Radiomics in Head and Neck Cancer

Geir Severin Rakh Elvatun Langberg

December 5, 2018

Contents

Abstract

Assessment of patient treatment response constitutes a prerequisite to personalized cancer therapy. The field of medical study, *radiomics*, hypothesizes quantification of disease state through medical image features. In this radiomics study, 36 combinations of predictive and feature selection algorithms were assessed in classification of patient treatment outcome.

A total of 188 radiomic features were extracted from 18F-FDG-PET/CT scans of 198 head and neck cancer patients, and combined with clinical variables. Each patient had undergone scanning prior to radiotherapy in the period between 2007 and 2014. Both PET and CT scans were subjected to square root transformation with subsequent discretization into 16 and 128 bins respectively. A total of 87 features were included in the experiments. Both disease-free survival and locoregional relapse were considered as radiotherapy outcomes.

The partial least squares algorithm, in conjunction with permutation importance of L1 regularized logistic regression for feature selection, was identified as the optimal modeling procedure. The model evaluation procedure combined the .632+ bootstrap method with AUC as objective function. A .632+ based AUC of 0.71 ± 0.011 constituted highest obtainable score.

The employment of predictive models in clinical decision support serves as a complement to individual treatment selection and cancer diagnostics.

Introduction

Predicting the treatment outcome constitutes a prerequisite to personalized cancer therapy. Improving patient screening procedures serves as a complement to individual treatment selection and cancer diagnostics.

With human cancers exhibiting phenotypic differences, medical imaging facilitates non-invasive spatial and temporal characterization of tumor heterogeneity.

The field of medical study termed *radiomics* hypothesizes that the state of disease may be quantified by image features describing tumor physiology characteristics [10]. To promote clinical decision support, radiomic features can be combined in constructing biomarkers with prognostic and predictive performance.

Studies have demonstrated the potential of radiomics in discriminating between clinical outcomes across different types of tumors and modalities. Increased predictive performance have been reported by including quantitative radiomic features compared to exclusive analysis of clinical parameters [2], [12]. An initiative to standardize radiomic analysis pipelines has been commenced through the development of the PyRadiomics software package [9]. Pyradiomics is an open-source Python package accepted in Cancer Research, including tools to conduct preprocessing and feature extraction from images.

Through the employment of computational methods and models, automated procedures can be constructed to predict patient treatment responses. The formulation of discriminative functions for estimating future observations constitutes a fundamental principle in machine learning. Thus, patterns identified in radiomics may be utilized in assisting clinical decisions.

Due to the number of defined radiomic features, *the curse of dimensionality* is readily introduced in analysis pipelines. Integration of feature selection procedures in data processing procedures contributes to reducing the feature space and enables ranking of features according to informative potential .

This study aims at identifying the optimal modeling procedure for predicting treatment response in a cohort of 198 head and neck cancer patients. Each patient had undergone a 18F-FDG-PET/CT scan prior to radiotherapy in the period between 2007 and 2014. The original data set was provided by, and belongs to, Oslo University Hospital. Two forms of treatment response were considered: loco-regional relapse, hereby LRR, and disease-free survival, hereby DFS. Additionally, the features obtained in each experiment were ranked according to predictive relevance with respect to the identified modeling procedure.

Theory

Recommendations and frameworks are described in the literature for performing unbiased model comparisons. An exhaustive search is guaranteed to find the best performing model in a set of candidates [22], [19], [4].

Model Selection and Comparison

The recommended procedure in estimating the general algorithm performance is to divide the original data set into a training and validation set [4]. Information leakage is prevented as the model is built from the training set and evaluated on the validation set, separately.

According to the Out-of-Bag, hereby OOB, bootstrap method, n samples are randomly selected with replacement from the original data set, X, generating a training set, $X_T \subset X$, and a validation set $X_V = X \setminus X_T$ [22]. Compared to random splitting techniques, such as the K-fold cross-validation, hereby CV, the random sampling approach allows a variable number of samples, although including repetitions. In addition, the variance associated with CV may be reduced with the OOB technique [22], [3].

The performance of a learning algorithm, $\lambda \in \Lambda$, given a combination of hyperparameters θ, from the hyperparameter space Θ, can be ranked in terms of a score function $L(\cdot)$. Hyperparameter optimization according to the grid search technique exhaustively determines the configuration $\theta \in \Theta$ corresponding to the highest score [4]. The score function evaluates the quality of the algorithm output, $\pi_{\theta,\lambda} = \lambda(X,\theta)$ by comparing predictions to a ground truth. That is, the algorithm performance is evaluated as the achieved score, $L(\pi_{\theta,\lambda}, X_H)$. The grid search procedure in conjunction with K OOB iterations is outlined in algorithm 1.

Algorithm 1 Grid Search Out-of-Bag

procedure GRIDSEARCHOOB($\Theta_\lambda, K, X, y, \lambda, L$)
  $\pi_{max}, \theta_{opt} \leftarrow 0, \emptyset$ ▷ Setup
  for $\theta \in \Theta_\lambda$ **do**
    for $(X_{T\kappa}, y_{T\kappa}) \subset (X,y) \mid \kappa \in [1,K]$ **do**
      $(X_{V\kappa}, y_{V\kappa}) \leftarrow (X,y) \setminus (X_{T\kappa}, y_{T\kappa})$
      $Train(\lambda(\theta, X_{T_\kappa}), y_{T_\kappa})$
      $\pi_{\theta,\lambda,\kappa} \leftarrow L(\lambda(\theta, X_{V_\kappa}), y_{V_\kappa})$
    $\overline{\pi} = \frac{1}{K}\sum_\kappa \pi_{\theta,\lambda,\kappa}$
    if $\overline{\pi} > \pi_{max}$ **then**
      $\pi_{max} := \overline{\pi}$
      $\theta_{opt} := \theta$
  return $(\lambda(\theta_{opt}), \pi_{max})$

In algorithm 1, the validation performance of a model $\lambda(\theta, X_{V_\kappa})$ with respect to a score function $L(\cdot)$, is denoted $\pi_{\theta,\lambda,\kappa}$. The general model performance $\overline{\pi}$ is given as the average performance across all K OOB samples. The model resulting in the highest average performance is selected as optimal.

Building on the model selection procedure given in algorithm 1, estimating the average model error in a nested procedure is the recommended approach in comparing the performance of different algorithms [4]. The nested OOB procedure described in algorithm 2 includes two levels of subset sampling. Average model performance is estimated at the first level of iterations, while the second level of iterations performs an OOB grid search as given in algorithm 1.

Algorithm 2 Nested Out-of-Bag

procedure NESTEDOOB($X, y, \lambda, \Theta_\lambda, L$)
  $\pi \leftarrow \emptyset$ ▷ Setup
  for $(X_{T\kappa}, y_{T\kappa}) \subset (X, y) \mid \kappa \in [1, K]$ **do**
    $(X_{V\kappa}, y_{V\kappa}) \leftarrow (X, y) \setminus (X_{T\kappa}, y_{T\kappa})$
    $\lambda(\theta_{opt}) \leftarrow$ GridSearchOOB($\Theta_\lambda, K, X_{T_\kappa}, y_{T_\kappa}, \lambda, L$) ▷ Contains the inner loop
    $Train(\lambda(\theta, X_{T_\kappa}), y_{T_\kappa})$
    $\pi_{\theta,\lambda,\kappa} \leftarrow L(\lambda(\theta, X_{V_\kappa}), y_{V_\kappa})$
  return $\frac{1}{K}\sum_{\kappa \in K} \pi_{\theta,\lambda,\kappa}$

Algorithm 2 describes a setup for performing model comparisons. The average performance of each model, $\lambda(\theta_{opt})$ selected in a grid search procedure, is assessed by repeated OOB samplings.

Algorithm 3 proposes a procedure for conducting repeated model comparison experiments including feature selection. Given model candidates, $\lambda \in \Lambda$, and feature selection algorithms, $\psi \in \Psi$, results for each combination (λ_θ, ψ) are obtained for identification of the optimal modeling procedure. Each experiment is repeated N times per λ and ψ for a given comparison scheme, $CS(\cdot)$.

Algorithm 3 Model Comparison

procedure MODELCOMPARISON($X, y, \Lambda, \Theta, CS, \Psi, L$)
  $L \leftarrow \emptyset$ ▷ Setup
  for $\lambda \in \Lambda$ **do**
    $\Theta_\lambda \subset \Theta$
    for $\psi \in \Psi$ **do**
      while $n < N$ **do**
        $L_{\lambda,\psi,n} \leftarrow CS(X, y, \lambda, \Theta_\lambda, \psi, L)$
  return L

As included in algorithm 3, transformations occur at the deepest level of iterations as recommended to avoid information leakage between steps in the procedure.

The .632+ Rule

The .632+ bootstrap method was proposed by Efron and Tibhirani as an improved alternative to the CV approach in estimating model error rates [7]. The .632+ rule builds on the .632 estimator stated as

$$\varepsilon_{.632} = 0.367\varepsilon_r + 0.632\varepsilon_v$$

where ε_r and ε_v denotes the resubstitution and apparent accuracy, respectively[6]. The coefficients e^{-1} and $1-e^{-1}$ are motivated by the average support of 63.2% original observations found in bootstrap samples. The .632+ rule stated in equation 1 as

$$\varepsilon_{.632+} = \omega\varepsilon_r + (1-\omega)\varepsilon_v \tag{1}$$

was suggested to manage situations with model overfitting. The upward bias in ε_v of the .632 estimator is accounted for by replacing the weights of $\varepsilon_{.632}$ with ω given by

$$\omega = \frac{0.632}{1-0.368R}$$

including a rate of relative overfitting, R, where

$$R = \frac{\varepsilon_r - \varepsilon_v}{\gamma - \varepsilon_v}$$

The no-information error rate, γ, for a dichotomous classification problem is given as

$$\gamma = p_1(1-q_1) + (1-p_1)q_1$$

where p_1 and q_1 denote the number of observations and predictions equal to one, respectively.

If $\gamma \leq \varepsilon_r$ or $\varepsilon_r < \gamma \leq \varepsilon_v$, then R must be bounded in order to maintain a fractional range. An alternative expression for $\varepsilon_{.632+}$ in equation 1 ensuring $R \in [0,1]$ is given by equation 2

$$\varepsilon_{.632+} = \varepsilon_{.632} + (\overline{\varepsilon_r} - \varepsilon_v)\frac{0.368(1-0.368)\overline{R}}{1-0.368\overline{R}} \tag{2}$$

where $\overline{\varepsilon_r} = \min\{\varepsilon_r, \gamma\}$, and

$$\overline{R} = \begin{cases} R & \varepsilon_r, \gamma > \varepsilon_v \\ 0 & \varepsilon_r, \gamma \leq \varepsilon_v \end{cases}$$

The redefinition of $\varepsilon_{.632+}$ including $\overline{\varepsilon_r}$ and $\overline{R}$ maintains the properties of the relative overfitting rate. If $\varepsilon_r = \varepsilon_v$, then $\overline{R} = 0$ represents no overfitting, while $\overline{R} = 1$ denotes the maximum amount of overfitting. Additionally, $\omega \in [0.632, 1] \propto \overline{R}$.

Feature Importance Ranking

Consider a set of selected features $X_{sub} \subset X$. The principle of feature permutation importance ranking builds on the change in model performance with respect to feature permutations [1]. Algorithm 4 describes the permutation approach in feature importance ranking

Algorithm 4 Feature Permutation Importance

procedure FEATUREIMPORTANCE($X, y, \lambda, \theta, L, K$)
  $I \leftarrow \emptyset$
  $L_0 \leftarrow L(\lambda(\theta, X), y)$ ▷ Baseline score
  for $\kappa \in K$ **do**
    for $x \in X$ **do**
      $X_{perm} \leftarrow SHUFFLE(x)$ ▷ Permute feature x
      $L \leftarrow L(\lambda(\theta, X_{perm}), y)$
      $I_\kappa \leftarrow L_0 - L$
  return $\frac{1}{K}\sum_{\kappa \in K} I_\kappa$ ▷ Return the average permutation importance

Initially, a baseline performance L_0 is computed prior to the permutation steps. The purpose of L_0 is to serve as a benchmark for scores L_κ obtained by recomputing model performance after permuting individual features.

Methods

The basis for this study was carried out by A. D. Midtfjord analyzing the same data set provided by the Oslo University Hospital [14]. The data processing steps, model candidates and feature selection algorithms were selected according to recommendations described in the base study. Treatment responses were denoted: y_{LRR} for loco-regional relapse, and y_{DFS} for disease-free survival, with event ratios of 32.3% for y_{DFS}, and 24.7% for y_{LRC}. All material concerning feature extraction, experiments and parameter settings can be found at the project repository [11].

The project was carried out in accordance with Norwegian law.

Feature Extraction and Data Processing

The clinical parameters of each patient comprised:

- Patient age and sex
- TNM staging
- ICD10 classification
- Histology based on tissue samples

- HPV status
- ECOG description of patient condition prior to treatment
- Charlson comorbidity index
- Days with Naxogin medication
- The number of Cisplatin courses

Additional available features were the metabolic tumor volume, hereby MTV, total lesion glycolysis and standardized uptake value peak obtained from PET scans. Dummy encoding was applied to categorical variables resulting in 49 clinical parameters. Patients lacking HPV status information or histology diagnosis were encoded as separate categories.

The PET and CT image values, x were transformed according to

$$f(x) = \sqrt{|x|}$$

where $f(x)$ denotes the transformed value, and quantified into 16 bins for PET and 128 bins for CT. The bin width, δ, resulting in the target bin counts, N, for PET and CT was calculated by

$$\delta = \frac{\frac{1}{M}\sum_{i=1}^{M} \max\{I_i\} - \min\{I_i\}}{N}$$

given a set of images $\{I_i\}_{i=1}^{M}$ per modality.

Feature extraction was conducted with the PyRadiomics package, version 2.0.1. In total, 118 features were extracted from PET and CT images separately. Among these features were 13 shape features, 15 first order features and 74 image texture features. The texture features encompassed GLCM, GLRLM, GLSZM, GLDM and NGTDM features of which details can be found in the PyRadiomics documentation [9]. The parameter settings for which the extraction was carried out are located at the project repository [11]. The feature extraction procedure is implemented in the Jupyter Notebook `feature_extraction.ipynb`, with helper functions included in the script `feature_extraction.py`.

Unattainable or correlated features were removed from the data set subsequent to feature extraction. Feature correlation was evaluated with the Pearson correlation coefficient and a 0.85 threshold. The feature correlation matrices prior and subsequent to filtering are available at GitHub under `figures`, together with a complete list of all removed features in `feature_extraction/failed_extraction`. The steps in the data processing procedure was implemented in the Jupyter Notebook `data_preparation.ipynb` available at GitHub.

The final data set contained 198 samples and 87 features.

Model Comparisons

Nested OOB experiments were carried out according to the procedures described in algorithm 3 and algorithm 2. The predictive model candidates denoted by

- Logistic regression
- Partial least squares regression
- Gaussian Naive Bayes classifier
- Linear discriminant classifier
- Quadratic discriminant classifier
- Adaboost classifier

were collected from scikit-learn package, version 0.19.1 [16]. A single depth decision tree classifier constituted a base estimator for the AdaBoost ensemble. The feature selection algorithms denoted by

- Variance threshold
- ReliefF
- Mutual information
- L1 or L2 regularized logistic regression permutation importance
- Random forest permutation importance

were either implemented based on, or collected from scikit-learn package. Implementations of feature selection algorithms are available in `feature_selection.py` at the project repository [11], [16].

Ten repetitions of model comparison experiments were carried out with algorithm 3 for 50 OOB iterations. Each experiment was conducted for both target variables, y_{DFS} and y_{LRR}. Feature sets containing the 15 most informative features were selected in each grid search run. Feature Features were subjected to Z-score normalization prior to model training and feature selection, except for the variance threshold procedure relying on differences in feature variances. Models were evaluated according to equation 2 with the Area Under Curve, hereby AUC, score as resubstitution and apparent accuracy. Regarding subspace methods, the number of retained components was set equal to the number of features selected in each iteration. The general model performance, $\overline{\varepsilon_{.632+}}$, was calculated from the average scores obtained in each experiment. The best performing model, $\lambda(\theta_\lambda) \in \Lambda(\Theta)$, and feature selector algorithm, $\phi \in \Phi$, were selected according to

$$\max_{\overline{\varepsilon_{.632+}}} \{(\Lambda(\Theta), \Phi)\}$$

giving the optimal combination of learning algorithm and feature selector among candidates. The experimental setup is given in `model_comparison_setup.py` available in the project repository [11].

Feature Ranking

A set of consensus features was created by combining the features selected in each repetition of model comparison experiments. Ten repetitions of 50 OOB iterations with permutation importance ranking was carried out to estimate general consensus feature importances. The permutation importance procedure included $\lambda(\theta_\lambda)$ with the optimal hyperparameter configuration, θ_λ selected across all model comparison experiments. General feature importance were calculated as the average permutation importance scores across all iterations.

Materials

This study was based on open-source software. All code was developed in Python, version 3.6.6 and is available at the project repository [11], [17]. Code dependencies including external library versions are listed in requirements [11].

Simulations and experiments were carried out with a Apple MacBook Air©, with 8 GB 1600 MHz DDR3 memory and a 1.6 GHz Intel Core i5 processor.

Results

To assess the performance of different modeling procedures in predicting treatment outcome in a cohort of 198 head and neck cancer patients, radiomic features were extracted from 18F-FDG-PET/CT images. A total of 87 features, including clinical variables, were included in the experiments.

Model Comparisons

Combinations of six predictive and six feature selection algorithms were compared for both LRR and DFS treatment responses in separate experiments. The model comparison framework was built from a nested OOB procedure including the .632+ estimator. Each experiment was repeated ten times for 50 OOB iterations. The collected model performance scores averaged across experiments is shown in figure 1 for DFS response and figure 2 for LRR response.

Figure 1:
Averaged model $\overline{\varepsilon_{(.632+)}}$ estimates based on AUC, and including standard deviations with respect to DFS. All combinations of estimators and feature selectors were compared in ten repetitions of 50 nested OOB iterations.

According to figure 1, the highest performing estimator and feature selector pair constitutes the partial least squares regression, hereby PLSR, algorithm in combination with the feature permutation importance procedure and L1 regularized logistic regression, hereby PILRL1. The average PLSR performance is equivalent to the baseline performance achieved by predicting the majority class of y_{DFS} at 71% for all samples. Combining PLSR, and replacing L1 with L2 regularization in the PILRL1 procedure, reduce the performance by

0.2%, but increases stability by 0.29%. Figure 2 shows the result of model comparison experiments with LRR as target variable.

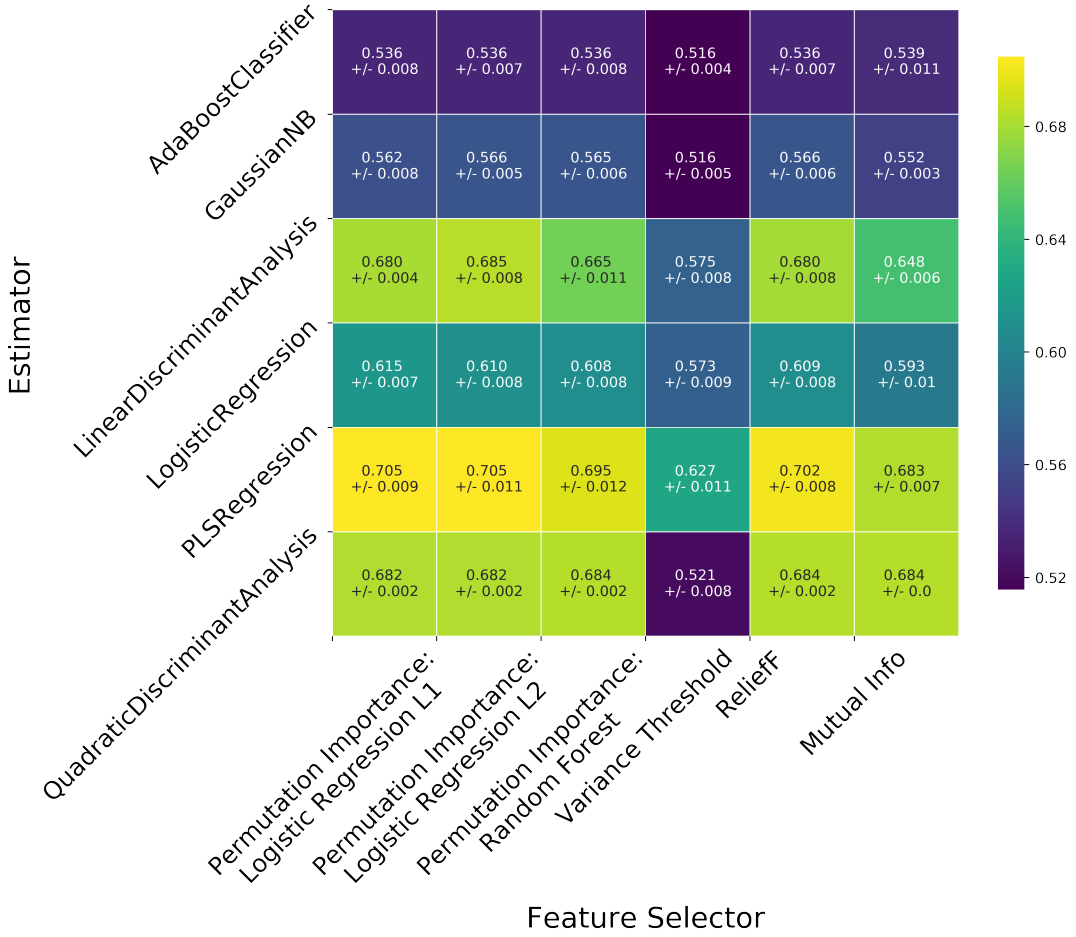

Figure 2:
Averaged model $\overline{\varepsilon_{(.632+)}}$ estimates based on AUC, and including standard deviations with respect to LRR. All combinations of estimators and feature selectors were compared in ten repetitions of 50 nested OOB iterations.

The trend among results shown in figure coincides with figure . However, the general performance of PLSR is lower than the baseline of 75.3% for the majority class of y_{LRR}. The hyperparameter configuration associated with the highest PLSR performance were tolerance equal to 0.001 and 70 components.

Feature Ranking

By identification of the optimal learning algorithm, consensus features were ranked in a permutation importance procedure to assess the predictive relevance. The five highest ranked features with respect to the PLSR AUC score across ten repetitions of 50 OOB iterations, for both DFS and LRR target variables, are shown i figure 3. In figure 3, δ AUC denotes the feature importance with respect to AUC.

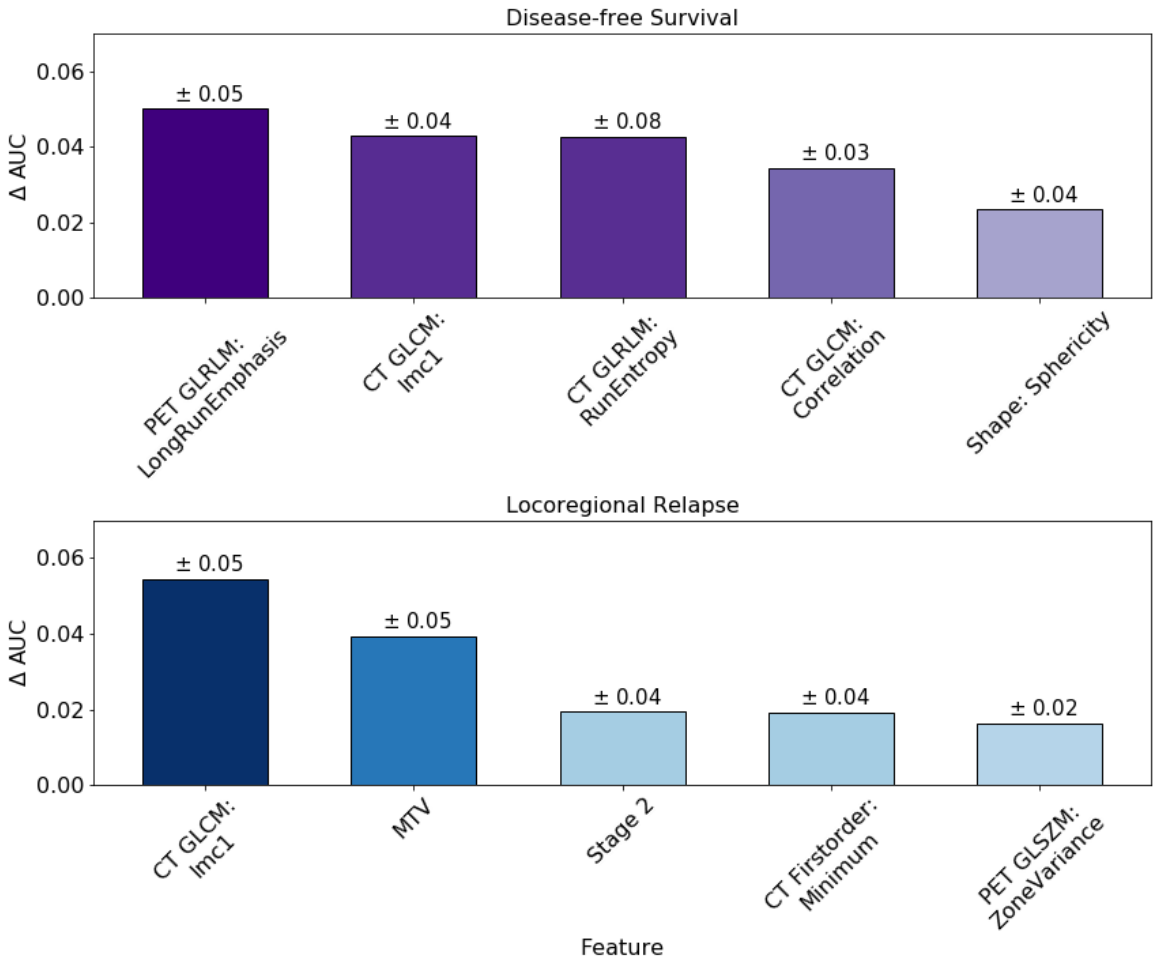

Figure 3:
Average feature importance across ten repetitions of 50 OOB iterations with PLSR permutation importance and AUC score. The PLSR algorithm was instantiated with 70 components and a tolerance of 0.001. Bar annotations denote the associated standard deviations of each feature importance score.

The highest ranked features with respect to DFS and LRR in figure are the Long Run

Emphasis, hereby LRE, PET textural feature and the Informational Measure of Correlation 1, hereby IMC1, CT textural feature, respectively. The LRE quantifies the length of run lengths and texture coarseness, while IMC1 indicates texture complexity [9]. The IMC1 CT feature are present in both subsets of feature selected with respect to DFS and LRR. Only the LRE, CT IMC1 and MTV features are ranked at the level of corresponding standard deviations. The remaining features are associated with standard deviations exceeding the respective importance scores.

Discussion

Assessing patient treatment response constitutes an inherent part of personalized cancer medicine. Extracting information from radiomic data enables the employment of predictive models in patient screening strategies. In this study, the performance of 36 different modelling procedures were compared with respect to DFS and LRR in a head and neck cancer cohort. The preprocessing steps and model candidates were selected based on the results obtained in [14].

Data Processing

Studies have demonstrated that feature robustness regarding CT and PET image vary according to the level of discretization [13], [21]. Recommendations adopted by PyRadiomics describe a fixed bin width producing between 30 and 128 bins [**"cite –pyrad-binning"**]. A total of 16 bins was used to quantify PET images. However, all redundant features originated from PET images this suggests that the quantification level should be reconsidered. Flat regions in PET images were identified as a cause for failed extraction of textural features. Regarding CT images, no miscalculation of features were associated with discretization into 128 bins.

In total, 42 features was identified as redundant where one feature contained missing values, while the rest were associated with failed extraction. One of the features was related to PET first order entropy, while the remaining features represented the texture categories. In addition, alternative filtering procedures provided by the PyRadiomics package can be investigated.

Modeling Procedure

Compared to the base study achieving 0.66 AUC from model comparison experiments, figure 1 and 2 shows a general improvement in scores across all models. The highest obtained score of .632+ based AUC yielded 0.71 ± 0.011, which exceeds the baseline of 0.67 associated with the majority DFS class. The highest score obtained for LRR corresponded to 0.70 ± 0.009 which is below the baseline according to the LLR majority event ratio of

0.75. However, the improvement in model performances may be due to the .632+ estimator correcting for variance and downward bias associated with K-fold CV [7]. Additionally, the .632+ bootstrap method accounts for model overfitting as opposed to the K-fold CV procedure.

Comparing figure 1 and 2 reveals similar patterns in model performances despite being linked to different target outcomes. Observing similar trends in model behaviour could express that the same underlying pattern is detected in the data across both targets. However, since the accuracy scores obtained for DFS exceeds LRR, the results imply that DFS is the most informative response.

The general trends of model behaviours, as shown in figure 1 and 2, suggests that variations in performance scores are more closely linked to the estimators compared to the feature selectors. Parmar et al. reported that the choice of estimator algorithm accounted for more than five times the score variance compared to the choice of feature selection algorithm. Thus, the tendencies inferred from model comparison results corresponds with Parmar and Grossman et al. On the other hand, despite claiming an unbiased comparison framework, Parmar et al. does not apply a nested approach in comparing algorithms. Also, by adopting parameter configurations from another study the analyzed algorithms may be provided with different outsets potentially favouring particular models.

Presenting the median rather than the mean of results may provide more accurate representation in the analysis of data with significant sample variations.

Feature Importance

The feature permutation importance procedure is claimed to be an unbiased wrapper approach to feature selection [20], [1]. However, the predictive information detected by the wrapped algorithm may not be transferable to the considered learning algorithm producing suboptimal feature sets.

Feature importance ranking comprised ten repetitions of 50 OOB iterations, following with ten repetitions of assessing feature permutation importance. Thus, the importance of each feature was assessed ten times per OOB sample. The variation in results suggests that additional data is required to prevent random sampling effects leading to statistical insignificance.

Observing that the standard deviations are either greater or approximately equal to the corresponding feature importance scores, suggests that the results are statistically insignificant.

Recommendations for Further Study

To assess the described results, in addition to extending radoimics insights and methodology, the following suggestions are presented:

- Explore characteristics of the data set to identify variations between samples, descriptive statistics and feature distributions.
- Include boosting methods to address model bias.
- Evaluate a range of filtering and discretization transformations to identify the optimal image preprocessing with respect to outcome prediction.
- Explore the similarity between PET and CT feature sets with matrix correlation coefficient methods.
- Attempt sample augmentation with Generative Adversarial Networks, or the Synthetic Minority Over-sampling Technique [8], [5].
- Apply the Local Interpretable Model-Agnostic Explanations algorithm in interpreting model candidates [18].
- Consider feature engineering approaches to handle feature correlation.
- Inspect a range of performance metrics as alternatives to AUC.
- Identify publicly available data sets to benchmark model comparison procedures.
- Consider extracting deep features and transfer learning to enhance prediction accuracy, or in combination with engineered features.

Acknowledgements

Sincere gratitude is directed to Aurora Grønvoll, Cecilia M. Futsæther, Kristian H. Liland, Oliver Tomic and Ulf Indahl at NMBU for their contribution and collaboration.

References

[1] André Altmann et al. “Permutation importance: a corrected feature importance measure”. In: *Bioinformatics* 26.10 (2010), pp. 1340–1347.

[2] Marta Bogowicz et al. “Computed tomography radiomics predicts hpv status and local tumor control after definitive radiochemotherapy in head and neck squamous cell carcinoma”. In: *International Journal of Radiation Oncology* Biology* Physics* 99.4 (2017), pp. 921–928.

[3] Leo Breiman. *Out-of-bag estimation*. 1996.

[4] Gavin C Cawley and Nicola LC Talbot. “On over-fitting in model selection and subsequent selection bias in performance evaluation”. In: *Journal of Machine Learning Research* 11.Jul (2010), pp. 2079–2107.

[5] Nitesh V Chawla et al. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[6] Bradley Efron. “Estimating the error rate of a prediction rule: improvement on cross-validation”. In: *Journal of the American statistical association* 78.382 (1983), pp. 316–331.

[7] Bradley Efron and Robert Tibshirani. “Improvements on cross-validation: the 632+ bootstrap method”. In: *Journal of the American Statistical Association* 92.438 (1997), pp. 548–560.

[8] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[9] Joost JM van Griethuysen et al. “Computational radiomics system to decode the radiographic phenotype”. In: *Cancer research* 77.21 (2017), e104–e107.

[10] Patrick Benedict Hans Juan Grossmann. “Defining the biological and clinical basis of radiomics: towards clinical imaging biomarkers”. English. PhD thesis. Netherlands: Maastricht University, 2018. ISBN: 9789461598103. DOI: 10.26481/dis.20180308pg.

[11] *head-and-neck*. `https://github.com/GSEL9/head-and-neck`. Accessed: 2018-12-2.

[12] Yanqi Huang et al. “Radiomics signature: A potential biomarker for the prediction of disease-free survival in early-stage (i or ii) non—small cell lung cancer”. In: *Radiology* 281.3 (2016), pp. 947–957.

[13] Ralph TH Leijenaar et al. “The effect of SUV discretization in quantitative FDG-PET Radiomics: the need for standardized methodology in tumor texture analysis”. In: *Scientific reports* 5 (2015), p. 11075.

[14] Alise Danielle Midtfjord. “Prediksjon av behandlingsutfall for hode- og halskreft ved bruk av radiomics av PET/CT-bilder (eng.: Prediction of treatment outcome for head and neck cancer using radiomics of PET/CT images)”. 2018.

[15] Chintan Parmar et al. “Machine learning methods for quantitative radiomic biomarkers”. In: *Scientific reports* 5 (2015), p. 13087.

[16] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[17] *Python*. `https://www.python.org/`. Accessed: 2018-12-2.

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Why should i trust you?: Explaining the predictions of any classifier”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM. 2016, pp. 1135–1144.

[19] Jun Shao. “Bootstrap model selection”. In: *Journal of the American Statistical Association* 91.434 (1996), pp. 655–665.

[20] Carolin Strobl et al. “Bias in random forest variable importance measures: Illustrations, sources and a solution”. In: *BMC bioinformatics* 8.1 (2007), p. 25.

[21] Florent Tixier et al. “Intratumor heterogeneity characterized by textural features on baseline 18F-FDG PET images predicts response to concomitant radiochemotherapy in esophageal cancer”. In: *Journal of Nuclear Medicine* 52.3 (2011), p. 369.

[22] Sudhir Varma and Richard Simon. “Bias in error estimation when using cross-validation for model selection”. In: *BMC bioinformatics* 7.1 (2006), p. 91.

Appendix

Supplementary material is available at GitHub [11].