Deep learning
Gradient boost

ABSTRACT

When researchers are faced with building machine learning (ML) radiomic models, the first choice they have to make is what model to use. Naturally, the goal is to use the model with the best performance. But what is the best model? It is well known in ML that modern techniques such as gradient boosting and deep learning have better capacity than traditional models to solve complex problems in high dimensions. Despite this, most radiomics researchers still do not focus on these models in their research. As access to high-quality and large data sets increase, these high-capacity ML models may become even more relevant. In this article, we use a large dataset of 949 prostate cancer patients to compare the performance of a few of the most promising ML models for tabular data: gradient-boosted decision trees (GBDTs), multilayer perceptions, convolutional neural networks, and transformers. To this end, we predict nine different prostate cancer pathology outcomes of clinical interest. Our goal is to give a rough overview of how these models compare against one another in a typical radiomics setting. We also investigate if multitask learning improves the performance of these models when multiple targets are available. Our results suggest that GBDTs perform well across all targets, and that multitask learning does not provide a consistent improvement.

1. Introduction

Quantitative image analysis as a tool to improve decision-making and management in healthcare has been growing steadily. A specific analytic technique called Radiomics, where predefined mathematical features are calculated from regions of interest (ROIs) in medical images, has recently gained particular attention in radiation oncology [1–3]. The idea is to use the radiomic features to build predictive models for purposes such as treatment planning or outcome prognosis. The overwhelming majority of radiomic studies to date have focused on proof-of-concept experiments demonstrating the usefulness of radiomics in specific circumstances [4–7]. By and large, the discrepancies between different predictive models have not been studied, with researchers often relying on relatively simple but intuitive models such

* Corresponding author at: Division of Radiation Oncology, IEO European Institute of Oncology IRCCS, Milan, Italy.
** Corresponding author.
E-mail addresses: LarsJohannes.Isaksson@ieo.it (L.J. Isaksson), mattia.zaffaroni@ieo.it (M. Zaffaroni).

https://doi.org/10.1016/j.imu.2023.101161
Received 15 September 2022; Received in revised form 10 December 2022; Accepted 3 January 2023
Available online 13 January 2023

as logistic regression, random forests, basic support vector machines, or cox regression. These generally work well for small datasets (e.g. under 100 samples) but their performance likely deteriorates in high dimensions and non-linear scenarios [8–12].

Prediction models for tabular data (like the ones constructed from radiomic features) are a staple in machine learning (ML) research since considerable amounts of real-world data are collected in the form of spreadsheets. In medical research, traditional algorithms such as logistic regression are often preferred when building prediction models, in part because they naturally integrate with statistical inference such as the likelihood ratio test. On the other hand, developments in data science and ML have largely rendered traditional models obsolete in favor of better-performing approaches such as deep learning (DL) and gradient-boosted models. One of the most popular models is gradient-boosted decision trees (GBDTs), which use the gradient-boosting technique to iteratively build an ensemble of decision trees (analogous to a gradient descent algorithm). A great deal of research has also been devoted to developing DL models that can compete with GBDTs; some noteworthy efforts being TabNet [13], NODE [14], DNF-Net [15], SNN [16], GrowNet [17], DCN V2 [18], and AutoInt [19]. However, when evaluated on a wide range of different tasks, many of these models have shown little to no improvement over simpler baselines like 1D convolutional neural networks (CNNs) or multilayer perceptrons (MLPs) [11,12,20–22]. Therefore, standard MLPs and CNNs may be preferred when evaluating the utility of DL on a new task such as radiomics, particularly when comparing many models is impractical. The question remains as to whether these DL models have the potential to compete with GBDTs in radiomics, where the datasets are characterized by few samples, many variables, and a mix of categorical, ordinal, and continuous data.

In clinical prostate oncology, it is recommended practice to characterize the tumor via both MR-image-related parameters (such as prostate volume, number of dominant lesions, PI-RADS score, EPE value, and ADC value) and biopsy-related parameters (such as initial PSA, ISUP grade group, tumor stage, lymph node status, and risk class) in order to select an appropriate treatment and follow-up schedule. Surgery by radical prostatectomy (removal of the prostate) and radiotherapy are the two most common treatments for localized prostate cancer and have comparable oncological outcomes. Pre-treatment access to the information from the pathological assessment after surgery would enable doctors to refine their evaluation of the patient so that better decisions and prognoses can be made. Thus, non-invasive prediction of pathological determinants of prostate cancer could reduce risks and improve outcomes, but so far this has not been the primary focus in prostate cancer radiomics. In this work, we focus on predicting nine different endpoints from prostate cancer pathology, all of which hold critical clinical value.

In this article, we address the issue of building the best-performing radiomic models within the context of prostate radiomics. To do this, we compare the performance of different prediction models and learning techniques on a large dataset of 949 prostate cancer patients with nine different pathology endpoints of clinical value in prostate cancer care. Our chosen methods are a mix of common high-performance ML models: gradient-boosted decision trees (GBDTs), multilayer perceptron (MLP), one-dimensional convolutional neural network (1DCNN), and a transformer model with a feature tokenizer (FT-Transformer) [20] specialized for tabular data. To combat the problem of overfitting and overoptimistic performance estimates, we employ rigorous training, validation, and test procedures in all experiments. We also evaluate the benefit of multitask learning, which is a training technique in which the model learns to predict every available endpoint simultaneously, leading to a potential improvement in both performance and speed. To the best of our knowledge, multitask/multi-target learning has not been adequately explored in previous radiomics studies.

2. Methods

2.1. Dataset

Patient data were retrospectively collected from 949 prostate cancer patients who had undergone multiparametric prostate MRI and prostatectomy in the European Institute of Oncology (IEO) from 2015 to 2018. For each patient, we used the T2-weighted MRI sequences and all potentially relevant clinical variables.

The MRI images were acquired using a 1.5 T MR scanner with slice thickness 3.0–3.6 mm, slice gap 0.3 mm, pixel spacing 0.59 × 0.59 mm, echo time 114–118 ms, and repetition time 3780 ms (median). Eight out of the 949 images were acquired with non-habitual protocol sequences or a second scanner from another manufacturer.

The clinical variables included age, initial PSA, and comorbidity as well as MRI-related data such as prostate volume, number of dominant lesions, PI-RADSv2 score, extraprostatic extension score, and ADC (apparent diffusion coefficient). The following tumor-related pre-treatment variables were obtained: initial ISUP grade group, clinical tumor stage (T), clinical lymph node status (N), and NCCN2019 risk class. We also collected six post-treatment pathology variables to use as prediction targets: post-operative ISUP grade group, pathological T, pathological N, surgical margin, biochemical progression, and clinical progression. An overview of the clinical characteristics within the cohort is presented in Table 3 in Appendix A.

2.2. Experiments

To compare the different models, we trained them to predict six pathological endpoints of clinical interest:

1. Post-operation ISUP grade group
2. Pathological tumor stage (T)
3. Pathological lymph node status (N)
4. Surgical margin
5. Biochemical progression
6. Clinical progression

as well as three "delta"-variables encoding change between the clinical and pathological assessments of the ISUP, T, and N endpoints:

7. ΔISUP
8. ΔT
9. ΔN

Details of the models and their training routines are given below.

The models were compared in terms of their overall test performance in a nested five-fold cross-validation routine (see Section 2.5 for details). The performance was measured in terms of three different classification metrics: Matthews correlation coefficient (MCC), AUC, and accuracy.

2.3. Models

We compared four different tabular data models: GBDTs implemented with the CatBoost [23] package, an MLP, an FT-Transformer (FTT), and a 1D CNN. We also trained multitask variants of each model. An additional MLP-style model tailored specifically towards multi-objective training was also tested. The models are described briefly below while their training routines are presented in Section 2.5.

2.3.1. CatBoost

CatBoost [23] is a free open-source library for GBDT models (similar to XGBoost [24] and LightGBM [25]). GBDT models have been incredibly successful in tabular data analysis applications, in large part due

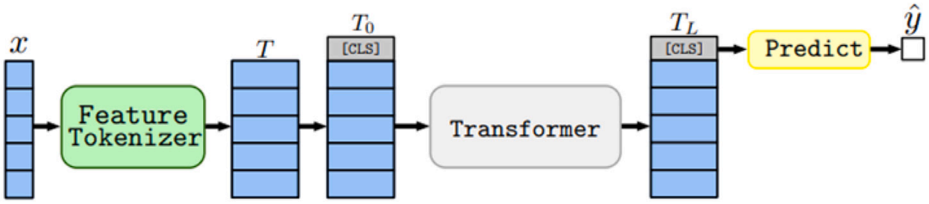

Fig. 1. The architecture of the FT-Transformer. An input x is passed through a feature tokenizer and converted into embeddings T. Multiple transformer layers (T_0, ..., T_L) are connected in sequence after which the CLS token is used for prediction.
Image source: [20].

to their high discriminative power yet simple structure and high speed. For instance, the top positions in the Kaggle competitive ML platform are most often held by gradient-boosted models [26]. Compared to DL, building and training GBDT models is considerably easier since it only requires specifying a set number of parameters (e.g. the number of decision trees in the ensemble, their maximum depth, the learning rate, and a handful of optional regularization parameters), after which the tree-growing algorithm builds and trains the model. Consequently, it also requires much less computation time and resources.

2.3.2. MLP

The MLP model is one of the earliest and most straightforward DL architectures. Despite their relative simplicity, they have been shown to perform well on tabular data if trained properly [12,22,27]. They are often faster and use fewer hyperparameters than their alternatives, which makes them attractive candidates for search-based optimization pipelines. Apart from regularization parameters, MLPs can be parameterized by just the number of layers and the number of neurons within each layer. Our implementation used two layers (we found that more layers did not improve the performance but increased the training time) and a swish activation function followed by batch norm and dropout.

2.3.3. FT-Transformer

The FT-Transformer [20] was recently proposed as a transformer architecture specifically tailored for tabular data. The use of transformers in this scenario is not commonplace, but the research direction is attractive given the success of transformers in natural language processing [28] and more recently computer vision [29]. The model consists of a feature tokenizer module and several consecutive transformer layers (Fig. 1), where the former converts the input variables (both categorical and numeric) into embeddings, and the latter performs the recurrent self-attention that is characteristic of the transformer. The final layer performs prediction with the CLS (“classification”) token, which is designed to contain information about the sentiment of the whole input sequence (as opposed to information about individual elements in the sequence).

2.3.4. 1D-CNN

Convolutional archetypes are commonly used for image analysis because they can handle local correlations efficiently and are not impaired by positional and morphological variances. Tabular data, however, do not display these types of characteristics, which makes the value of CNNs less apparent in these scenarios. One way to circumvent this is to map the data into a higher-dimensional superspace in which the locality of convolutions is not an impediment. The implementation we used is taken from the second-place submission (the 1D-CNN itself had the best single-model performance) in a 2020 Kaggle competition for tabular data [30] and uses a learnable dense/linear layer to perform this mapping. After the initial dense layer, several 1D-convolution and pooling layers are connected in sequence along with a skip connection and a flatten operation (see Fig. 2).

2.3.5. Multitask-tailored MLP

In addition to multitask variants of the four models above, an additional MLP model was created and optimized specifically for the multitask scenario. This is motivated by the fact that some design and architectural choices have no analogous counterpart in the single-task setting. In particular, the multitask model was constructed by two distinct partitions: a base that is shared among all different tasks, and N classification heads that have distinct parameters for each output task. The base handles the input and latent representations of the model while the heads are fine-tuned for their respective tasks. The architecture (displayed in Fig. 3) builds upon the MLP and uses skip connections and varying activation functions.

2.4. Data processing

2.4.1. Image processing

Each image was corrected with the N4 bias-field correction algorithm (implemented in sITK 2.1.1 using default parameters), and the image intensities were subsequently normalized with an outlier-aware range normalization that linearly maps the 0th and 99th percentile values of every image to a predefined range (in this case between 0 and 424). Unique values in the 100th percentile were appended after 424 using a 1:1 linear map (the first value was mapped to 424+1, the second to 424+2, etc.)

2.4.2. Image segmentation

The prostate in each image was segmented with a bespoke deep learning segmentation algorithm with a 3D U-net-like architecture [31,32]. The Dice coefficients for the segmentations were subsequently estimated by a deep learning quality assurance model [33]. In order to ascertain an acceptable standard for the automatically generated contours, a subset of the segmentations was selected and sent for correction by an expert radiologist. Segmentations were included in this subset if at least one of the following two criteria was met: (1) the segmentation had an estimated Dice coefficient of 0.8 or lower, and (2) the volume of the segmentation was in the upper or lower two percentiles. In total, 98 segmentations were selected.

2.4.3. Radiomic feature extraction and pruning

Radiomic features were extracted from the whole prostate using PyRadiomics [34] 3.0 in Python 3.7. All available features were extracted from all available filter classes (Laplacian of Gaussian, wavelet, square, square root, logarithm, exponential, gradient, local binary pattern 2D, and local binary pattern 3D). For the Laplacian of Gaussian filter, we calculate features for three different values of sigma corresponding to 1, 2, and 5 times the in-plane spacing (0.59375 mm). No resampling was performed since all the images had nearly identical resolution and spacing. In total, 1967 features were extracted. Features were removed if their variance was lower than 10^{-6} or if they had an absolute Spearman correlation above 0.98 with any other feature. In the latter case, the feature with the lowest cumulative correlation with other features was kept. After these steps, 737 features remained.

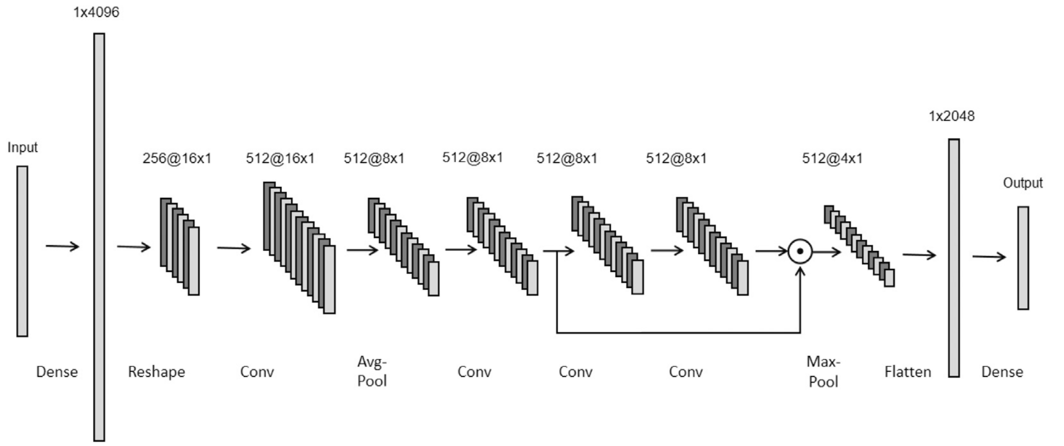

Fig. 2. The architecture of the 1DCNN model. An initial dense layer maps the input feature vector into a larger vector more suited for convolutional operations. Standard convolutional and pooling operations are connected, and a normal dense layer performs the final prediction. The numbers represent the dimension at each step ($n_{channels}@n_{depth}$).

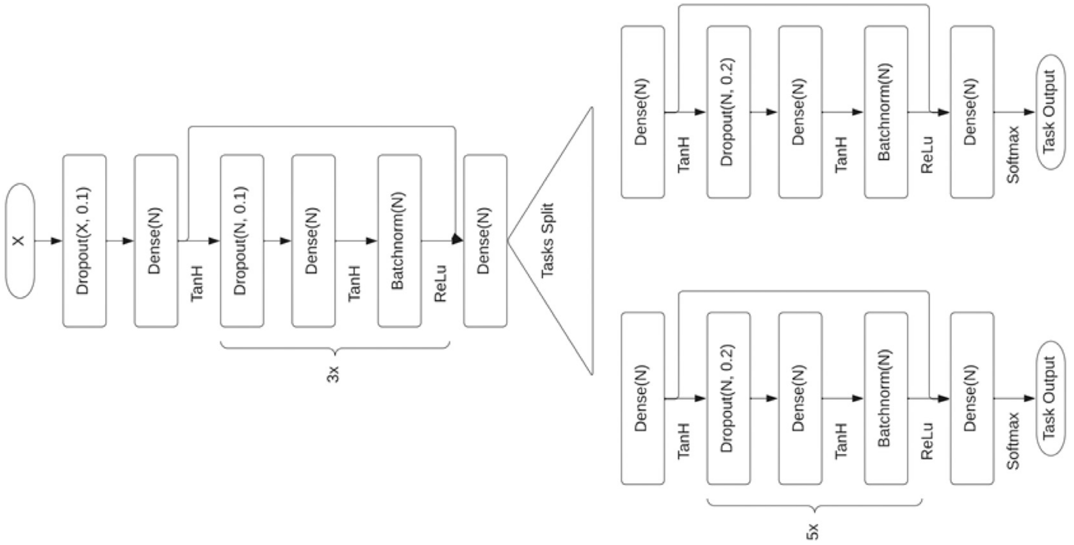

Fig. 3. The architecture of the multitask-tailored MLP model. A base (left) structure is shared between all different tasks, and multiple prediction heads (right) are trained to optimize each individual task separately.

2.4.4. Data preprocessing

The clinical variables were ordinally encoded as follows:

- Clinical and pathological T: stages were encoded by their integer value such that both T3a and T3b were encoded as 3, and T2a and T2b as 2, etc.
- Clinical and pathological N: stages were encoded by their integer value (N1→1, N0→0).
- Risk class: "low" and "very low" were encoded as 0, "intermediate favorable" and "favorable" as 1, "intermediate unfavorable" and "unfavorable" as 2, and "high" or "very high" as 3.
- Clinical progression: absence of progression was encoded as 0, and any type of progression (pelvic, extrapelvic, or both) was encoded as 1.

Prior to training the models, we binarized the non-binary target variables (post-operation ISUP grade, pathological T, and the delta variables), which allowed us to readily compare the performance of all different targets with the same metrics. Figure 9 shows the class distribution of the targets after the binarization. We also observed that this achieved better classification performance than binarizing the predictions after training the models with regression. This can be done without much loss of clinical utility since many of the decisions within the clinical workflow surrounding these parameters are primarily made based on threshold values.

For the DL models, all input variables were normalized with a quantile transformer with the number of quantiles set to the number of training samples, and missing values were imputed with a k-nearest neighbors imputer with $k = 8$ and distance-based weights (see Appendix B.1 for a deeper analysis on the imputation and its parameters). Neither imputation nor quantile transformation is needed in CatBoost, since CatBoost inherently handles missing values[1] and the tree-growing algorithm uses cutoff values that are scale-independent.

2.5. Model training & feature selection

2.5.1. CatBoost

The CatBoost model was trained with a nested 5-by-5 stratified cross-validation (CV) and a standard binary cross-entropy (log-loss) loss function. The inner 5-fold was used to search for parameters and the outer 5-fold was used to estimate the performance. This eliminates the selection bias from selecting the lowest error model. To search

[1] In CatBoost, missing values are treated as smaller than the smallest real-valued entry, effectively making them a separate category.

for parameters, we used the Tree-structured Parzen Estimator (TPE) in the Optuna (v2.10) python package [35] and ran it for 600 trials with default parameters. Among the trials, we selected the model with the highest Matthews correlation coefficient (MCC) on the validation sets and proceeded to retrain it on the full training+validation set. The performance of the retrained model was then evaluated on the previously held-out test sets.

The hyperparameter search space is shown in Table 4 in Appendix B.2. Note that we directly optimize the number of radiomic features to include ("n_features") based on their estimated predictive power over the target variable (see Appendix C for details).

2.5.2. Deep learning models

The DL models were trained with a similar cross-validation procedure, but instead of a full 5-fold validation in the inner loop, we used a single train–test split (80%–20%) due to computational constraints. Moreover, the DL hyperparameter search was carried out with 64 trials instead of 600. All random seeds were fixed so every model was trained and evaluated on the same data (including the CatBoost model).

The hyperparameter search spaces and complete training details for the different DL models are shown in Appendix B.2. Like for the CatBoost model, we directly optimize the number of radiomic features to use within the parameter search (see Appendix C for details).

We trained the DL models to directly optimize a differentiable version of the MCC, which can be achieved by defining continuous versions of the true positives/negatives (TP and TN), and false positives/negatives (FP and FN). In other words, if we let y be the real target label and $h(x)$ be the network's prediction:

$$TP = \sum_i \mathbf{y}_i h(\mathbf{x}_i) \tag{1}$$

$$TN = \sum_i (1 - \mathbf{y}_i)(1 - h(\mathbf{x}_i)) \tag{2}$$

$$FP = \sum_i (1 - \mathbf{y}_i) h(\mathbf{x}_i) \tag{3}$$

$$FN = \sum_i \mathbf{y}_i (1 - h(\mathbf{x}_i)). \tag{4}$$

In the above formulae, it is assumed that the last operation in h is a logistic function such that the outputs are in the (0, 1) range. This loss function has been argued to exhibit many attractive properties compared to its alternatives, particularly for imbalanced and medical datasets [36–38].

2.5.3. Multitask learning

For each of the four models mentioned above, we built alternative versions that were trained with a multitask/multi-objective loss function. This can be done with very minor modifications to the architectures (e.g. simply adjusting the number of outputs of the final layer) and has the potential to improve both speed and accuracy [39–41]. The multitask versions of the models were trained in the same way as the regular ones, but with three important modifications:

1. Since the loss of the models needs to be a single scalar value, we calculated the total loss by averaging the individual classification errors over all different targets (the binarization of the target values allows us to make this aggregation without needing to tune the weights between different types of targets).
2. A consequence of simultaneously predicting all targets is that missing target data needs to be handled differently. In the single-task case, we did not implement any target-specific imputation considerations since we could simply select and train on all the patients with target data available. In the multi-task case, the patients have varying target values missing, which means that some loss values are not defined. Hence, we divided the full data set into training and test data as the first step, and whenever a batch of samples with missing target data was encountered, we calculated the total loss for each patient as the mean loss over all its non-missing values.[2] A consequence of this is that regular (e.g. non-stratified) cross-validation has to be used.
3. An issue similar to point 2 is faced when we calculate the predictive power of the features prior to feature selection. The predictive power was thus also calculated by averaging the loss over all available targets.

Table 1
Class distributions of the pathological target variables in terms of the number of positive and negative cases and percentages. In total, 949 patients were considered.

Class	n_{cases} (+/−)	% (+/−)	Missing
Post-op ISUP Group	877/68	93/7	4
Pathological T	582/367	61/39	
Pathological N	495/76	87/13	378
Surgical margin	716/232	76/24	1
Biochemical progression	637/140	82/18	172
Clinical progression	726/50	94/6	173
ΔISUP	603/343	64/36	3
ΔT	2030/715	24/76	4
ΔN	493/75	87/13	381

3. Results

3.1. Dataset

Table 1 shows the class distribution of the target variables (the number of positive, negative, and missing values) after discretization. A detailed overview of the clinical properties of the patient cohort is given in Appendix A, including a graphical representation of the distributions.

3.2. Radiomic model performance

An overview of the performance of the models is displayed in Fig. 4 and a summary of their relative scores in terms of their rank is displayed in Fig. 5. ROC curves for the AUC values are shown in Figure 14 in Appendix D. For all endpoints, the CatBoost model achieved the highest MCC whereas the FTT and 1DCNN models appear to be nearly equivalent (2.33 and 2.78 mean rank, respectively). The MLP only achieved a similar MCC to the other DL models in one of the nine cases (biochemical progression), resulting in the worst overall mean rank of 3.89. In terms of AUC, the results are similar, with the exception that the 1DCNN achieved a better mean rank than the FTT (2.33 vs. 3.0). The results for the accuracy are different: all four models performed at a comparable level, with MLP achieving the best mean rank. However, the mean rank difference between the MLP and the worst-performing model, which is a tie between the FTT and the 1DCNN (mean rank of 2.67), is relatively small (0.56 mean ranks).

3.3. Multitask performance

Fig. 6 shows the performance of the multitask models and Fig. 7 shows their respective ranks. The 1DCNN appears to be the overall best model in terms of both MCC and AUC (though tied for first place with the CatBoost model in the AUC case). CatBoost, MLP, and FTT all had comparable performance in terms of MCC, but the MT-MLP clearly performed worse. The MT-MLP has no AUC score since it was trained with regression, but between the remaining two models, MLP and FTT, the FTT performed better. The accuracy ranks are very different: the 1DCNN was the worst-performing model and MLP was the best. The MT-MLP still performs poorly with its second-to-last place, but CatBoost and FTT perform very similarly.

[2] We can do this without worrying about undefined gradients since no sample has all its target values missing. If such a patient existed, it would not have been included in the dataset in the first place since we cannot hope to receive any signal from it.

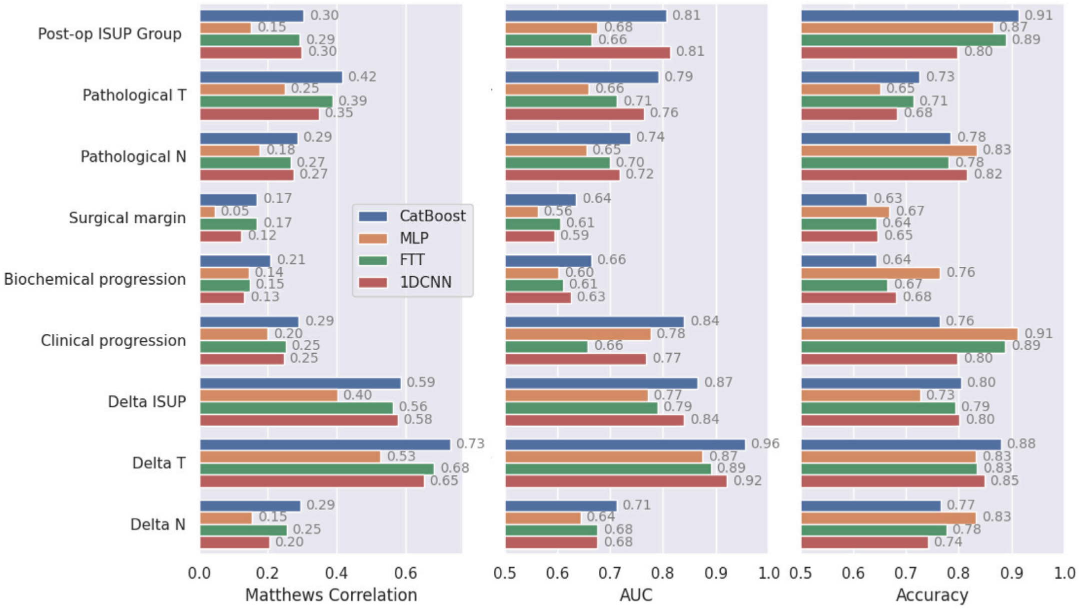

Fig. 4. Performance of the different models in terms of Matthews correlation, AUC, and accuracy for the nine different prediction targets. Higher scores are better. MLP: multilayer perceptron, FTT: feature-tokenizer transformer, 1DCNN: 1D convolutional neural network.

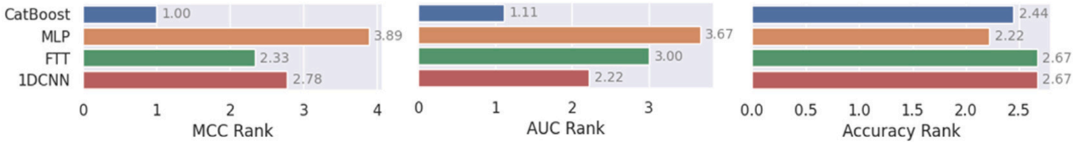

Fig. 5. Total mean rank of the different models, aggregated as the mean over all nine different prediction targets. A lower rank is better.

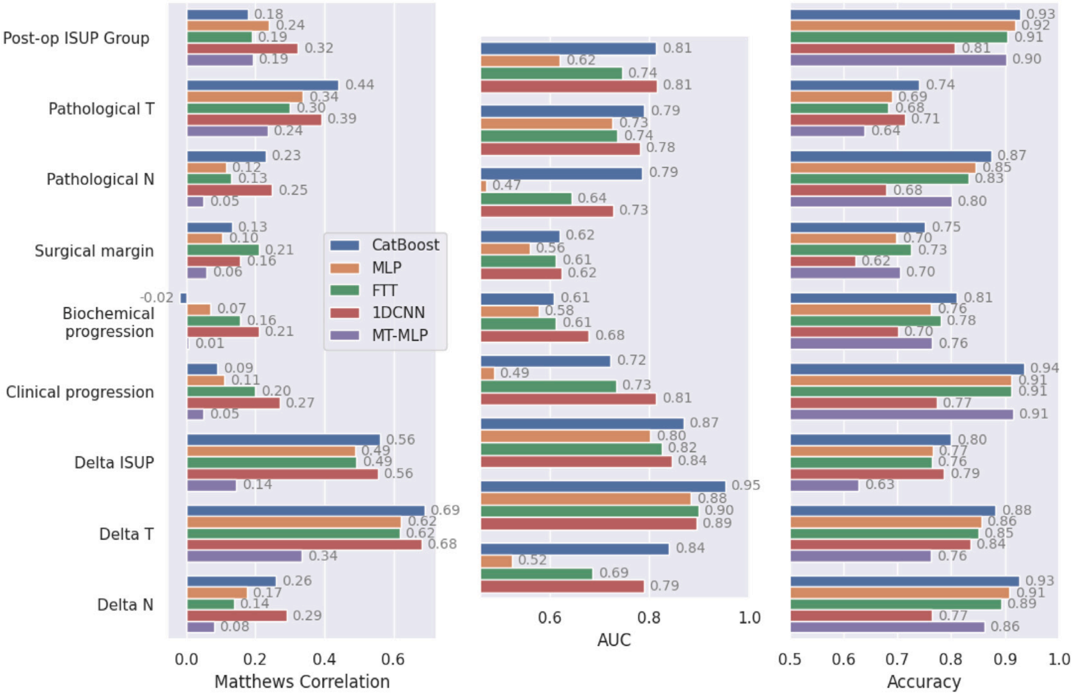

Fig. 6. Performance of the multitask versions of the different models in terms of Matthews correlation, AUC, and accuracy for the nine different prediction targets. Higher scores are better. The MT-MLP model has no AUC since it does not output produce probability estimates.

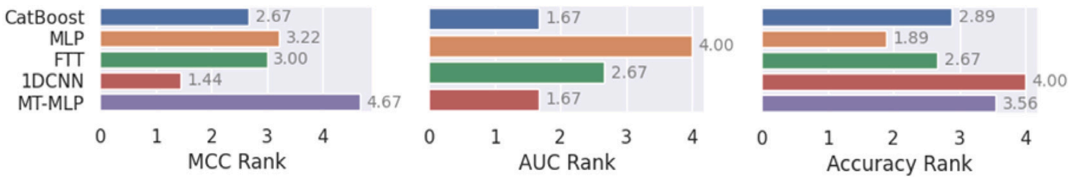

Fig. 7. Total mean rank of the multitask versions of the different models, aggregated as the mean over all nine different prediction targets. A lower rank is better.

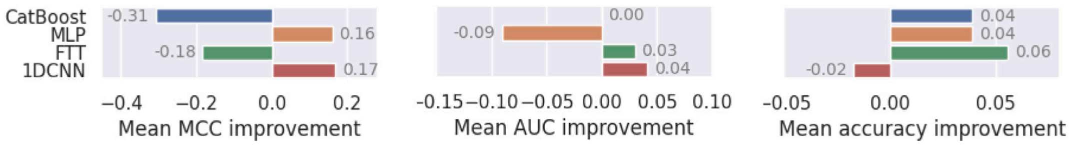

Fig. 8. Mean improvement gained from multitask training of the different models, aggregated over all nine different prediction targets.

Table 2
The single best model, including both single-task (ST) and multitask (MT), in every prediction target and performance metric.

Target	MCC	AUC	Accuracy
Post-op ISUP Group	1DCNN (MT)	CatBoost (ST + MT),1DCNN (ST + MT)	CatBoost (MT)
Pathological T	CatBoost (MT)	CatBoost (ST + MT)	CatBoost (MT)
Pathological N	CatBoost	CatBoost (MT)	CatBoost (MT)
Surgical margin	CatBoost	CatBoost	CatBoost (MT)
Biochemical progression	CatBoost,1DCNN (MT)	1DCNN (MT)	CatBoost (MT)
Clinical progression	CatBoost	CatBoost	CatBoost (MT)
ΔISUP	CatBoost	CatBoost (ST + MT)	CatBoost (ST + MT),1DCNN (MT)
ΔT	CatBoost	CatBoost	CatBoost (ST + MT)
ΔN	CatBoost,1DCNN (MT)	CatBoost (MT)	CatBoost (MT)

3.4. *Multitask improvement*

In Fig. 8, we show the mean percentage performance gained from training multitask models compared to several single-task models. The overall difference is very large in terms of MCC (16%–31%) but fairly small for AUC (0%–9%) and accuracy (2%–6%). The added benefit of multitask training differs between models and metrics which makes it hard to declare a single best approach. Note that the performance difference can be positive for some endpoints even though the mean improvement is negative, and vice versa. The most drastic change was, by far, the MCC for the CatBoost model, which performed 31% worse in the multitask case on average. The second largest change was the MCC for the FTT model (−18%). In contrast, the MCC of both the MLP and 1DCNN benefited greatly (16% and 17%) from the multitask training. In summary, the mean performance was better in one case (and a tie in one case) for the CatBoost model, two cases for the MLP, two cases for the FTT, and two cases for the 1DCNN.

By comparing the best multitask models with the best single-task models, we can discern the overall best-performing model across all endpoints (Table 2). Only two architectures are represented: CatBoost and, to a lesser extent, the 1DCNN. In the accuracy metric, multitask models are clearly overrepresented. In MCC and AUC, the single-task CatBoost model is the most common, and multitask 1DCNN the second most common.

4. Discussion

CatBoost performed consistently well and was the model that most often achieved the best score. Of the deep learning models, the multitask 1DCNN was generally the best, followed by the FT-Transformer. There appears to be a tendency for the multitask models to perform better in terms of accuracy than in terms of the other metrics, which may indicate a slight tradeoff between MCC/AUC and accuracy. This is also supported by observing that the MLP generally performed much better in terms of accuracy than the other two scores. It is however likely that the MLP is not inherently better at achieving higher accuracy, and that CatBoost would still be the best model if we decided to optimize for accuracy instead.

The stability of models and variance of their performance: A major concern in studies comparing the performance of different models is the variance and reproducibility of the results. In the model validation pipeline, there is a delicate balance between exploration and validation in the sense that finding higher-performing models via deeper exploration in the model space competes with performing more thorough validation procedures (such as repeated cross-validation). If the model space is not sufficiently explored, there will be large variations in the selected models (since there will be a sparser sampling of the error landscape). Conversely, if the models are not thoroughly validated (e.g. with repeated experiments with different random seeds), there may be large variations due to instabilities in the training data or model weights. Multiple sources of variations make it hard to analyze the performance with regular statistical tools without excessive amounts of compute. In the medical field, these instabilities are largely overlooked, even though the small datasets may exacerbate the problem. Due to limited computational resources, this study focused on finding high-performance models instead of repeated experiments, which is currently the standard practice. To ensure the reliability of this approach, we performed informal repeated tests (only on the first target variable due to the required computational investment) and concluded that this variability would not invalidate the results. It is worth noting that another way of improving the reliability of predictions is to create model ensembles, but this also requires an increased investment of computational resources.

The optimization objective and different performance measures: We chose to optimize the MCC because it is known to be a more reliable and informative measure of performance than other metrics like accuracy, F1-score, and AUC [36–38,42]. For example, the MCC only achieves a high value when the classifier produces good results in all four quadrants of the binary confusion matrix. In medical research, the AUC has been the standard reporting metric for prediction models, which in many ways can be seen as a cause for concern (see e.g. [43–45]). For instance, the definition of AUC allows classifiers to increase their AUC without modifying a single prediction since it considers all decision thresholds and not just the actual operating threshold. A similar mechanism also allows them to simultaneously increase their AUC and decrease their accuracy. This is not to say that AUC does not have its uses, but we believe these are legitimate reasons to not optimize for AUC directly. Accuracy is another natural candidate for optimization, but this metric can be misleading for unbalanced data. In preliminary experiments, we observed that optimizing either AUC or accuracy often leads to majority classifiers that classify all patients into the majority class, which are essentially useless in practice (they do not utilize any information available in the variables). When deploying prediction models in real clinical environments, it will be crucial to discuss and clarify if true/false positives and negatives should be weighted equally, and then optimize the appropriate metric.

Feature selection: Feature selection is one of the most important aspects of model development because it dictates what information the model will have at its disposal. Despite this, it is not commonly discussed or researched within studies of medical prediction models. Since the procedure's outcome is heavily data and problem-dependent, it can be wise to explore the options when presented with a new problem or data set (though this requires additional effort). In most studies, features are selected prior to training (e.g. by a clustering procedure or statistical testing) or internally to the model (e.g. with LASSO), which is problematic in several ways. First, it does not properly account for feature–parameter interactions. Second, it may introduce leakage if not incorporated correctly (once for each training set) into the validation pipeline. It is also susceptible to additional variance and lower performance. For these reasons, we instead incorporated the selection procedure into the parameter optimization (see Appendix C), which should generally be preferred if its implementation is possible.

Multitask features & loss weighting: When selecting features for the multitask models, we chose to select the feature with the highest mean predictive power over all different endpoints. An exciting alternative to this is to favor features with high predictive power for tasks the model struggles with, which would effectively act as an indirect loss-weighing for different targets. In loss-weighting, each task's loss function is weighted differently to optimize an aggregated loss function that may be more ideal. This raises the question of how exactly to weigh the different tasks, which is a complex optimization problem in and of itself. Furthermore, the desired balance is heavily influenced by external factors such as preference and the individual samples being evaluated. It is also possible that other loss functions and/or different weight-sharing strategies would improve the multitask performance. We chose not to explore these considerations due to the foreseen complexity, but it is of interest for future research.

Limitations: The radiomic features used in this study were extracted from the whole prostate in the T2-w MRI images. As such, the performance may not be as good as models incorporating other imaging modalities, such as DWI (diffusion-weighted images) and ADC, and/or features extracted from the dominant lesion. However, some information from the DWI and ADC images is encoded with the PI-RADS and ADC values that were used as input features, meaning that the information is not entirely lost. Another factor that may influence the performance is the automatic segmentation procedure, but the quality assurance step was incorporated to address this. Moreover, a degree of variation in the acquisition protocol was present that may contribute to noise in the radiomics features across subjects. However, the small differences in echo and repetition time would not be expected to affect the ordering of signal intensities across tissues, and thus should be largely compensated by the image normalization process. The differences in b-values may have had a similar effect on the calculated ADC values. That said, our images were acquired in the course of routine clinical practice, and should therefore reflect real-world context. Finally, even though the dataset of 949 patients is the largest one to date (the previous one being 489 patients, according to a recent review covering 57 different prostate cancer radiomics studies [6]), it is still not big enough to make conclusive statements about the validity of these models in clinical practice. It is conceivable that the small dataset contributed to the somewhat subpar performance of the DL models given that DL models are known to be data-hungry. A potential solution for this that was not studied in this article is data augmentation, which is standard practice for most non-tabular DL models. But even with these limitations in mind, the comparison between the models should be fair and relevant since all models were trained under the same conditions.

5. Conclusion

In this study, we have compared the prostate radiomics performance of one popular GBDT model and three popular DL models for tabular data: an MLP, the FT-Transformer, and a one-dimensional CNN. We also investigated whether these models benefit from multitask learning when multiple pathological target variables are available. Our experiments indicate that the GBDT model implemented with CatBoost was generally the most consistently high-performing model (in terms of both MCC and AUC). The multitask version of the 1DCNN also performed well overall. Multitask learning brought a considerable benefit for the MLP and 1DCNN models but was detrimental for the CatBoost model and the FT-Transformer, which makes neither training procedure a clear winner.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

IEO, the European Institute of Oncology, is partially supported by the Italian Ministry of Health (with "Ricerca Corrente" and "5x1000" funds) and by Institutional grants from Accuray Inc. Lars J. Isaksson is a Ph.D. student within the European School of Molecular Medicine (SEMM), Milan, Italy. Mattia Zaffaroni and Maria Giulia Vincini supported by a research fellowship from the Associazione Italiana per la Ricerca sul Cancro (AIRC) entitled "Radioablation ±hormonotherapy for prostate cancer oligo-recurrences (RADIOSA trial): potential of imaging and biology" registered at ClinicalTrials.gov NCT03940235. Prof. Jereczek-Fossa reports grants from Accuray, AIRC and from Fondazione IEO-CCM (Istituto Europeo di Oncologia-Centro Cardiologico Monzino) personal fees for lecturers, presentations or participation on data safety monitoring or advisory boards from ESTRO, ESO, ASCO, ROCHE, BAYER, JANSSEN, CARL ZEISS, ACCURAY, ASTELLAS, ELEKTA, IBA and ACCMED all outside the submitted work. The Sponsors did not play any role in the study design, collection, analysis and interpretation of data, nor in the writing of the manuscript, nor in the decision to submit the manuscript for publication. All the other authors have nothing to disclose.

Ethical statement

The study was performed with the approval of the Ethics Committee of IRCCS Istituto Europeo di Oncologia and Centro Cardiologico Monzino (via Ripamonti 435, 20,141 Milano, Italy), CE notification n. UID 2438. Informed consent was obtained from all subjects for use of their data for research and educational purposes. All methods were performed in accordance with the relevant guidelines and regulations under the Declaration of Helsinki (as revised in 2013).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.imu.2023.101161.

References

[1] Kumar V, Gu Y, Basu S, Berglund A, Eschrich SA, Schabath MB, Forster K, Aerts HJ, Dekker A, Fenstermacher D, et al. Radiomics: the process and the challenges. Magn Reson Imaging 2012;30(9):1234–48.

[2] Aerts HJ, Velazquez ER, Leijenaar RT, Parmar C, Grossmann P, Carvalho S, Bussink J, Monshouwer R, Haibe-Kains B, Rietveld D, et al. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. Nature Commun 2014;5(1):1–9.

[3] Mayerhoefer ME, Materka A, Langs G, Häggström I, Szczypiński P, Gibbs P, Cook G. Introduction to radiomics. J Nucl Med 2020;61(4):488–95.

[4] Song J, Yin Y, Wang H, Chang Z, Liu Z, Cui L. A review of original articles published in the emerging field of radiomics. Eur J Radiol 2020;127:108991.

[5] Isaksson LJ, Pepa M, Zaffaroni M, Marvaso G, Alterio D, Volpe S, Corrao G, Augugliaro M, Starzyńska A, Leonardi MC, et al. Machine learning-based models for prediction of toxicity outcomes in radiotherapy. Front Oncol 2020;10:790.

[6] Ferro M, de Cobelli O, Musi G, Del Giudice F, Carrieri G, Busetto GM, Falagario UG, Sciarra A, Maggi M, Crocetto F, et al. Radiomics in prostate cancer: an up-to-date review. Ther Adv Urol 2022;14:17562872221109020.

[7] Kothari G. Role of radiomics in predicting immunotherapy response. J Med Imaging Radiat Oncol 2022;66(4):575–91.

[8] Zekić-Sušac M, Pfeifer S, Šarlija N. A comparison of machine learning methods in a high-dimensional classification problem. Bus Syst Res: Int J Soc Adv Innov Res Econ 2014;5(3):82–96.

[9] Pappu V, Pardalos PM. High-dimensional data classification. In: Clusters, orders, and trees: Methods and applications. Springer; 2014, p. 119–50.

[10] Spooner A, Chen E, Sowmya A, Sachdev P, Kochan NA, Trollor J, Brodaty H. A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction. Sci Rep 2020;10(1):1–10.

[11] Somepalli G, Goldblum M, Schwarzschild A, Bruss CB, Goldstein T. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. 2021, arXiv preprint arXiv:2106.01342.

[12] Kossen J, Band N, Lyle C, Gomez AN, Rainforth T, Gal Y. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. Adv Neural Inf Process Syst 2021;34:28742–56.

[13] Arik SÖ, Pfister T. Tabnet: Attentive interpretable tabular learning. In: Proceedings of the AAAI conference on artificial intelligence, Vol. 35. 2021, p. 6679–87, 8.

[14] Popov S, Morozov S, Babenko A. Neural oblivious decision ensembles for deep learning on tabular data. 2019, arXiv preprint arXiv:1909.06312.

[15] Abutbul A, Elidan G, Katzir L, El-Yaniv R. Dnf-net: A neural architecture for tabular data. 2020, arXiv preprint arXiv:2006.06465.

[16] Klambauer G, Unterthiner T, Mayr A, Hochreiter S. Self-normalizing neural networks. Adv Neural Inf Process Syst 2017;30:972–82.

[17] Badirli S, Liu X, Xing Z, Bhowmik A, Doan K, Keerthi SS. Gradient boosting neural networks: Grownet. 2020, arXiv preprint arXiv:2002.07971.

[18] Wang R, Shivanna R, Cheng D, Jain S, Lin D, Hong L, Chi E. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In: Proceedings of the web conference 2021. 2021, p. 1785–97.

[19] Song W, Shi C, Xiao Z, Duan Z, Xu Y, Zhang M, Tang J. Autoint: Automatic feature interaction learning via self-attentive neural networks. In: Proceedings of the 28th ACM international conference on information and knowledge management. 2019, p. 1161–70.

[20] Gorishniy Y, Rubachev I, Khrulkov V, Babenko A. Revisiting deep learning models for tabular data. Adv Neural Inf Process Syst 2021;34:18932–43.

[21] Shwartz-Ziv R, Armon A. Tabular data: Deep learning is not all you need. Inf Fusion 2022;81:84–90.

[22] Kadra A, Lindauer M, Hutter F, Grabocka J. Regularization is all you need: Simple neural nets can excel on tabular data. 2021, arXiv preprint arXiv:2106.11189.

[23] Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. CatBoost: unbiased boosting with categorical features. Adv Neural Inf Process Syst 2018;31:6638–49.

[24] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd Acm sigkdd international conference on knowledge discovery and data mining. 2016, p. 785–94.

[25] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y. Lightgbm: A highly efficient gradient boosting decision tree. Adv Neural Inf Process Syst 2017;30.

[26] Rajkumar S. Winning solutions of kaggle competitions. 2022, https://www.kaggle.com/code/sudalairajkumar/winning-solutions-of-kaggle-competitions/notebook, online: accessed 2022-08-24.

[27] Gorishniy Y, Rubachev I, Babenko A. On embeddings for numerical features in tabular deep learning. 2022, arXiv preprint arXiv:2203.05556.

[28] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Adv Neural Inf Process Syst 2017;30.

[29] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al. An image is worth 16x16 words: Transformers for image recognition at scale. 2020, arXiv preprint arXiv:2010.11929.

[30] baosenguo. Kaggle-MoA 2nd place solution. 2020, https://github.com/baosenguo/Kaggle-MoA-2nd-Place-Solution, online: accessed 2022-08-21.

[31] Isaksson LJ, Pepa M, Summers P, Zaffaroni M, Vincini MG, Corrao G, Mazzola GC, Rotondi M, Presti GL, Raimondi S, et al. Comparison of automated segmentation techniques for magnetic resonance images of the prostate. 2022, PREPRINT (Version 1) Available at Research Square https://doi.org/10.21203/rs.3.rs-1850296/V1.

[32] Isaksson LJ, Summers P, Raimondi S, Gandini S, Bhalerao A, Marvaso G, Petralia G, Pepa M, Jereczek-Fossa BA. Mixup (sample pairing) can improve the performance of deep segmentation networks. J Artif Intell Soft Comput Res 2022;31:29–39.

[33] Isaksson LJ, Summers P, Bhalerao A, Gandini S, Raimondi S, Pepa M, Zaffaroni M, Corrao G, Mazzola GC, Rotondi M, et al. Quality assurance for automatically generated contours with additional deep learning. Insights Imaging 2022;13(1):1–10.

[34] Van Griethuysen JJ, Fedorov A, Parmar C, Hosny A, Aucoin N, Narayan V, Beets-Tan RG, Fillion-Robin J-C, Pieper S, Aerts HJ. Computational radiomics system to decode the radiographic phenotype. Cancer Res 2017;77(21):e104–7.

[35] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25rd ACM SIGKDD international conference on knowledge discovery and data mining. 2019.

[36] Chicco D, Tötsch N, Jurman G. The matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. BioData Min 2021;14(1):1–22.

[37] Boughorbel S, Jarray F, El-Anbari M. Optimal classifier for imbalanced data using matthews correlation coefficient metric. PLoS One 2017;12(6):e0177678.

[38] Chicco D, Jurman G. The advantages of the matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 2020;21(1):1–13.

[39] Thung K-H, Wee C-Y. A brief review on multi-task learning. Multimedia Tools Appl 2018;77(22):29705–25.

[40] Crawshaw M. Multi-task learning with deep neural networks: A survey. 2020, arXiv preprint arXiv:2009.09796.

[41] Vandenhende S, Georgoulis S, Proesmans M, Dai D, Van Gool L. Revisiting multi-task learning in the deep learning era. 2020, arXiv preprint arXiv:2004.13379, vol. 2, no. 3.

[42] Chicco D, Warrens MJ, Jurman G. The matthews correlation coefficient (MCC) is more informative than Cohen's Kappa and Brier score in binary classification assessment. IEEE Access 2021;9:78368–81.

[43] Lobo JM, Jiménez-Valverde A, Real R. AUC: a misleading measure of the performance of predictive distribution models. Global Ecol Biogeogr 2008;17(2):145–51.

[44] Halligan S, Altman DG, Mallett S. Disadvantages of using the area under the receiver operating characteristic curve to assess imaging tests: a discussion and proposal for an alternative approach. Eur Radiol 2015;25(4):932–9.

[45] Byrne S. A note on the use of empirical AUC for evaluating probabilistic forecasts. Electron J Stat 2016;10(1):380–93.

Further reading

[1] Jakobsen JC, Gluud C, Wetterslev Jr, Winkel P. When and how should multiple imputation be used for handling missing data in randomised clinical trials–a practical guide with flowcharts. BMC Med Res Methodol 2017;17(1):1–10.

[2] Sterne JA, White IR, Carlin JB, Spratt M, Royston P, Kenward MG, Wood AM, Carpenter JR. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. Bmj 2009;338.

[3] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in python. J Mach Learn Res 2011;12:2825–30.