Supplementary material

Table S1. Radiomic features

Feature class		Feature name
Shape Features	Shape 2D	- Mesh Surface - Spherical Disproportion - Pixel Surface - Maximum 2D diameter - Perimeter - Major Axis Length - Perimeter to Surface ratio - Minor Axis Length - Sphericity - Elongation
	Shape 3D	- Mesh Volume - Maximum 3D diameter - Voxel Volume - Maximum 2D diameter (Slice) - Surface Area - Maximum 2D diameter (Column) - Surface Volume ratio - Maximum 2D diameter (Row) - Sphericity - Major Axis Length - Elongation - Flatness
Histogram-Based Features (First-Order Statistics)		- 10Percentile - 90Percentile - Energy - Entropy - Interquartile Range - Kurtosis - Maximum - Mean Absolute Deviation - Mean - Median - Minimum - Range - Robust Mean Absolute Deviation - Root Mean Squared - Skewness - Total Energy - Uniformity - Variance

Textural Features (Second-Order Statistics)	Gray Level Co-occurrence Matrix Features (GLCM)	- Autocorrelation - Informational Measure of Correlation (IMC1) - Cluster Prominence - Informational Measure of Correlation (IMC2) - Joint Average - Inverse Difference Moment (IDM) - Cluster Shade - Inverse Difference Moment Normalized (IDMN) - Cluster Tendency - Maximal Correlation Coefficient (MCC) - Contrast - Inverse Difference (ID) - Correlation - Inverse Difference Normalized (IDN) - Difference Average - Inverse Variance - Difference Entropy - Maximum Probability - Difference Variance - Sum Average - Joint Energy - Sum Entropy - Joint Entropy - Sum of Squares
	Gray Level Dependence Matrix (GLDM)	- Dependence Entropy (DE) - Low Gray Level Emphasis (LGLE) - Dependence Variance (DV) - High Gray Level Emphasis (HGLE) - Gray Level Variance (GLV) - Gray Level Non-Uniformity (GLN) - Dependence Non-Uniformity (DN) - Dependence Non-Uniformity Normalized (DNN) - Small Dependence Emphasis (SDE) - Large Dependence Emphasis (LDE) - Small Dependence High Gray Level Emphasis (SDHGLE) - Small Dependence Low Gray Level Emphasis (SDLGLE) - Large Dependence Low Gray Level Emphasis (LDLGLE) - - Large Dependence High Gray Level Emphasis (LDHGLE)

	Gray Level Run Length Matrix (GLRLM)	- Short Run Emphasis (SRE) - Run Length Non-Uniformity (RLN) - Long Run Emphasis (LRE) - Run Length Non-Uniformity Normalized (RLNN) - Run Percentage (RP) - Gray Level Non-Uniformity (GLN) - Run Variance (RV) - Gray Level Non-Uniformity Normalized (GLNN) - Run Entropy (RE) - High Gray Level Run Emphasis (HGLRE) - Low Gray Level Run Emphasis (LGLRE) - Short Run Low Gray Level Emphasis (SRLGLE) - Short Run High Gray Level Emphasis (SRHGLE) - Long Run Low Gray Level Emphasis (LRLGLE) - Long Run High Gray Level Emphasis (LRHGLE)
	Gray Level Size Zone Matrix (GLSZM)	- Small Area Emphasis (SAE) - Zone Percentage (ZP) - Large Area Emphasis (LAE) - Zone Variance (ZV) - Gray Level Variance (GLV) - Zone Entropy (ZE) - Gray Level Non-Uniformity (GLN) - Gray Level Non-Uniformity Normalized (GLNN) - Size-Zone Non-Uniformity (SZN) - Size-Zone Non-Uniformity Normalized (SZNN) - Low Gray Level Zone Emphasis (LGLZE) - High Gray Level Zone Emphasis (HGLZE) - Small Area Low Gray Level Emphasis (SALGLE) - Small Area High Gray Level Emphasis (SAHGLE) - Large Area Low Gray Level Emphasis (LALGLE) - Large Area High Gray Level Emphasis (LAHGLE)
	Neighboring Gray Tone Difference Matrix (NGTDM)	- Coarseness - Complexity - Contrast - Strength - Busyness
Higher-Order Statistics Features		- Wavelet or Fourier transforms - Fractal analysis - Minkowski functionals - Laplacian transform of Gaussian-filtered images (Laplacian of Gaussian)

Table S2. Supervised Feature Selection Methods

Method	Description	Example techniques
Filter methods (univariate methods)	Assess the relationship between individual features statistically, without accounting for their interactions or correlations.	Chi-squared test, Student’s t-test, Wilcoxon rank sum test, and Fisher score.
Wrapper methods (multivariate methods)	Overcome some of the limitations of univariate approaches by considering the interactions and correlations among features. Instead of examining features in isolation, wrapper methods create and evaluate subsets of features by applying them to a predictive model. This process is repeated iteratively to find the best-performing feature set. Due to their iterative nature, wrapper methods are computationally demanding.	Bidirectional search, exhaustive feature selection, forward selection, and backward elimination.
Embedded methods	Integrate feature selection into the model training phase, combining the benefits of both filter and wrapper approaches. By accounting for feature interactions and correlations during training, embedded methods achieve more precise feature selection compared to filter methods. Moreover, these methods are generally faster than wrapper methods and less prone to overfitting.	Tree-based algorithms like the random forest classifier, LASSO (least absolute shrinkage and selection operator), and ridge regression.

Figure S1. Principal Component Analysis (PCA) Classifier

Technique for dimensionality reduction & data visualization. In high dimensional data, features are often correlated, so they don't all provide unique information. This is where PCA comes in, it reduces dimensionality while retaining as much variability as possible.

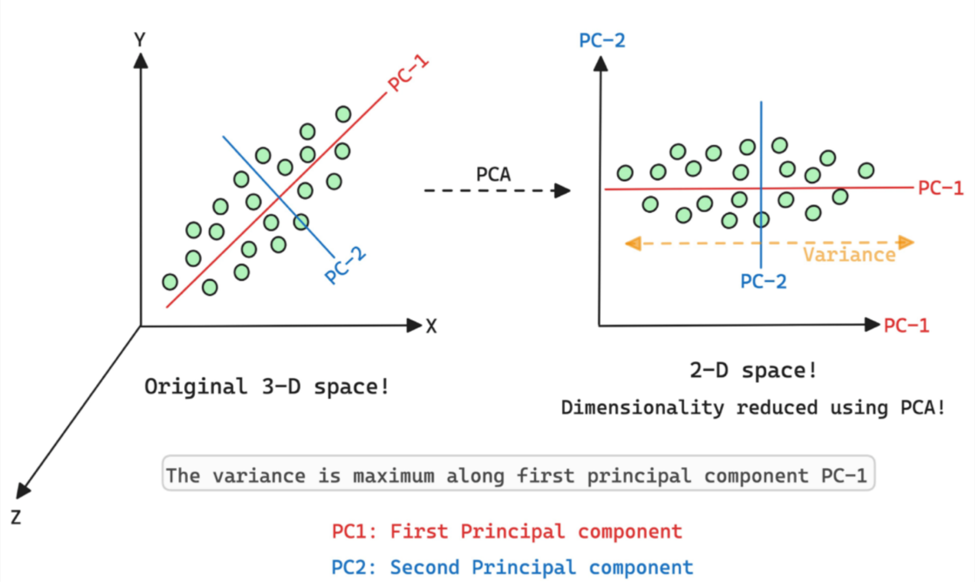
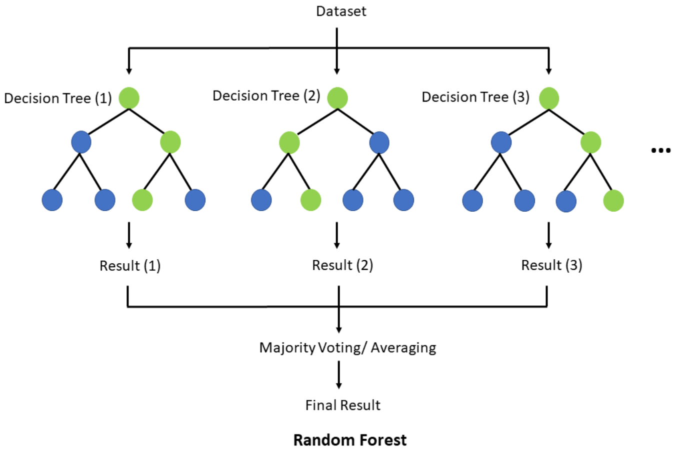

Y
PC-1
PC-2
X
Z
Original 3-D space!
PCA
PC-2
PC-1
Variance
PC-2
PC-1
2-D space!
Dimensionality reduced using PCA!
The variance is maximum along first principal component PC-1
PC1: First Principal component
PC2: Second Principal component

Figure Origin: "Pachaar, Akshay. (2023, October 28). [Image]. X. https://x.com/akshay_pachaar/status/1717519050706952695/photo/1"

On the left, it shows an original 3D space with data points scattered across three axes (X, Y, Z). The PCA process identifies the directions of maximum variance within this data, represented by the principal components: PC-1 (in red) and PC-2 (in blue). PC-1 captures the highest variance in the data, and PC-2 captures the next highest variance, orthogonal to PC-1. On the right side, the data has been transformed into a reduced 2D space using PCA. In this new space, PC-1 and PC-2 are the axes. The data is now represented in two dimensions, with most of the variance captured along PC-1, showing how PCA helps in simplifying the dataset while retaining the most important information. The dashed orange line emphasizes the variance captured by PC-1 in this reduced space.

Figure S2. Random Forest Classifier



"Random forest explain" available on Wikimedia Commons.

A Random Forest is an ensemble composed of decision trees combined using a method called bagging. This method involves running simple algorithms in parallel. The parallel use of simple algorithms leverages their independence, so the model's output is an average of all individual outputs. The independence of the different trees is achieved by providing each with different training data, ensuring that none of them are trained on the same data. The training dataset is fed into the different decision trees, resulting in a common output averaged from the individual results of all the trees:

Figure S3. Logistic Regression Classifier

Logistic Regression is method used for binary classification tasks, it is possible to implement it on multi-class tasks by dividing the problem in multiple binary classifiers of one vs the rest or OvR. The logistic regression model works by transforming a linear combination of input features using a *logit function* (left plot), which is the natural logarithm of the odds of the probability P of an event occurring:

$$P(Y) = \frac{1}{1 + e^{-z}}$$

where:

- P(Y) is the probability of the target (dependent) variable.
- e is the base of the natural logarithm.
- z is the linear combination of input features and their corresponding weights: $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... +$ βnxn.

The logit function maps any real-valued number to the range [0, 1], which is crucial for interpreting the output as a probability, and thus, assign each element to a given class based on the ratio of probabilities. Then, once the model is trained, we can use the inverse *logistic or sigmoid function* (right plot) to predict the probability that a certain sample belongs to a particular class given the input features.

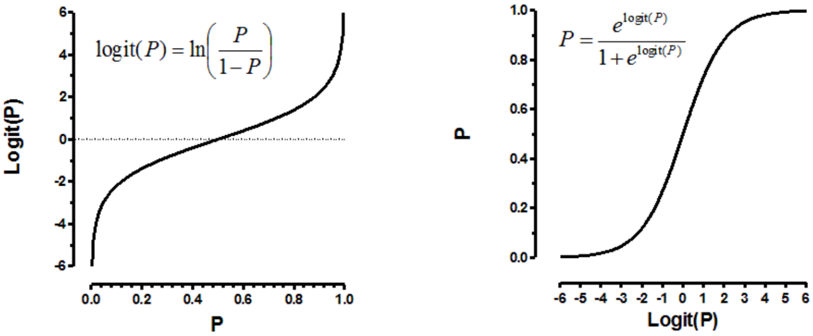


Figure Origin: "Wikipedia contributors. (n.d.). Logit. In Wikipedia, The Free Encyclopedia. Retrieved [10 June 2024], from https://en.wikipedia.org/wiki/Logit"

Table S3. Hyperparameters To Optimize for Maximizing Model Performance Random Forest.

Hyperparameter	Description	Recommendation
n_estimators	This hyperparameter determines the number of decision trees included in the forest. A higher number of trees generally improves model accuracy but can also increase training and execution time.	It is recommended to start with a low value (e.g., 100) and gradually increase it until the optimal value is found.
max_depth	This hyperparameter controls the complexity of each decision tree. A higher value allows the model to capture more complex patterns in the data but can also increase the risk of overfitting.	It is recommended to start with a low value (e.g., 5) and gradually increase it until the optimal value is found.
min_samples_split	This hyperparameter indicates the minimum number of samples required to consider splitting a node in a decision tree. A higher value can improve model accuracy but can also increase training and execution time.	It is recommended to start with a low value (e.g., 2) and gradually increase it until the optimal value is found.
min_samples_leaf	This hyperparameter indicates the minimum number of samples that must be present in each leaf of a decision tree. A higher value can improve model accuracy but can also increase the risk of overfitting.	It is recommended to start with a low value (e.g., 1) and gradually increase it until the optimal value is found.

Table S4. Model Performance Metrics: Concepts and Formulas

i. Accuracy: Is a metric that evaluates the proportion of correct predictions made by a model in relation to the total number of predictions made.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + False\ Positives + False\ Negatives + True\ Negatives}$$

This metric provides an overall view of the model's effectiveness in correctly classifying instances. However, it is important to note that accuracy can be misleading in situations where classes are imbalanced, as a model could achieve high accuracy simply by always predicting the majority class.

ii. Precision: Is a metric that evaluates the proportion of correct positive predictions in relation to the total positive predictions made by the model. In other words, precision measures the model's ability to avoid making incorrect positive predictions. A high precision value indicates that the model is less likely to incorrectly classify negative instances as positive.

$$Precision = \frac{True\ Positives}{True\ Postitives + False\ Positives}$$

Precision is especially relevant in situations where false positives have a significant impact or when minimizing classification errors in the positive class is crucial.

iii. Sensitivity: Also known as recall, is understood as the true positive rate. In this case, it measures the ability to correctly identify individuals who have a negative response to treatment.

$$Sensitivity = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Is interpreted as the proportion of positive instances that the model correctly identifies out of all the actual positive instances in the dataset. A high recall value indicates that the model is effective at capturing most of the positive instances, even if there might be an increase in false positives.

iv. Specificity: Is the ability of the model to correctly detect cases where the tumor has effectively responded to treatment, categorized as 0.

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

It measures the proportion of negative instances that the model correctly identifies out of all the actual negative instances in the dataset. A high specificity value indicates that the model is effective at correctly identifying most of the negative instances, minimizing the number of false positives.

v. F1 Score: Is a metric that combines precision and recall into a single value, providing a more balanced measure of a classification model's performance. This metric is especially useful when there is an imbalance between classes or when a balance between precision and the ability to identify positives is desired.

$$F1 = 2x\frac{Accuracy\ x\ Sensibility}{Accuracy + Sensibility}$$

The F1 score ranges from 0 to 1, where a higher value indicates better performance. It seeks a balance between precision and recall, as both components contribute to the final score. If both precision and recall are high, the F1 score will be close to 1, indicating strong performance in the model's ability to correctly classify both classes.

vi. Receiver Operating Characteristic (ROC) curves: Are graphical tools used to evaluate and compare the performance of classification models at different discrimination thresholds. These curves represent the true positive rate (sensitivity) versus the false positive rate (1 - specificity) for various threshold values. In the graphical representation of a ROC curve, the X-axis shows the false positive rate, while the Y-axis shows the true positive rate.

The area under the ROC curve (AUC-ROC) provides a numerical measure of the model's performance. An AUC-ROC close to 1 indicates good model performance, while a value close to 0.5 suggests performance similar to random chance. ROC curves and AUC-ROC are useful tools for comparing several models and selecting the most suitable one for the specific classification task

vii. Confusion Matrix: It provides a detailed view of the model's performance by comparing predictions with the actual class values. The matrix provides a detailed view of the relationship between predicted and actual labels, including true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). This information is vital for understanding under which specific conditions a model may fail and allows for adjusting parameters or modeling strategies to improve performance.

- True Positives (TP): Cases where the model correctly predicted the positive class.
- True Negatives (TN): Cases where the model correctly predicted the negative class.
- False Positives (FP): Cases where the model incorrectly predicted the positive class.
- False Negatives (FN): Cases where the model incorrectly predicted the negative class.

Representation of a Confusion Matrix:

	True Prediction	Negative Prediction	
True Condition	TN	FP Type I error	Specificity TN/(TN+FP)
Negative Condition	FN Type II error	TP	Sensibility (recall) TP/(TP+FN)
	Negative Rate TN/(FN+TN)	Precision TP/(FP+TP)	Accuracy (TN+TP)/(TN+FP+FN+TP)

Figure S4. PCA Scree Plot

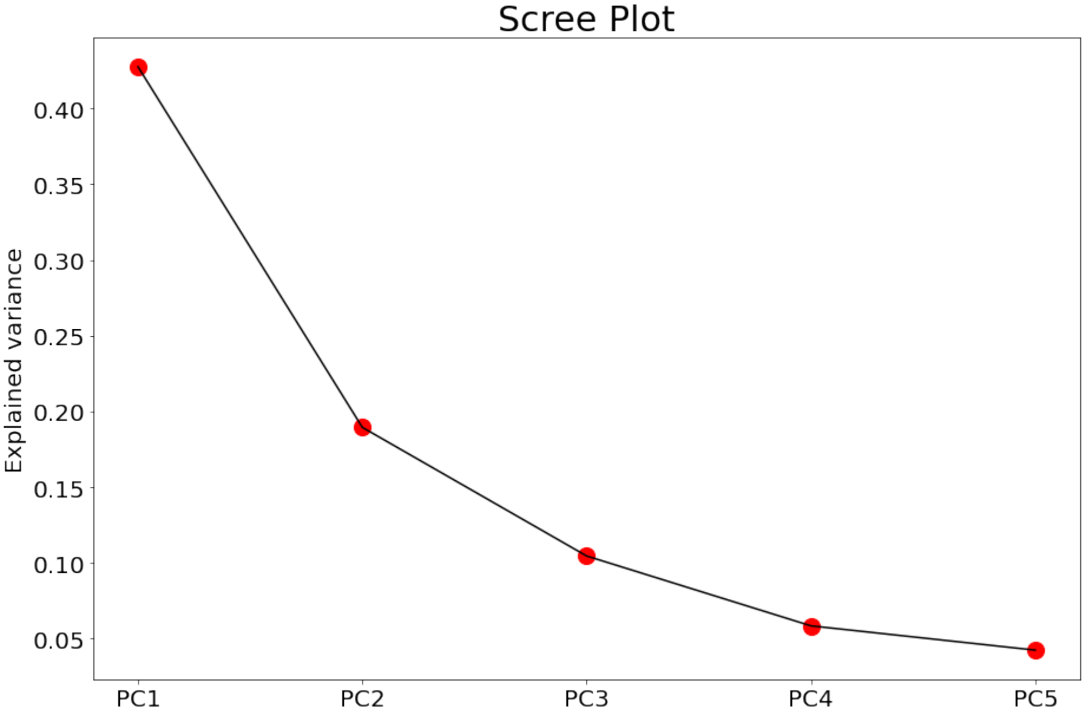


Figure S5. PCA Feature Selection according to the absolute value of their contribution

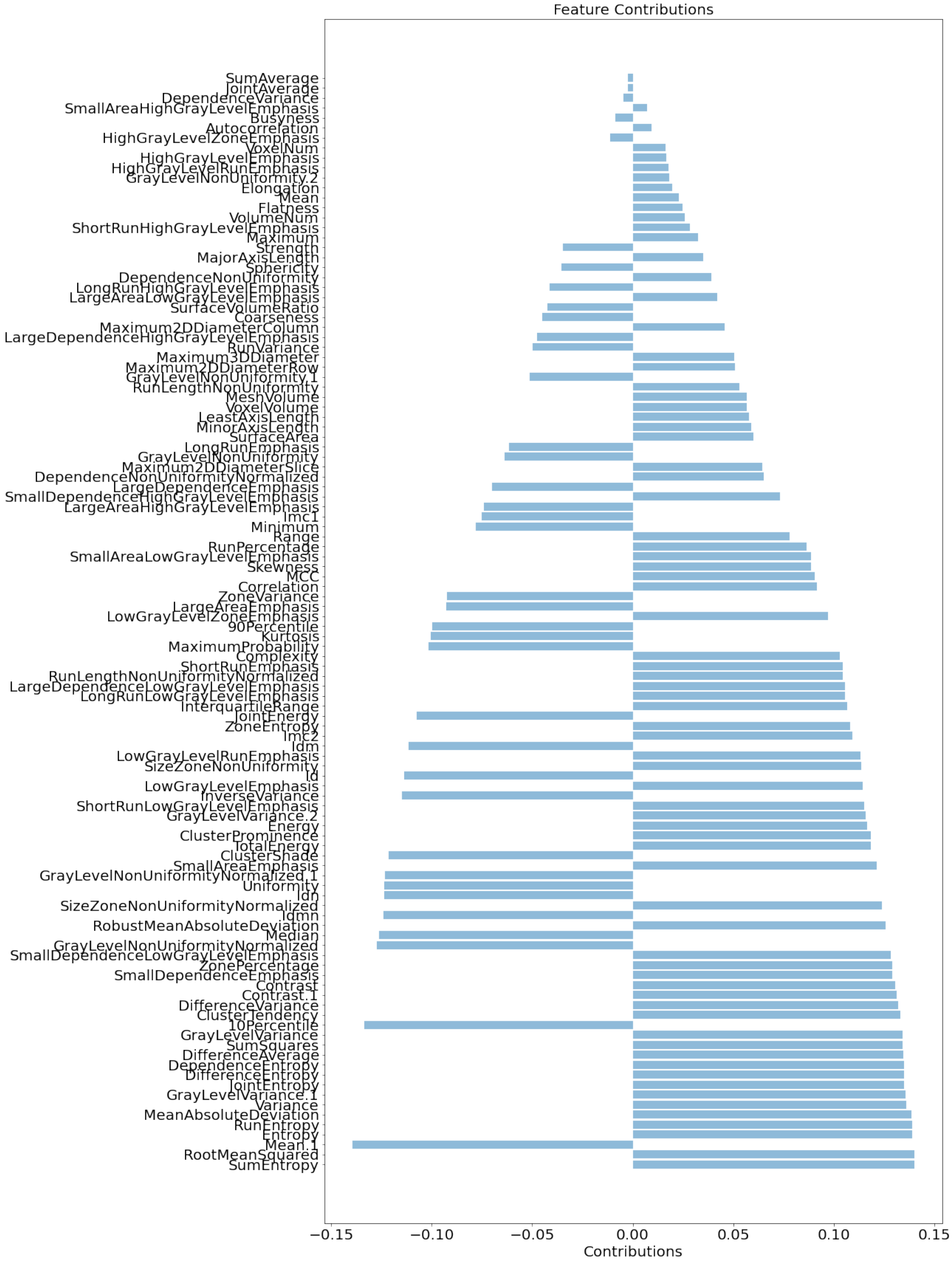


Figure S6. Logistic Regression Feature Selection ordered by their Coefficients

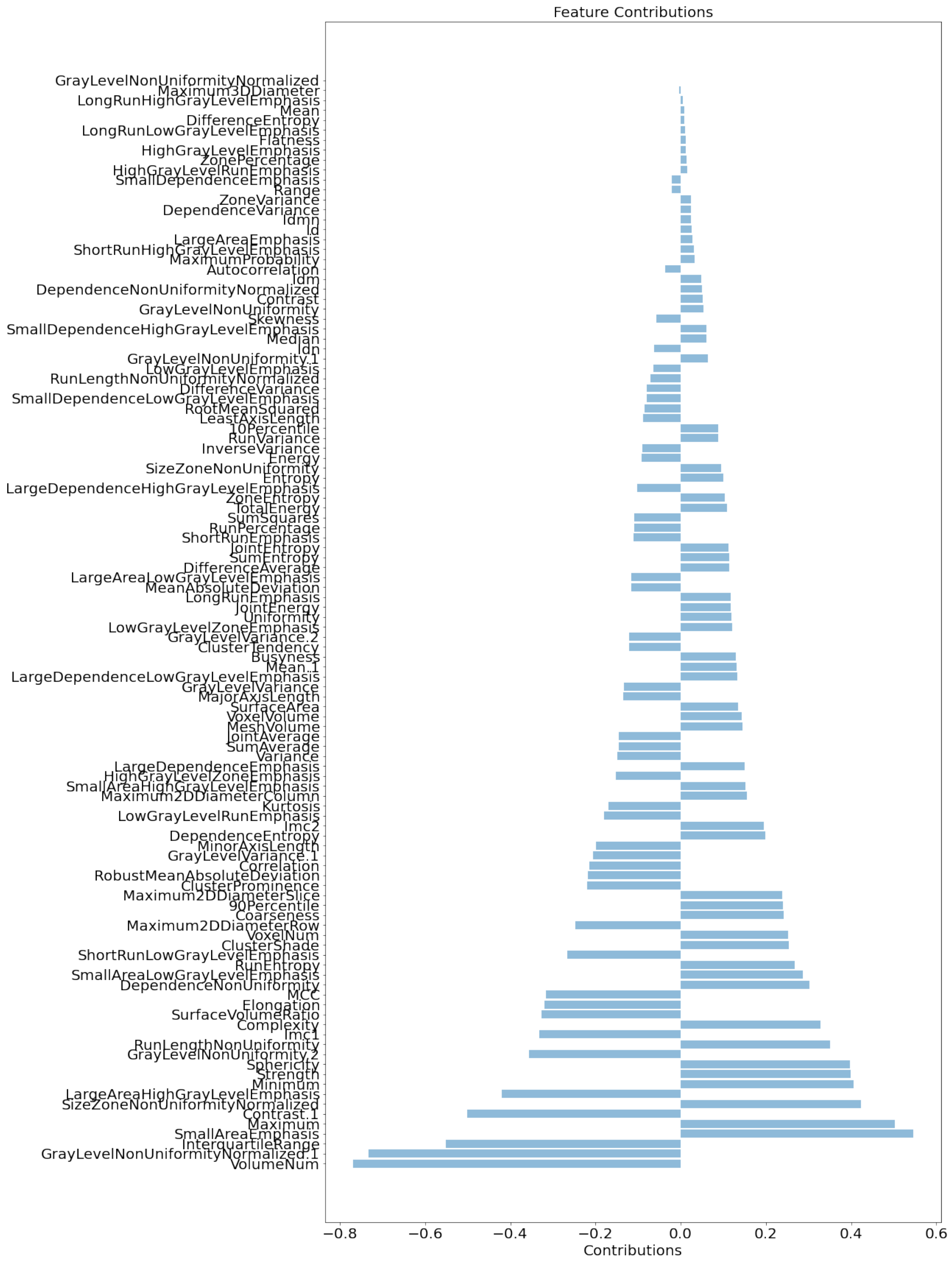


Figure S7. Random Forest Feature Selection ordered by their Importance

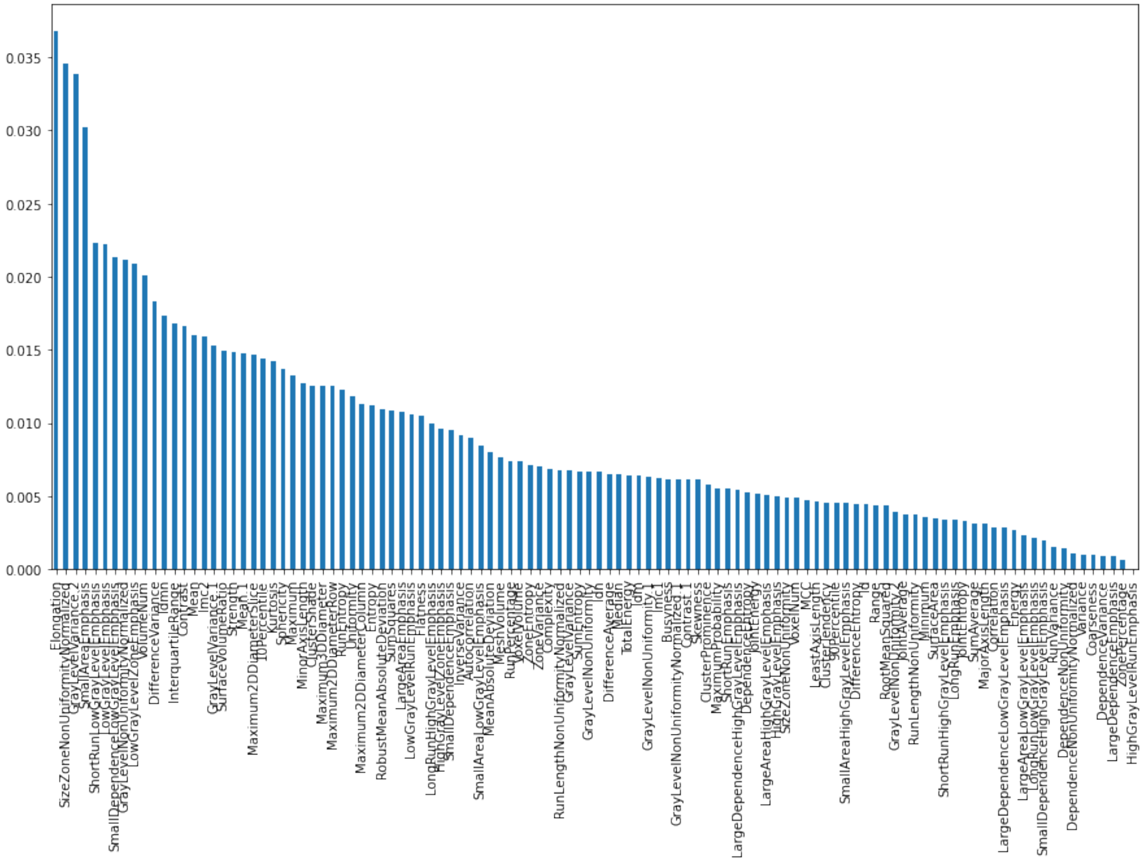


Table S5. Summary of our study and previous radiomics studies predicting pCR in rectal cancer based on simulation CT-scan imaging

Literature	Modality	Patient numbers	Features	Radiomics signature	AUC
Yuan et al[29]	Non-contrast enhanced simulation CT	91	Radiomics	RF	0.839 (Accuracy)
Lutsyk et al[30]	Non-contrast enhanced simulation CT	140	Radiomics wavelets	RF (no Radscore construction)	0.872
Bonomo et al[31]	Contrast-enhanced simulation CT	201	Radiomics	KNN	0.63
Bibault et al[12]	Contrast-enhanced simulation CT	95	Radiomics	DNN (no Radscore construction)	0.80 (Accuracy)
Zhuang et al[32]	Contrast-enhanced simulation CT	177	Radiomics + Clinical	LASSO	0.822
Wang et al[33]	Non-contrast enhanced simulation CT	217	Radiomics Texture + DVH + Clinical	RF (no Radscore construction)	0.828
Mao et al[34]	Contrast-enhanced CT	216	Radiomics + Clinical	LASSO	0.872
Li et al[35]	Contrast-enhanced simulation CT	211	Radiomics + Clinical	LASSO, RF, SVM	0.866
Our study	Non-contrast enhanced simulation CT	49	Radiomics + Clinicopathological	RR, NN	0.945

CT: computed tomography; LASSO: least absolute shrinkage and selection operator; RF: random forest; SVM: support vector machines; KNN: k-nearest neighbor; DNN: deep neural network