Radiomic Prediction of Tumor Grade and Overall Survival from the BraTS Glioma Dataset: An Exploratory Analysis of Dimensionality Reduction Techniques and Machine Learning Classifiers

Kareem Wahid
Colen Lab, MD Anderson Cancer Center

Supplementary A: Dataset

The Multimodal Brain Tumor Segmentation (BraTS) Challenge is an annual competition that seeks to employ the brightest minds in computational radiology to develop the following:

- Accurate automatic segmentation algorithms of gliomas in magnetic resonance images (MRI).
- Prediction of patient overall survival through radiomics and machine learning algorithms.

The 2017 BraTS Challenge provides a training dataset with 75 LGG and 210 GBM cases. This includes T1, T1ce, T2, and FLAIR pre-operative MRI scans for each case; scans are from multiple different institutions. The ground truth labels of segmentations from different phenotypes (sub-regions), i.e. necrotic/non-enhancing tumor (NCR), peritumoral edema (ED), and Gd-enhancing tumor (ET) are provided. NCR is usually hypo-intense when viewed on T1ce compared to T1. ED has a hyper-intense signal when viewed on FLAIR. ET has a hyper-intense signal on T1ce when compared to T1. Expert neuroradiologists have radiologically assessed the complete original TCIA glioma LGG and HGG collections and categorized each scan as pre-or post-operative. Subsequently, all the pre-operative TCIA scans were annotated by experts. Included in the dataset were select cases released in previous BraTS Challenges. A subset of the data was then used as the training set released on May 5, 2017. Validation and testing sets were released at later dates but these did not have labels (for segmentations or overall survival) so we did not consider them in the analysis. MRI scans and corresponding segmentations are provided in the form of .nii files. The provided data are distributed after their pre-processing, i.e. co-registered to the same anatomical template, interpolated to the same resolution (1 mm^3) and skull-stripped. An example visualization of a GBM segmentation superimposed on a FLAIR scan is shown in Fig. SA. In addition, 163 GBM cases include survival data and age in a .csv file. This leads us to believe 2 classification prediction tasks of interest are possible with this dataset, tumor grade prediction and overall survival prediction. More information on the dataset is available at http://braintumorsegmentation.org/ [1].

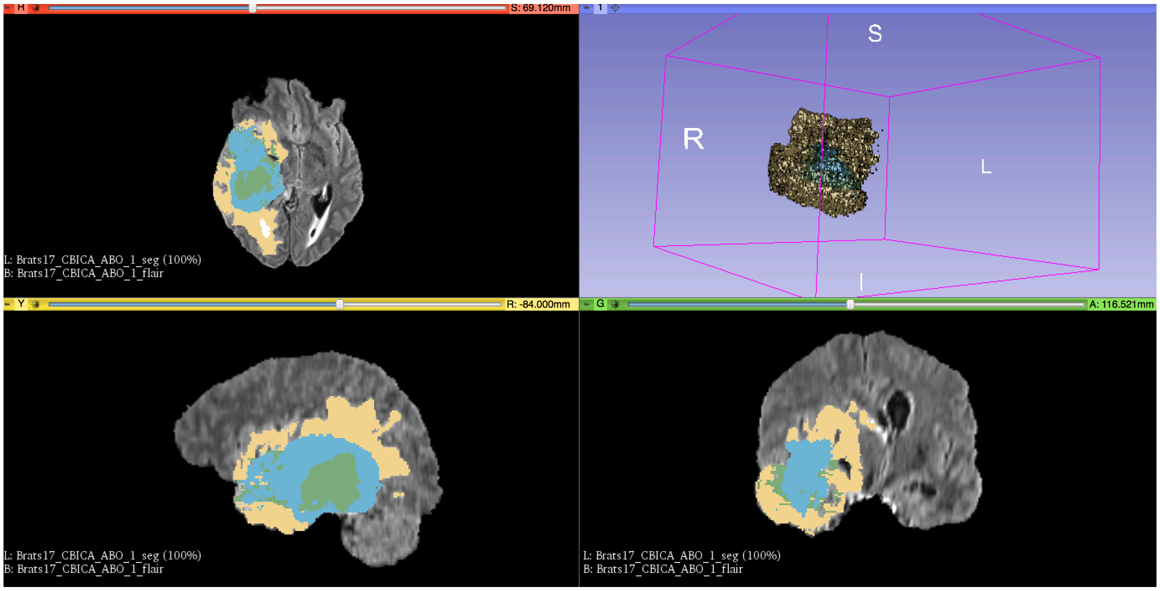

Figure SA. Visualization of GBM FLAIR MRI scan with corresponding tumor segmentation obtained from BraTS training dataset. Tumor segmentation represented by colored areas. Color-coding represents subregions of tumor (blue = ET, green = NCR, yellow = ED). Tumor segmentation 3D volume is visualized in top right corner. Visualization performed in 3D Slicer [2].

Supplementary B: Radiomic Features

All radiomic features were extracted via the Pyradiomics open source toolbox. We chose to utilize all the available features for every feature type. In addition, we applied a coiflet wavelet transform filter to images and calculated all intensity-based features (first-order, GLCM, GLSZM, GLRLM). This led to the creation of 718 features for each phenotype (ET, ED, NCR), and subsequently 2154 features for each image. Though many features would be redundant, our expectation was that dimensionality reduction methods would emphasize the most relevant features for their given outcome. More information on Pyradiomics can be found in the documentation at http://pyradiomics.readthedocs.io/en/latest/index.html [3].

Shape: Shape features describe the physical measurements of the tumor segmentation and are independent of intensity values in the image. The following shape features were utilized for our extraction: Maximum3DDiameter, Compactness2, Maximum2DDiameterSlice, Sphericity, MinorAxis, Compactness1, Elongation, SurfaceVolumeRatio, Volume, SphericalDisproportion, MajorAxis, LeastAxis, Flatness, SurfaceArea, Maximum2DDiameterColumn, Maximum2DDiameterRow.

First-order statistics: First-order statistics features describe the intensity of gray values in the region of interest (ROI). The following intensity features were utilized: InterquartileRange,Skewness, Uniformity, MeanAbsoluteDeviation, Energy, RobustMeanAbsoluteDeviation, Median, TotalEnergy, Maximum, RootMeanSquared, 90Percentile, Minimum, Entropy, StandardDeviation, Range, Variance, 10Percentile, Kurtosis, Mean.

Gray level co-occurrence matrix (GLCM): GLCM features describe mathematical relationships between co-occuring voxel intensity values in an image ROI. The following GLCM features were utilized: SumVariance, Homogeneity1, Homogeneity2, ClusterShade, MaximumProbability, Idmn, Contrast, DifferenceEntropy, InverseVariance, Dissimilarity, SumAverage, DifferenceVariance, Idn, Idm, Correlation, Autocorrelation, SumEntropy, AverageIntensity, Energy, SumSquares, ClusterProminence, Entropy, Imc2, Imc1, DifferenceAverage, Id, ClusterTendency.

Gray Level Size Zone Matrix (GLSZM): GLZSM features describe then number of connected voxels that share the same gray level intensity. The following GLSZM features were utilized: GrayLevelVariance, SmallAreaHighGrayLevelEmphasis, GrayLevelNonUniformityNormalized, SizeZoneNonUniformityNormalized, SizeZoneNonUniformity, GrayLevelNonUniformity, LargeAreaEmphasis, ZoneVariance, ZonePercentage, LargeAreaLowGrayLevelEmphasis, LargeAreaHighGrayLevelEmphasis, HighGrayLevelZoneEmphasis, SmallAreaEmphasis, LowGrayLevelZoneEmphasis, ZoneEntropy, SmallAreaLowGrayLevelEmphasis.

Gray Level Run Length Matrix (GLRLM): GLZSM features describe length of consecutive voxels that have the same gray level value. The following shape features were utilized: ShortRunLowGrayLevelEmphasis, GrayLevelVariance, LowGrayLevelRunEmphasis, GrayLevelNonUniformityNormalized, RunVariance, GrayLevelNonUniformity, LongRunEmphasis, ShortRunHighGrayLevelEmphasis, RunLengthNonUniformity, ShortRunEmphasis, LongRunHighGrayLevelEmphasis, RunPercentage, LongRunLowGrayLevelEmphasis, RunEntropy, HighGrayLevelRunEmphasis, RunLengthNonUniformityNormalized

Wavelet: Wavelet transforms decouple textural information by decomposing the image into different frequency domains [4]. We applied a discrete, one-level, undecimated 3 dimensional coiflet wavelet transform to each image:mask combination leading to 8 decompositions (LLL, LLH, LHH, LHL, HLL, HHL, HLH, HHH; L = low pass filter, H = high pass filter) for each phenotype. For each decomposition, we computed first-order statistics and textural features (GLCM, GLSZM, GLRLM), as described earlier, for analysis. A coiflet wavelet was chosen over other methods (e.g. haar, dmey, sym, etc.) due to its popularity and positive results in past studies [4-6].

Supplementary C: Unsupervised Dimensionality Reduction Methods

We utilized the popular open source python machine learning library scikit-learn (sklearn) for all model building. Sklearn employs easy to use machine learning modules and libraries that are built on NumPy, SciPy, and matplotlib. More information on sklearn can be found at: http://scikit-learn.org/ [7]. All the analysis and corresponding code can be found at: GITHUB.

The following unsupervised dimensionality reduction methods were utilized in this study:

Principal Component Analysis (PCA): PCA decomposes a multivariate dataset into a set of orthogonal linearly uncorrelated components, termed “principal components”, that explain a maximum amount of the variance; the greatest variance is explained by the first principal component, followed by the second, and so on [8]. In other words, PCA generates new orthogonal variables that are linear combinations of the original variables. The sklearn PCA implementation utilizes Single Value Decomposition, i.e. eigendecomposition of m x n matrix via polar decomposition.

Kernel PCA (KPCA): KPCA generalizes PCA by implementing a kernel to be better suited for non-linear data. A kernel function ϕ for N points x_i can be defined as a mapping from a d-dimensional nonlinear feature space $\mathbb{R}^d$ into a higher N-dimensional feature space $\mathbb{R}^N$ which is linearly separable such that $\phi(x_i): \mathbb{R}^d \to \mathbb{R}^N$. In this version of PCA, we mathematically formulate a version of PCA in which we never solve the eigenvectors and eigenvalues of the covariance matrix in $\phi(x)$ space. KPCA does not compute principal components themselves, but projections of our data onto those components. In this study, we utilize a sigmoid kernel for KPCA.

Independent Component Analysis (ICA): ICA is closely related to PCA, which also utilizes linear combinations of the original variables to create new variables. Where PCA can be thought to maximize the second moment of the measured data (variance) and Gaussian priors are assumed, ICA instead maximizes higher moments, such as kurtosis, and exploits non-Gaussian features [9]. This results in “independent components” of the original data. Sklearn utilizes a FastICA algorithm in its ICA implementation.

Factor Analysis (FA): FA is a similar method to PCA, since both methods involve linear combinations of original variables to create new variables. However, FA seeks to reproduce inter-correlations among variables, where unobserved factors represent common variance excluding unique variance [10]. Moreover, FA emphasizes off-diagonal terms in a covariation matrix, where PCA emphasizes on-diagonal terms [8]. Results between PCA and FA are often similar, but depending on the underlying data they can give radically different results. Sklearn uses a simple linear generative model with Gaussian latent variables for its FA implementation.

Supplementary D: Classifier Methods

Classifiers were selected from sklearn with default hyperparameter settings.
The following supervised classifier methods were utilized in this study:

Decision Tree (DT): DTs are non-parametric methods that create models by using simple decision rules. DTs are by far some of the most interpretable methods (white-box model) for machine learning, since the decision processes closely mimic conditional statements. DTs were some of the most widely used machine learning methods in the past, but have fallen out of favor due to their relatively low predictive power compared to newer methods. Some radiomics studies still find they can perform relatively well though [11]. Sklearn utilizes the following default hyperparameter settings for DT classifiers:

`DecisionTreeClassifier`(*criterion='gini'*, *splitter='best'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features=None*, *random_state=None*, *max_leaf_nodes=None*, *min_impurity_split=1e-07*, *class_weight=None*, *presort=False*).

Random Forrest (RF): Ensemble techniques combine a set of "weak" learners to create a "strong" learner [12]. RFs build upon DTs in an ensemble method by fitting multiple DT classifiers on subsets of data through a bootstrap approach and averaging results to improve predictive performance and reduce over-fitting. Sklearn utilizes the following default hyperparameter settings for RF classifiers:
`RandomForestClassifier`(*n_estimators=10*, *criterion='gini'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features='auto'*, *max_leaf_nodes=None*, *min_impurity_split=1e-07*, *bootstrap=True*, *oob_score=False*, *n_jobs=1*, *random_state=None*, *verbose=0*, *warm_start=False*, *class_weight=None*).

Bagging (BAG): BAG is similar to RF. However, RF utilizes random subsets of features when splitting nodes, while BAG uses all the features. Sklearn utilizes thte following default hyperparameter settings for RF classifiers:
`BaggingClassifier`(*base_estimator=None*, *n_estimators=10*, *max_samples=1.0*, *max_features=1.0*, *bootstrap=True*, *bootstrap_features=False*, *oob_score=False*, *warm_start=False*, *n_jobs=1*, *random_state=None*, *verbose=0*).

Boosting (BST): Boosting is similar to RF and BAG in that it is an ensemble technique built off of DT. However, BST uses all the data to train each learner and "boosts" performance by giving more weight to instances that were misclassified by previous learners through a loss function. Sklearn utilizes the following default hyperparameter settings for BST classifiers:
`GradientBoostingClassifier`(*loss='deviance'*, *learning_rate=0.1*, *n_estimators=100*, *subsample=1.0*, *criterion='friedman_mse'*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_depth=3*, *min_impurity_split=1e-07*, *init=None*, *random_state=None*, *max_features=None*, *verbose=0*, *max_leaf_nodes=None*, *warm_start=False*, *presort='auto'*).

Naïve Bayes (NB): Given a class variable y and dependent feature vector x_1 through x_n, Bayes' theorem states [8]:

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)}$$

NB classifiers take advantage of Bayes' theorem with the assumption of independence between pairs of features. Of course, in the real world, features are often correlated. Despite this, NB algorithms still work exceedingly well for many datasets. Additionally, NB algorithms are computationally very cheap. In this study we utilize a Gaussian NB algorithm which assumes the likelihood of features is distributed as a Gaussian. Sklearn utilizes the following default hyperparameter settings for Gaussian NB classifiers: `GaussianNB`(*priors=None*).

Multi-Layer Perceptron (MLP): A perceptron is a simple implementation of a neural network with a feed-forward architecture. Perceptrons crudely model the way a biological neuron conducts signals by employing activation functions that allow an artificial neuron to fire if a threshold value is crossed. Input vectors are multiplied with weights in an attempt to correctly classify datapoints. If a point is misclassified, weights are adjusted until the point is classified correctly. We outline a perceptron mathematically as follows. Consider a binary classification task where we refer to the 1 as our positive class and 0 as our negative class. We will define an activation function $\phi(z)$ as a function that takes a linear combination of input values **x** (feature vector) and a weight vector **w**, where z is the input [13]:

$$z = w_1 x_1 + \cdots + w_m x_m = \sum_{j=1}^{m} w_j x_j = \boldsymbol{w}^T \cdot \boldsymbol{x}$$

$$\phi(z) = \begin{cases} 1, & if\ z \geq \theta \\ 0, & if\ z \geq \theta \end{cases}$$

here θ is an activation threshold. We then define a perceptron learning rule that updates weights **w** so that all samples are classified correctly given enough iterations. Our perceptron can now correctly classify all samples. MLPs simulate many chain-wise activation functions through the implementation of hidden layers. MLPs can be seen as progenitors to todays state of the art "deep" neural networks. In sklearn the log-loss function is optimized using stochastic gradient descent. Sklearn utilizes the following default hyperparameter settings for MLP classifiers: `MLPClassifier`(*hidden_layer_sizes=(100,), activation='relu', solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08*).

Logistic Regression (LR): LR is a technique native to statistics that is heavily implemented in machine learning applications [8]. LR uses a logistic function, such as the sigmoid, to output the probability of a data point belonging to a certain class. Sklearn utilizes the following default hyperparameter settings for LR classifiers: `LogisticRegression`(*penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='liblinear', max_iter=100, multi_class='ovr', verbose=0, warm_start=False, n_jobs=1*).

Support Vector Machine (SVM): SVMs work on the principle of creating a decision boundary between classes by maximizing the separation distance ("margin") between the data points of different classes [14]. The main advantage of support vector machines are that they are very effective in high dimensional spaces, are memory efficient, and can be

used to separate non-linear data with the kernel trick. Though support vector machines can work well in high dimensions, it is possible if the number of features are much larger than the number of training points performance will be poor. Sklearn utilizes the following default hyperparameter settings for SVM classifiers:
`SVC`(*C=1.0*, *kernel='rbf'*, *degree=3*, *gamma='auto'*, *coef0=0.0*, *shrinking=True*, *probability=False*, *tol=0.001*, *cache_size=200*, *class_weight=None*, *verbose=False*, *max_iter=-1*, *decision_function_shape=None*, *random_state=None*).

K-nearest Neighbor (KNN): KNN works on the principal of picking a set number of samples closest in distance to a new data point to classify that point [8]. Major advantages and disadvantages arise from the fact that KNN is a lazy learner, i.e., generalization is delayed until a query is made. Sklearn utilizes the following default hyperparameter settings for KNN classifiers:
`KNeighborsClassifier`(*n_neighbors=5*, *weights='uniform'*, *algorithm='auto'*, *leaf_size=30*, *p=2*, *metric='minkowski'*, *metric_params=None*, *n_jobs=1*, ***kwargs*).

Supplementary Figures/Tables

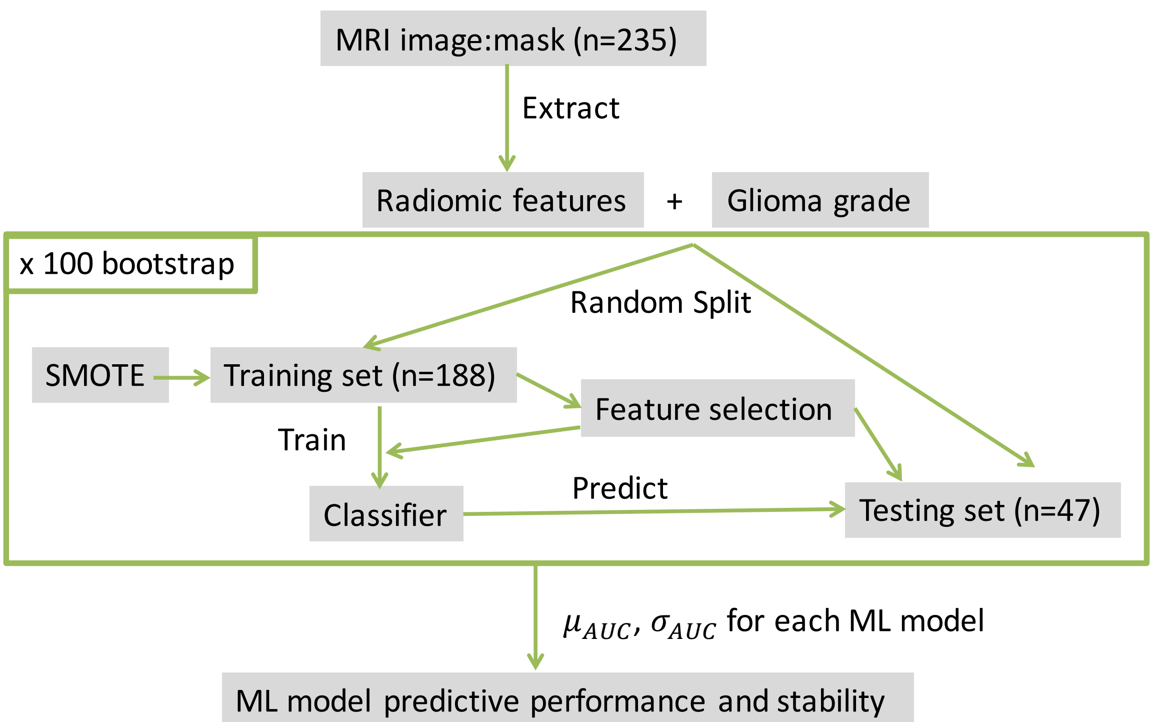

Figure S1. Workflow for Classification task. Survival task is identical with exception of SMOTE upsampling and glioma grade is replaced with survival class.

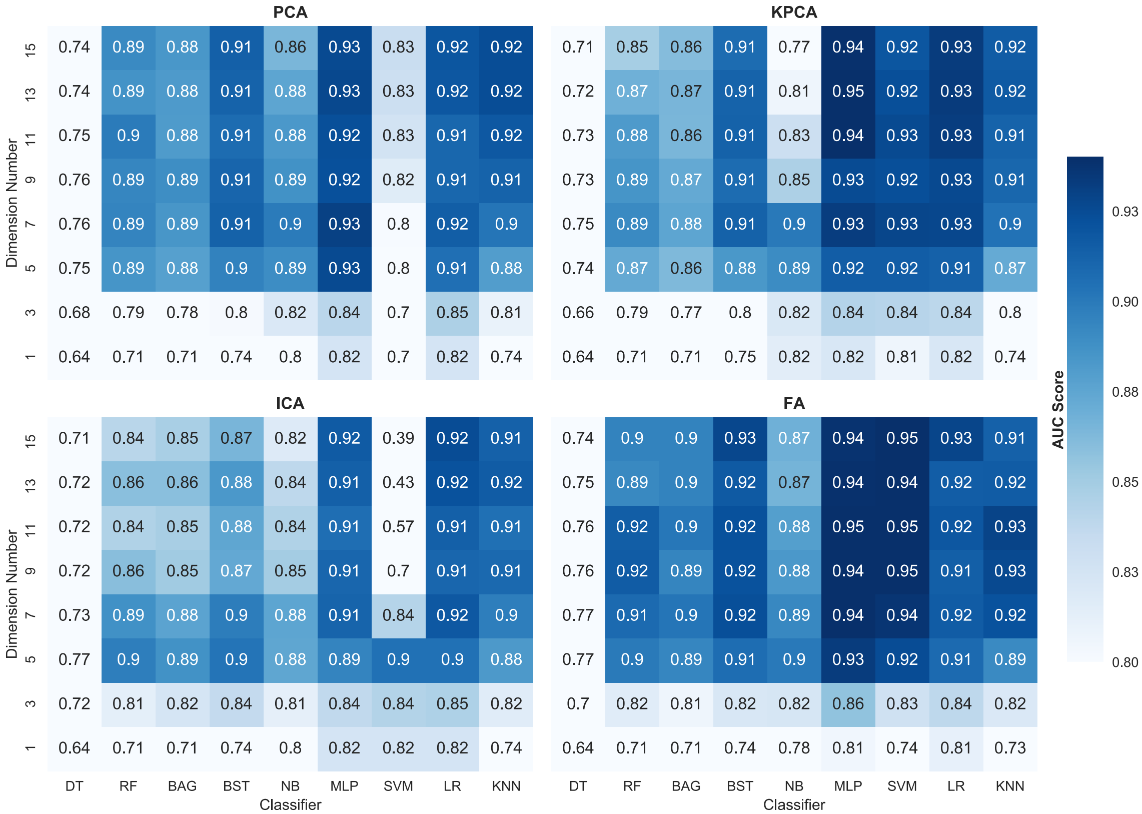

Figure S2. Predictive performance corresponding to classification methods (in columns) and the number of dimensions (in rows) for each dimensionality reduction method for grade classification task.

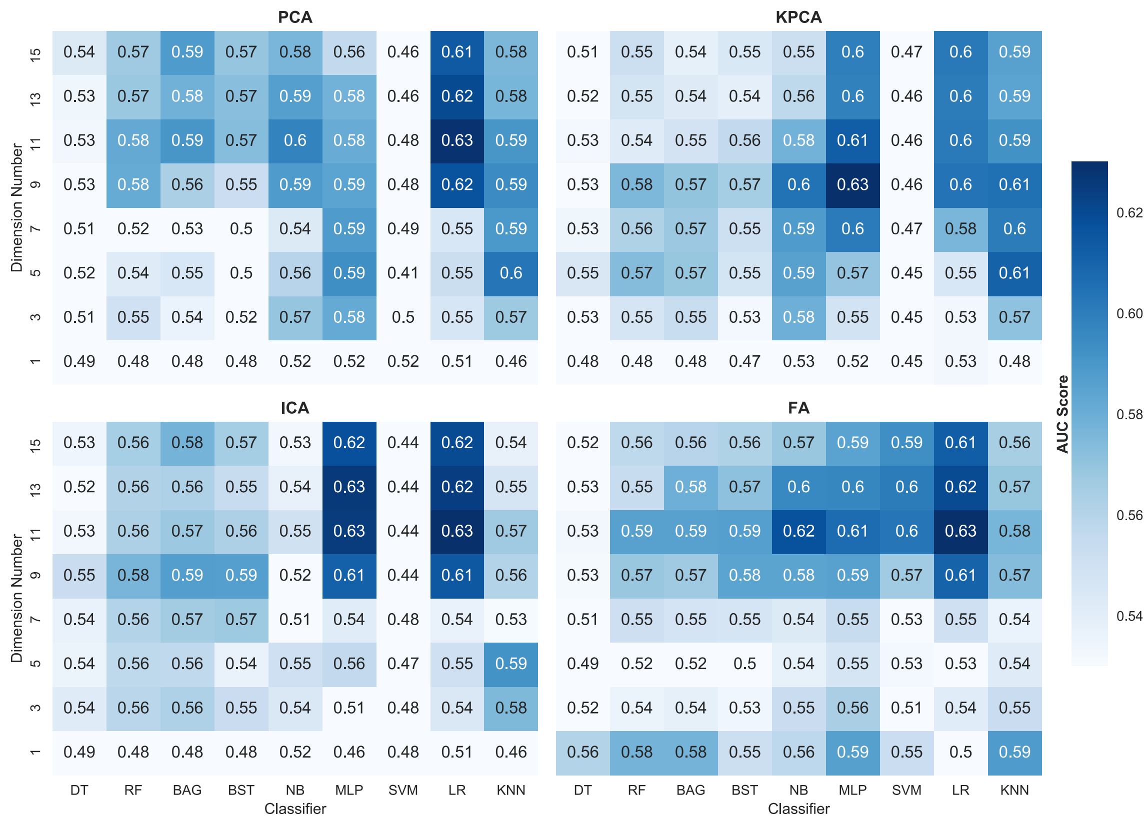

Figure S3. Predictive performance corresponding to classification methods (in columns) and the number of dimensions (in rows) for each dimensionality reduction method for survival classification task.

Table S1. Comparison of scoring function results for grade classification task. AUC is compared to accuracy and sensitivity.

Classifier	Selection Method	Dimension Number	μ_{AUC}	$\mu_{Accuracy}$	$\mu_{Sensitivity}$
RandomForestClassifier	FILT	1	0.93	0.90	0.92
RandomForestClassifier	PCA	1	0.71	0.65	0.64
RandomForestClassifier	PCA	3	0.79	0.75	0.76
RandomForestClassifier	PCA	5	0.89	0.84	0.87
RandomForestClassifier	PCA	7	0.89	0.84	0.87
RandomForestClassifier	PCA	9	0.89	0.86	0.88
RandomForestClassifier	PCA	11	0.90	0.86	0.89
RandomForestClassifier	PCA	13	0.89	0.85	0.89
RandomForestClassifier	PCA	15	0.89	0.85	0.89
RandomForestClassifier	KPCA	1	0.71	0.65	0.64
RandomForestClassifier	KPCA	3	0.79	0.75	0.76
RandomForestClassifier	KPCA	5	0.87	0.83	0.86
RandomForestClassifier	KPCA	7	0.89	0.85	0.88
RandomForestClassifier	KPCA	9	0.89	0.85	0.89
RandomForestClassifier	KPCA	11	0.88	0.85	0.90
RandomForestClassifier	KPCA	13	0.87	0.85	0.89
RandomForestClassifier	KPCA	15	0.85	0.84	0.89
RandomForestClassifier	ICA	1	0.71	0.65	0.64
RandomForestClassifier	ICA	3	0.81	0.76	0.77
RandomForestClassifier	ICA	5	0.90	0.85	0.88
RandomForestClassifier	ICA	7	0.89	0.85	0.88
RandomForestClassifier	ICA	9	0.86	0.83	0.88
RandomForestClassifier	ICA	11	0.84	0.83	0.89
RandomForestClassifier	ICA	13	0.86	0.85	0.91
RandomForestClassifier	ICA	15	0.84	0.84	0.92
RandomForestClassifier	FA	1	0.71	0.65	0.64
RandomForestClassifier	FA	3	0.82	0.76	0.77
RandomForestClassifier	FA	5	0.90	0.84	0.86
RandomForestClassifier	FA	7	0.91	0.86	0.89
RandomForestClassifier	FA	9	0.92	0.87	0.89
RandomForestClassifier	FA	11	0.92	0.86	0.89
RandomForestClassifier	FA	13	0.89	0.85	0.89
RandomForestClassifier	FA	15	0.90	0.86	0.89
GaussianNB	FILT	1	0.86	0.80	0.81

GaussianNB	PCA	1	0.80	0.67	0.65
GaussianNB	PCA	3	0.82	0.72	0.70
GaussianNB	PCA	5	0.89	0.81	0.81
GaussianNB	PCA	7	0.90	0.83	0.85
GaussianNB	PCA	9	0.89	0.84	0.86
GaussianNB	PCA	11	0.88	0.85	0.88
GaussianNB	PCA	13	0.88	0.85	0.88
GaussianNB	PCA	15	0.86	0.85	0.88
GaussianNB	KPCA	1	0.82	0.69	0.67
GaussianNB	KPCA	3	0.82	0.73	0.73
GaussianNB	KPCA	5	0.89	0.83	0.84
GaussianNB	KPCA	7	0.90	0.87	0.89
GaussianNB	KPCA	9	0.85	0.85	0.89
GaussianNB	KPCA	11	0.83	0.85	0.91
GaussianNB	KPCA	13	0.81	0.85	0.92
GaussianNB	KPCA	15	0.77	0.84	0.92
GaussianNB	ICA	1	0.80	0.67	0.65
GaussianNB	ICA	3	0.81	0.73	0.72
GaussianNB	ICA	5	0.88	0.82	0.84
GaussianNB	ICA	7	0.88	0.83	0.85
GaussianNB	ICA	9	0.85	0.83	0.85
GaussianNB	ICA	11	0.84	0.86	0.91
GaussianNB	ICA	13	0.84	0.87	0.93
GaussianNB	ICA	15	0.82	0.87	0.95
GaussianNB	FA	1	0.78	0.67	0.64
GaussianNB	FA	3	0.82	0.72	0.70
GaussianNB	FA	5	0.90	0.82	0.82
GaussianNB	FA	7	0.89	0.83	0.85
GaussianNB	FA	9	0.88	0.83	0.86
GaussianNB	FA	11	0.88	0.83	0.86
GaussianNB	FA	13	0.87	0.83	0.87
GaussianNB	FA	15	0.87	0.85	0.89
DecisionTreeClassifier	FILT	1	0.82	0.87	0.91
DecisionTreeClassifier	PCA	1	0.64	0.65	0.66
DecisionTreeClassifier	PCA	3	0.68	0.74	0.77
DecisionTreeClassifier	PCA	5	0.75	0.82	0.86
DecisionTreeClassifier	PCA	7	0.76	0.83	0.88
DecisionTreeClassifier	PCA	9	0.76	0.83	0.87
DecisionTreeClassifier	PCA	11	0.75	0.82	0.87

DecisionTreeClassifier	PCA	13	0.74	0.81	0.87
DecisionTreeClassifier	PCA	15	0.74	0.81	0.86
DecisionTreeClassifier	KPCA	1	0.64	0.65	0.66
DecisionTreeClassifier	KPCA	3	0.66	0.72	0.76
DecisionTreeClassifier	KPCA	5	0.74	0.80	0.85
DecisionTreeClassifier	KPCA	7	0.75	0.81	0.85
DecisionTreeClassifier	KPCA	9	0.73	0.81	0.86
DecisionTreeClassifier	KPCA	11	0.73	0.80	0.85
DecisionTreeClassifier	KPCA	13	0.72	0.80	0.86
DecisionTreeClassifier	KPCA	15	0.71	0.80	0.86
DecisionTreeClassifier	ICA	1	0.64	0.65	0.66
DecisionTreeClassifier	ICA	3	0.72	0.76	0.79
DecisionTreeClassifier	ICA	5	0.77	0.83	0.87
DecisionTreeClassifier	ICA	7	0.73	0.82	0.87
DecisionTreeClassifier	ICA	9	0.72	0.80	0.86
DecisionTreeClassifier	ICA	11	0.72	0.80	0.87
DecisionTreeClassifier	ICA	13	0.72	0.81	0.88
DecisionTreeClassifier	ICA	15	0.71	0.81	0.89
DecisionTreeClassifier	FA	1	0.64	0.65	0.66
DecisionTreeClassifier	FA	3	0.70	0.75	0.79
DecisionTreeClassifier	FA	5	0.77	0.83	0.88
DecisionTreeClassifier	FA	7	0.77	0.84	0.88
DecisionTreeClassifier	FA	9	0.76	0.83	0.88
DecisionTreeClassifier	FA	11	0.76	0.84	0.88
DecisionTreeClassifier	FA	13	0.75	0.83	0.88
DecisionTreeClassifier	FA	15	0.74	0.82	0.87
MLPClassifier	FILT	1	0.95	0.91	0.94
MLPClassifier	PCA	1	0.82	0.70	0.68
MLPClassifier	PCA	3	0.84	0.76	0.76
MLPClassifier	PCA	5	0.93	0.87	0.90
MLPClassifier	PCA	7	0.93	0.89	0.92
MLPClassifier	PCA	9	0.92	0.89	0.93
MLPClassifier	PCA	11	0.92	0.89	0.93
MLPClassifier	PCA	13	0.93	0.90	0.93
MLPClassifier	PCA	15	0.93	0.91	0.94
MLPClassifier	KPCA	1	0.82	0.71	0.69
MLPClassifier	KPCA	3	0.84	0.75	0.75
MLPClassifier	KPCA	5	0.92	0.86	0.86
MLPClassifier	KPCA	7	0.93	0.90	0.91

MLPClassifier	KPCA	9	0.93	0.89	0.91
MLPClassifier	KPCA	11	0.94	0.89	0.91
MLPClassifier	KPCA	13	0.95	0.89	0.91
MLPClassifier	KPCA	15	0.94	0.89	0.90
MLPClassifier	ICA	1	0.82	0.70	0.68
MLPClassifier	ICA	3	0.84	0.73	0.72
MLPClassifier	ICA	5	0.89	0.83	0.84
MLPClassifier	ICA	7	0.91	0.86	0.87
MLPClassifier	ICA	9	0.91	0.88	0.90
MLPClassifier	ICA	11	0.91	0.88	0.91
MLPClassifier	ICA	13	0.91	0.89	0.91
MLPClassifier	ICA	15	0.92	0.88	0.91
MLPClassifier	FA	1	0.81	0.67	0.62
MLPClassifier	FA	3	0.86	0.77	0.76
MLPClassifier	FA	5	0.93	0.87	0.89
MLPClassifier	FA	7	0.94	0.89	0.91
MLPClassifier	FA	9	0.94	0.91	0.93
MLPClassifier	FA	11	0.95	0.91	0.94
MLPClassifier	FA	13	0.94	0.90	0.94
MLPClassifier	FA	15	0.94	0.90	0.94
BaggingClassifier	FILT	1	0.92	0.90	0.92
BaggingClassifier	PCA	1	0.71	0.65	0.64
BaggingClassifier	PCA	3	0.78	0.74	0.75
BaggingClassifier	PCA	5	0.88	0.84	0.86
BaggingClassifier	PCA	7	0.89	0.84	0.87
BaggingClassifier	PCA	9	0.89	0.85	0.87
BaggingClassifier	PCA	11	0.88	0.83	0.86
BaggingClassifier	PCA	13	0.88	0.84	0.86
BaggingClassifier	PCA	15	0.88	0.84	0.87
BaggingClassifier	KPCA	1	0.71	0.65	0.64
BaggingClassifier	KPCA	3	0.77	0.73	0.75
BaggingClassifier	KPCA	5	0.86	0.82	0.85
BaggingClassifier	KPCA	7	0.88	0.83	0.87
BaggingClassifier	KPCA	9	0.87	0.84	0.87
BaggingClassifier	KPCA	11	0.86	0.83	0.87
BaggingClassifier	KPCA	13	0.87	0.83	0.87
BaggingClassifier	KPCA	15	0.86	0.82	0.86
BaggingClassifier	ICA	1	0.71	0.65	0.64
BaggingClassifier	ICA	3	0.82	0.76	0.77

BaggingClassifier	ICA	5	0.89	0.84	0.87
BaggingClassifier	ICA	7	0.88	0.84	0.87
BaggingClassifier	ICA	9	0.85	0.83	0.88
BaggingClassifier	ICA	11	0.85	0.83	0.88
BaggingClassifier	ICA	13	0.86	0.84	0.90
BaggingClassifier	ICA	15	0.85	0.84	0.91
BaggingClassifier	FA	1	0.71	0.65	0.64
BaggingClassifier	FA	3	0.81	0.74	0.76
BaggingClassifier	FA	5	0.89	0.84	0.87
BaggingClassifier	FA	7	0.90	0.85	0.88
BaggingClassifier	FA	9	0.89	0.84	0.87
BaggingClassifier	FA	11	0.90	0.85	0.88
BaggingClassifier	FA	13	0.90	0.85	0.87
BaggingClassifier	FA	15	0.90	0.84	0.87
GradientBoostingClassifier	FILT	1	0.94	0.91	0.94
GradientBoostingClassifier	PCA	1	0.74	0.64	0.64
GradientBoostingClassifier	PCA	3	0.80	0.75	0.77
GradientBoostingClassifier	PCA	5	0.90	0.85	0.89
GradientBoostingClassifier	PCA	7	0.91	0.87	0.90
GradientBoostingClassifier	PCA	9	0.91	0.86	0.90
GradientBoostingClassifier	PCA	11	0.91	0.86	0.90
GradientBoostingClassifier	PCA	13	0.91	0.87	0.91
GradientBoostingClassifier	PCA	15	0.91	0.87	0.91
GradientBoostingClassifier	KPCA	1	0.75	0.65	0.64
GradientBoostingClassifier	KPCA	3	0.80	0.74	0.76
GradientBoostingClassifier	KPCA	5	0.88	0.84	0.88
GradientBoostingClassifier	KPCA	7	0.91	0.87	0.91
GradientBoostingClassifier	KPCA	9	0.91	0.86	0.91
GradientBoostingClassifier	KPCA	11	0.91	0.87	0.92
GradientBoostingClassifier	KPCA	13	0.91	0.87	0.92
GradientBoostingClassifier	KPCA	15	0.91	0.87	0.91
GradientBoostingClassifier	ICA	1	0.74	0.64	0.64
GradientBoostingClassifier	ICA	3	0.84	0.78	0.79
GradientBoostingClassifier	ICA	5	0.90	0.86	0.89
GradientBoostingClassifier	ICA	7	0.90	0.86	0.91
GradientBoostingClassifier	ICA	9	0.87	0.86	0.91
GradientBoostingClassifier	ICA	11	0.88	0.86	0.92
GradientBoostingClassifier	ICA	13	0.88	0.87	0.93
GradientBoostingClassifier	ICA	15	0.87	0.87	0.95

GradientBoostingClassifier	FA	1	0.74	0.64	0.62
GradientBoostingClassifier	FA	3	0.82	0.76	0.78
GradientBoostingClassifier	FA	5	0.91	0.85	0.88
GradientBoostingClassifier	FA	7	0.92	0.87	0.90
GradientBoostingClassifier	FA	9	0.92	0.87	0.90
GradientBoostingClassifier	FA	11	0.92	0.87	0.90
GradientBoostingClassifier	FA	13	0.92	0.87	0.90
GradientBoostingClassifier	FA	15	0.93	0.88	0.92
SVC	FILT	1	0.94	0.90	0.92
SVC	PCA	1	0.70	0.65	0.65
SVC	PCA	3	0.70	0.80	0.98
SVC	PCA	5	0.80	0.80	1.00
SVC	PCA	7	0.80	0.80	1.00
SVC	PCA	9	0.82	0.80	1.00
SVC	PCA	11	0.83	0.80	1.00
SVC	PCA	13	0.83	0.80	1.00
SVC	PCA	15	0.83	0.80	1.00
SVC	KPCA	1	0.81	0.68	0.64
SVC	KPCA	3	0.84	0.74	0.73
SVC	KPCA	5	0.92	0.83	0.82
SVC	KPCA	7	0.93	0.86	0.86
SVC	KPCA	9	0.92	0.85	0.85
SVC	KPCA	11	0.93	0.83	0.83
SVC	KPCA	13	0.92	0.82	0.82
SVC	KPCA	15	0.92	0.81	0.81
SVC	ICA	1	0.82	0.66	0.62
SVC	ICA	3	0.84	0.62	0.55
SVC	ICA	5	0.90	0.83	0.83
SVC	ICA	7	0.84	0.86	0.87
SVC	ICA	9	0.70	0.87	0.88
SVC	ICA	11	0.57	0.84	0.84
SVC	ICA	13	0.43	0.80	0.78
SVC	ICA	15	0.39	0.83	0.82
SVC	FA	1	0.74	0.64	0.57
SVC	FA	3	0.83	0.75	0.73
SVC	FA	5	0.92	0.86	0.88
SVC	FA	7	0.94	0.88	0.91
SVC	FA	9	0.95	0.89	0.93
SVC	FA	11	0.95	0.90	0.94

SVC	FA	13	0.94	0.90	0.95
SVC	FA	15	0.95	0.90	0.96
LogisticRegression	FILT	1	0.94	0.89	0.91
LogisticRegression	PCA	1	0.82	0.71	0.69
LogisticRegression	PCA	3	0.85	0.75	0.75
LogisticRegression	PCA	5	0.91	0.86	0.87
LogisticRegression	PCA	7	0.92	0.88	0.89
LogisticRegression	PCA	9	0.91	0.88	0.90
LogisticRegression	PCA	11	0.91	0.88	0.91
LogisticRegression	PCA	13	0.92	0.89	0.91
LogisticRegression	PCA	15	0.92	0.89	0.91
LogisticRegression	KPCA	1	0.82	0.71	0.70
LogisticRegression	KPCA	3	0.84	0.74	0.74
LogisticRegression	KPCA	5	0.91	0.84	0.84
LogisticRegression	KPCA	7	0.93	0.87	0.88
LogisticRegression	KPCA	9	0.93	0.87	0.88
LogisticRegression	KPCA	11	0.93	0.87	0.87
LogisticRegression	KPCA	13	0.93	0.87	0.87
LogisticRegression	KPCA	15	0.93	0.87	0.87
LogisticRegression	ICA	1	0.82	0.70	0.68
LogisticRegression	ICA	3	0.85	0.74	0.73
LogisticRegression	ICA	5	0.90	0.85	0.85
LogisticRegression	ICA	7	0.92	0.88	0.89
LogisticRegression	ICA	9	0.91	0.88	0.90
LogisticRegression	ICA	11	0.91	0.89	0.91
LogisticRegression	ICA	13	0.92	0.89	0.91
LogisticRegression	ICA	15	0.92	0.89	0.91
LogisticRegression	FA	1	0.81	0.68	0.66
LogisticRegression	FA	3	0.84	0.73	0.72
LogisticRegression	FA	5	0.91	0.85	0.86
LogisticRegression	FA	7	0.92	0.85	0.86
LogisticRegression	FA	9	0.91	0.86	0.87
LogisticRegression	FA	11	0.92	0.87	0.88
LogisticRegression	FA	13	0.92	0.88	0.89
LogisticRegression	FA	15	0.93	0.89	0.91
KNeighborsClassifier	FILT	1	0.91	0.87	0.88
KNeighborsClassifier	PCA	1	0.74	0.67	0.65
KNeighborsClassifier	PCA	3	0.81	0.75	0.76
KNeighborsClassifier	PCA	5	0.88	0.83	0.85

KNeighborsClassifier	PCA	7	0.90	0.85	0.86
KNeighborsClassifier	PCA	9	0.91	0.86	0.87
KNeighborsClassifier	PCA	11	0.92	0.86	0.87
KNeighborsClassifier	PCA	13	0.92	0.87	0.88
KNeighborsClassifier	PCA	15	0.92	0.86	0.88
KNeighborsClassifier	KPCA	1	0.74	0.67	0.66
KNeighborsClassifier	KPCA	3	0.80	0.73	0.73
KNeighborsClassifier	KPCA	5	0.87	0.83	0.84
KNeighborsClassifier	KPCA	7	0.90	0.85	0.86
KNeighborsClassifier	KPCA	9	0.91	0.85	0.86
KNeighborsClassifier	KPCA	11	0.91	0.85	0.86
KNeighborsClassifier	KPCA	13	0.92	0.86	0.87
KNeighborsClassifier	KPCA	15	0.92	0.86	0.87
KNeighborsClassifier	ICA	1	0.74	0.67	0.65
KNeighborsClassifier	ICA	3	0.82	0.76	0.76
KNeighborsClassifier	ICA	5	0.88	0.83	0.85
KNeighborsClassifier	ICA	7	0.90	0.85	0.86
KNeighborsClassifier	ICA	9	0.91	0.85	0.86
KNeighborsClassifier	ICA	11	0.91	0.85	0.87
KNeighborsClassifier	ICA	13	0.92	0.86	0.87
KNeighborsClassifier	ICA	15	0.91	0.85	0.87
KNeighborsClassifier	FA	1	0.73	0.65	0.63
KNeighborsClassifier	FA	3	0.82	0.76	0.75
KNeighborsClassifier	FA	5	0.89	0.83	0.83
KNeighborsClassifier	FA	7	0.92	0.86	0.86
KNeighborsClassifier	FA	9	0.93	0.86	0.87
KNeighborsClassifier	FA	11	0.93	0.86	0.87
KNeighborsClassifier	FA	13	0.92	0.86	0.88
KNeighborsClassifier	FA	15	0.91	0.86	0.88

Table S2. Comparison of scoring function results for survival classification task.

Classifier	Selection Method	Dimension Number	μ_{AUC}	$\mu_{Accuracy}$	$\mu_{Sensitivity}$
RandomForestClassifier	FILT	1	0.59	0.56	0.50
RandomForestClassifier	PCA	1	0.48	0.48	0.48
RandomForestClassifier	PCA	3	0.55	0.53	0.47
RandomForestClassifier	PCA	5	0.54	0.51	0.47
RandomForestClassifier	PCA	7	0.52	0.51	0.43
RandomForestClassifier	PCA	9	0.58	0.56	0.49
RandomForestClassifier	PCA	11	0.58	0.56	0.48
RandomForestClassifier	PCA	13	0.57	0.55	0.49
RandomForestClassifier	PCA	15	0.57	0.55	0.47
RandomForestClassifier	KPCA	1	0.48	0.48	0.48
RandomForestClassifier	KPCA	3	0.55	0.53	0.48
RandomForestClassifier	KPCA	5	0.57	0.54	0.49
RandomForestClassifier	KPCA	7	0.56	0.54	0.47
RandomForestClassifier	KPCA	9	0.58	0.54	0.49
RandomForestClassifier	KPCA	11	0.54	0.53	0.46
RandomForestClassifier	KPCA	13	0.55	0.53	0.47
RandomForestClassifier	KPCA	15	0.55	0.53	0.48
RandomForestClassifier	ICA	1	0.48	0.48	0.48
RandomForestClassifier	ICA	3	0.56	0.54	0.48
RandomForestClassifier	ICA	5	0.56	0.53	0.47
RandomForestClassifier	ICA	7	0.56	0.54	0.49
RandomForestClassifier	ICA	9	0.58	0.55	0.49
RandomForestClassifier	ICA	11	0.56	0.54	0.48
RandomForestClassifier	ICA	13	0.56	0.54	0.47
RandomForestClassifier	ICA	15	0.56	0.55	0.49
RandomForestClassifier	FA	1	0.58	0.56	0.54
RandomForestClassifier	FA	3	0.54	0.52	0.48
RandomForestClassifier	FA	5	0.52	0.51	0.46
RandomForestClassifier	FA	7	0.55	0.53	0.46
RandomForestClassifier	FA	9	0.57	0.54	0.47
RandomForestClassifier	FA	11	0.59	0.56	0.50
RandomForestClassifier	FA	13	0.55	0.53	0.46
RandomForestClassifier	FA	15	0.56	0.54	0.47
GaussianNB	FILT	1	0.63	0.58	0.67
GaussianNB	PCA	1	0.52	0.49	0.71

GaussianNB	PCA	3	0.57	0.55	0.50
GaussianNB	PCA	5	0.56	0.52	0.40
GaussianNB	PCA	7	0.54	0.51	0.35
GaussianNB	PCA	9	0.59	0.55	0.39
GaussianNB	PCA	11	0.60	0.57	0.47
GaussianNB	PCA	13	0.59	0.56	0.48
GaussianNB	PCA	15	0.58	0.56	0.48
GaussianNB	KPCA	1	0.53	0.50	0.72
GaussianNB	KPCA	3	0.58	0.55	0.56
GaussianNB	KPCA	5	0.59	0.55	0.52
GaussianNB	KPCA	7	0.59	0.56	0.56
GaussianNB	KPCA	9	0.60	0.56	0.57
GaussianNB	KPCA	11	0.58	0.56	0.57
GaussianNB	KPCA	13	0.56	0.54	0.59
GaussianNB	KPCA	15	0.55	0.53	0.57
GaussianNB	ICA	1	0.52	0.49	0.71
GaussianNB	ICA	3	0.54	0.53	0.49
GaussianNB	ICA	5	0.55	0.54	0.36
GaussianNB	ICA	7	0.51	0.51	0.27
GaussianNB	ICA	9	0.52	0.51	0.29
GaussianNB	ICA	11	0.55	0.54	0.38
GaussianNB	ICA	13	0.54	0.53	0.41
GaussianNB	ICA	15	0.53	0.53	0.44
GaussianNB	FA	1	0.56	0.50	0.75
GaussianNB	FA	3	0.55	0.53	0.50
GaussianNB	FA	5	0.54	0.51	0.44
GaussianNB	FA	7	0.54	0.54	0.47
GaussianNB	FA	9	0.58	0.57	0.50
GaussianNB	FA	11	0.62	0.58	0.53
GaussianNB	FA	13	0.60	0.57	0.51
GaussianNB	FA	15	0.57	0.55	0.46
DecisionTreeClassifier	FILT	1	0.52	0.52	0.53
DecisionTreeClassifier	PCA	1	0.49	0.48	0.51
DecisionTreeClassifier	PCA	3	0.51	0.50	0.52
DecisionTreeClassifier	PCA	5	0.52	0.51	0.53
DecisionTreeClassifier	PCA	7	0.51	0.51	0.52
DecisionTreeClassifier	PCA	9	0.53	0.53	0.54
DecisionTreeClassifier	PCA	11	0.53	0.53	0.54
DecisionTreeClassifier	PCA	13	0.53	0.53	0.54

DecisionTreeClassifier	PCA	15	0.54	0.54	0.55
DecisionTreeClassifier	KPCA	1	0.48	0.48	0.52
DecisionTreeClassifier	KPCA	3	0.53	0.52	0.55
DecisionTreeClassifier	KPCA	5	0.55	0.54	0.56
DecisionTreeClassifier	KPCA	7	0.53	0.53	0.55
DecisionTreeClassifier	KPCA	9	0.53	0.53	0.55
DecisionTreeClassifier	KPCA	11	0.53	0.53	0.55
DecisionTreeClassifier	KPCA	13	0.52	0.52	0.53
DecisionTreeClassifier	KPCA	15	0.51	0.51	0.53
DecisionTreeClassifier	ICA	1	0.49	0.48	0.51
DecisionTreeClassifier	ICA	3	0.54	0.54	0.55
DecisionTreeClassifier	ICA	5	0.54	0.53	0.54
DecisionTreeClassifier	ICA	7	0.54	0.54	0.54
DecisionTreeClassifier	ICA	9	0.55	0.55	0.56
DecisionTreeClassifier	ICA	11	0.53	0.53	0.55
DecisionTreeClassifier	ICA	13	0.52	0.52	0.53
DecisionTreeClassifier	ICA	15	0.53	0.53	0.54
DecisionTreeClassifier	FA	1	0.56	0.56	0.56
DecisionTreeClassifier	FA	3	0.52	0.51	0.52
DecisionTreeClassifier	FA	5	0.49	0.49	0.50
DecisionTreeClassifier	FA	7	0.51	0.51	0.52
DecisionTreeClassifier	FA	9	0.53	0.53	0.54
DecisionTreeClassifier	FA	11	0.53	0.53	0.53
DecisionTreeClassifier	FA	13	0.53	0.53	0.54
DecisionTreeClassifier	FA	15	0.52	0.52	0.52
MLPClassifier	FILT	1	0.60	0.57	0.57
MLPClassifier	PCA	1	0.52	0.50	0.38
MLPClassifier	PCA	3	0.58	0.57	0.56
MLPClassifier	PCA	5	0.59	0.56	0.55
MLPClassifier	PCA	7	0.59	0.56	0.57
MLPClassifier	PCA	9	0.59	0.56	0.58
MLPClassifier	PCA	11	0.58	0.57	0.59
MLPClassifier	PCA	13	0.58	0.57	0.59
MLPClassifier	PCA	15	0.56	0.55	0.57
MLPClassifier	KPCA	1	0.52	0.47	0.66
MLPClassifier	KPCA	3	0.55	0.51	0.58
MLPClassifier	KPCA	5	0.57	0.53	0.55
MLPClassifier	KPCA	7	0.60	0.56	0.57
MLPClassifier	KPCA	9	0.63	0.59	0.62

MLPClassifier	KPCA	11	0.61	0.57	0.59
MLPClassifier	KPCA	13	0.60	0.57	0.60
MLPClassifier	KPCA	15	0.60	0.57	0.60
MLPClassifier	ICA	1	0.46	0.44	0.69
MLPClassifier	ICA	3	0.51	0.46	0.70
MLPClassifier	ICA	5	0.56	0.51	0.55
MLPClassifier	ICA	7	0.54	0.51	0.52
MLPClassifier	ICA	9	0.61	0.57	0.60
MLPClassifier	ICA	11	0.63	0.58	0.63
MLPClassifier	ICA	13	0.63	0.58	0.63
MLPClassifier	ICA	15	0.62	0.57	0.63
MLPClassifier	FA	1	0.59	0.52	0.67
MLPClassifier	FA	3	0.56	0.55	0.56
MLPClassifier	FA	5	0.55	0.53	0.54
MLPClassifier	FA	7	0.55	0.54	0.54
MLPClassifier	FA	9	0.59	0.56	0.57
MLPClassifier	FA	11	0.61	0.59	0.62
MLPClassifier	FA	13	0.60	0.59	0.60
MLPClassifier	FA	15	0.59	0.56	0.58
BaggingClassifier	FILT	1	0.58	0.55	0.48
BaggingClassifier	PCA	1	0.48	0.48	0.48
BaggingClassifier	PCA	3	0.54	0.51	0.45
BaggingClassifier	PCA	5	0.55	0.52	0.46
BaggingClassifier	PCA	7	0.53	0.51	0.45
BaggingClassifier	PCA	9	0.56	0.54	0.48
BaggingClassifier	PCA	11	0.59	0.56	0.50
BaggingClassifier	PCA	13	0.58	0.55	0.48
BaggingClassifier	PCA	15	0.59	0.56	0.49
BaggingClassifier	KPCA	1	0.48	0.48	0.48
BaggingClassifier	KPCA	3	0.55	0.53	0.47
BaggingClassifier	KPCA	5	0.57	0.54	0.49
BaggingClassifier	KPCA	7	0.57	0.54	0.47
BaggingClassifier	KPCA	9	0.57	0.55	0.47
BaggingClassifier	KPCA	11	0.55	0.53	0.47
BaggingClassifier	KPCA	13	0.54	0.52	0.46
BaggingClassifier	KPCA	15	0.54	0.53	0.46
BaggingClassifier	ICA	1	0.48	0.48	0.48
BaggingClassifier	ICA	3	0.56	0.53	0.47
BaggingClassifier	ICA	5	0.56	0.54	0.48

BaggingClassifier	ICA	7	0.57	0.53	0.46
BaggingClassifier	ICA	9	0.59	0.56	0.51
BaggingClassifier	ICA	11	0.57	0.54	0.49
BaggingClassifier	ICA	13	0.56	0.54	0.48
BaggingClassifier	ICA	15	0.58	0.56	0.48
BaggingClassifier	FA	1	0.58	0.56	0.54
BaggingClassifier	FA	3	0.54	0.53	0.48
BaggingClassifier	FA	5	0.52	0.50	0.46
BaggingClassifier	FA	7	0.55	0.53	0.46
BaggingClassifier	FA	9	0.57	0.55	0.48
BaggingClassifier	FA	11	0.59	0.56	0.51
BaggingClassifier	FA	13	0.58	0.56	0.51
BaggingClassifier	FA	15	0.56	0.53	0.47
GradientBoostingClassifier	FILT	1	0.58	0.56	0.56
GradientBoostingClassifier	PCA	1	0.48	0.49	0.53
GradientBoostingClassifier	PCA	3	0.52	0.50	0.51
GradientBoostingClassifier	PCA	5	0.50	0.50	0.53
GradientBoostingClassifier	PCA	7	0.50	0.49	0.51
GradientBoostingClassifier	PCA	9	0.55	0.52	0.56
GradientBoostingClassifier	PCA	11	0.57	0.55	0.56
GradientBoostingClassifier	PCA	13	0.57	0.55	0.56
GradientBoostingClassifier	PCA	15	0.57	0.55	0.56
GradientBoostingClassifier	KPCA	1	0.47	0.48	0.53
GradientBoostingClassifier	KPCA	3	0.53	0.52	0.54
GradientBoostingClassifier	KPCA	5	0.55	0.53	0.55
GradientBoostingClassifier	KPCA	7	0.55	0.53	0.55
GradientBoostingClassifier	KPCA	9	0.57	0.54	0.55
GradientBoostingClassifier	KPCA	11	0.56	0.53	0.54
GradientBoostingClassifier	KPCA	13	0.54	0.53	0.54
GradientBoostingClassifier	KPCA	15	0.55	0.53	0.55
GradientBoostingClassifier	ICA	1	0.48	0.49	0.53
GradientBoostingClassifier	ICA	3	0.55	0.54	0.56
GradientBoostingClassifier	ICA	5	0.54	0.53	0.54
GradientBoostingClassifier	ICA	7	0.57	0.55	0.55
GradientBoostingClassifier	ICA	9	0.59	0.56	0.58
GradientBoostingClassifier	ICA	11	0.56	0.54	0.56
GradientBoostingClassifier	ICA	13	0.55	0.53	0.54
GradientBoostingClassifier	ICA	15	0.57	0.54	0.55
GradientBoostingClassifier	FA	1	0.55	0.56	0.58

GradientBoostingClassifier	FA	3	0.53	0.52	0.54
GradientBoostingClassifier	FA	5	0.50	0.50	0.54
GradientBoostingClassifier	FA	7	0.55	0.52	0.53
GradientBoostingClassifier	FA	9	0.58	0.55	0.59
GradientBoostingClassifier	FA	11	0.59	0.55	0.57
GradientBoostingClassifier	FA	13	0.57	0.55	0.57
GradientBoostingClassifier	FA	15	0.56	0.54	0.56
SVC	FILT	1	0.61	0.57	0.59
SVC	PCA	1	0.52	0.46	0.50
SVC	PCA	3	0.50	0.47	0.73
SVC	PCA	5	0.41	0.45	0.75
SVC	PCA	7	0.49	0.45	0.74
SVC	PCA	9	0.48	0.45	0.75
SVC	PCA	11	0.48	0.45	0.75
SVC	PCA	13	0.46	0.45	0.75
SVC	PCA	15	0.46	0.45	0.75
SVC	KPCA	1	0.45	0.49	0.81
SVC	KPCA	3	0.45	0.45	0.70
SVC	KPCA	5	0.45	0.46	0.73
SVC	KPCA	7	0.47	0.46	0.73
SVC	KPCA	9	0.46	0.46	0.74
SVC	KPCA	11	0.46	0.46	0.75
SVC	KPCA	13	0.46	0.46	0.75
SVC	KPCA	15	0.47	0.46	0.75
SVC	ICA	1	0.48	0.45	0.75
SVC	ICA	3	0.48	0.45	0.73
SVC	ICA	5	0.47	0.45	0.72
SVC	ICA	7	0.48	0.45	0.73
SVC	ICA	9	0.44	0.46	0.74
SVC	ICA	11	0.44	0.46	0.76
SVC	ICA	13	0.44	0.46	0.77
SVC	ICA	15	0.44	0.46	0.77
SVC	FA	1	0.55	0.52	0.63
SVC	FA	3	0.51	0.53	0.55
SVC	FA	5	0.53	0.53	0.54
SVC	FA	7	0.53	0.51	0.51
SVC	FA	9	0.57	0.53	0.57
SVC	FA	11	0.60	0.56	0.61
SVC	FA	13	0.60	0.55	0.58

SVC	FA	15	0.59	0.55	0.58
LogisticRegression	FILT	1	0.58	0.57	0.58
LogisticRegression	PCA	1	0.51	0.48	0.63
LogisticRegression	PCA	3	0.55	0.51	0.60
LogisticRegression	PCA	5	0.55	0.52	0.53
LogisticRegression	PCA	7	0.55	0.52	0.53
LogisticRegression	PCA	9	0.62	0.58	0.61
LogisticRegression	PCA	11	0.63	0.59	0.63
LogisticRegression	PCA	13	0.62	0.57	0.61
LogisticRegression	PCA	15	0.61	0.57	0.63
LogisticRegression	KPCA	1	0.53	0.48	0.66
LogisticRegression	KPCA	3	0.53	0.48	0.62
LogisticRegression	KPCA	5	0.55	0.50	0.58
LogisticRegression	KPCA	7	0.58	0.52	0.61
LogisticRegression	KPCA	9	0.60	0.55	0.65
LogisticRegression	KPCA	11	0.60	0.55	0.64
LogisticRegression	KPCA	13	0.60	0.54	0.65
LogisticRegression	KPCA	15	0.60	0.54	0.64
LogisticRegression	ICA	1	0.51	0.46	0.75
LogisticRegression	ICA	3	0.54	0.45	0.69
LogisticRegression	ICA	5	0.55	0.48	0.67
LogisticRegression	ICA	7	0.54	0.47	0.66
LogisticRegression	ICA	9	0.61	0.52	0.67
LogisticRegression	ICA	11	0.63	0.55	0.70
LogisticRegression	ICA	13	0.62	0.54	0.70
LogisticRegression	ICA	15	0.62	0.54	0.69
LogisticRegression	FA	1	0.50	0.47	0.65
LogisticRegression	FA	3	0.54	0.51	0.59
LogisticRegression	FA	5	0.53	0.50	0.56
LogisticRegression	FA	7	0.55	0.52	0.58
LogisticRegression	FA	9	0.61	0.58	0.62
LogisticRegression	FA	11	0.63	0.59	0.64
LogisticRegression	FA	13	0.62	0.58	0.63
LogisticRegression	FA	15	0.61	0.58	0.63
KNeighborsClassifier	FILT	1	0.57	0.55	0.54
KNeighborsClassifier	PCA	1	0.46	0.45	0.47
KNeighborsClassifier	PCA	3	0.57	0.55	0.56
KNeighborsClassifier	PCA	5	0.60	0.57	0.55
KNeighborsClassifier	PCA	7	0.59	0.56	0.51

KNeighborsClassifier	PCA	9	0.59	0.56	0.51
KNeighborsClassifier	PCA	11	0.59	0.56	0.51
KNeighborsClassifier	PCA	13	0.58	0.55	0.48
KNeighborsClassifier	PCA	15	0.58	0.55	0.47
KNeighborsClassifier	KPCA	1	0.48	0.48	0.52
KNeighborsClassifier	KPCA	3	0.57	0.54	0.56
KNeighborsClassifier	KPCA	5	0.61	0.58	0.57
KNeighborsClassifier	KPCA	7	0.60	0.56	0.53
KNeighborsClassifier	KPCA	9	0.61	0.57	0.53
KNeighborsClassifier	KPCA	11	0.59	0.56	0.51
KNeighborsClassifier	KPCA	13	0.59	0.56	0.50
KNeighborsClassifier	KPCA	15	0.59	0.55	0.51
KNeighborsClassifier	ICA	1	0.46	0.45	0.47
KNeighborsClassifier	ICA	3	0.58	0.56	0.55
KNeighborsClassifier	ICA	5	0.59	0.54	0.55
KNeighborsClassifier	ICA	7	0.53	0.51	0.50
KNeighborsClassifier	ICA	9	0.56	0.53	0.47
KNeighborsClassifier	ICA	11	0.57	0.54	0.49
KNeighborsClassifier	ICA	13	0.55	0.53	0.46
KNeighborsClassifier	ICA	15	0.54	0.52	0.46
KNeighborsClassifier	FA	1	0.59	0.57	0.61
KNeighborsClassifier	FA	3	0.55	0.54	0.53
KNeighborsClassifier	FA	5	0.54	0.52	0.49
KNeighborsClassifier	FA	7	0.54	0.51	0.48
KNeighborsClassifier	FA	9	0.57	0.54	0.52
KNeighborsClassifier	FA	11	0.58	0.55	0.54
KNeighborsClassifier	FA	13	0.57	0.54	0.55
KNeighborsClassifier	FA	15	0.56	0.54	0.55

Table S3. Classifier performance (μ_{AUC}) using subsets of features for grade classification task. All features in the subsets were used with no feature reduction.

Classifier	Shape	Intensity/Texture	Intensity/Texture + Wavelet	All
RF	0.89	0.88	0.94	0.94
NB	0.90	0.83	0.80	0.80
DT	0.78	0.75	0.79	0.79
MLP	0.91	0.92	0.92	0.92
BAG	0.88	0.89	0.92	0.92
BST	0.91	0.91	0.94	0.94
SVM	0.92	0.91	0.94	0.94
LR	0.93	0.90	0.90	0.90
KNN	0.89	0.86	0.93	0.93

Table S4. Classifier performance (μ_{AUC}) using subsets of features for survival classification task. All features in the subsets were used with no feature reduction.

Classifier	Shape	Intensity/Texture	Intensity/Texture + Wavelet	All
RF	0.61	0.58	0.57	0.56
NB	0.64	0.54	0.54	0.55
DT	0.56	0.52	0.52	0.52
MLP	0.58	0.55	0.59	0.58
BAG	0.61	0.57	0.58	0.57
BST	0.63	0.55	0.58	0.58
SVM	0.66	0.61	0.59	0.60
LR	0.60	0.55	0.57	0.58
KNN	0.60	0.55	0.57	0.57

Table S5. Multivariate ANOVA results for grade classification task. CLF = Classifier method, DRM = Dimensionality Reduction Method, DN = Dimension number.

Experimental Factors	**% Variance**	**df**	**F**	**PR(>F)**
CLF	36.393001	8	60.119487	2.19E-45
DRM	3.098637	3	13.65013	5.19E-08
DN	27.893836	7	52.662029	3.49E-39
CLF:DN	5.9535	56	1.404985	5.13E-02
DRM:DN	3.557652	21	2.238884	2.54E-03
CLF:DRM	10.39114	24	5.721888	2.26E-12
Residual	12.712235	168	NaN	NaN

Table S6. Multivariate ANOVA results for survival classification task.

Experimental Factors	**% Variance**	**df**	**F**	**PR(>F)**
CLF	36.624461	8	67.209435	2.08E-48
DRM	2.379755	3	11.645548	5.68E-07
DN	19.92892	7	41.795999	1.08E-33
CLF:DN	9.824412	56	2.575535	1.65E-06
DRM:DN	8.324374	21	5.819441	1.09E-11
CLF:DRM	11.474541	24	7.018965	2.05E-15
Residual	11.443537	168	NaN	NaN

References

1 Menze, B. H. *et al.* The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging* **34**, 1993-2024, doi:10.1109/TMI.2014.2377694 (2015).

2 Fedorov, A. *et al.* 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic Resonance Imaging* **30**, 1323-1341, doi:https://doi.org/10.1016/j.mri.2012.05.001 (2012).

3 Joost JM van Griethuysen, A. F., Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina GH Beets-Tan, Jean-Christophe Fillion-Robin, Steve Pieper, Hugo JWL Aerts. Computational Radiomics System to Decode the Radiographic Phenotype. *Cancer Research* (2017).

4 Aerts, H. J. W. L. *et al.* Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. *Nature Communications* **5**, 4006, doi:10.1038/ncomms5006 (2014).

5 Parmar, C., Grossmann, P., Bussink, J., Lambin, P. & Aerts, H. J. W. L. Machine Learning methods for Quantitative Radiomic Biomarkers. *Scientific Reports* **5**, 13087, doi:10.1038/srep13087

https://www.nature.com/articles/srep13087 - supplementary-information (2015).
6 Parmar, C. *et al.* Radiomic Machine-Learning Classifiers for Prognostic Biomarkers of Head and Neck Cancer. *Frontiers in Oncology* **5**, 272, doi:10.3389/fonc.2015.00272 (2015).
7 Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825-2830 (2011).
8 James, G., Witten, D., Hastie, T. & Tibshirani, R. *An introduction to statistical learning*. Vol. 112 (Springer, 2013).
9 Hyvärinen, A., Karhunen, J. & Oja, E. *Independent component analysis*. Vol. 46 (John Wiley & Sons, 2004).
10 Brown, T. A. *Confirmatory factor analysis for applied research*. (Guilford Publications, 2014).
11 Hawkins, S. H. *et al.* Predicting Outcomes of Nonsmall Cell Lung Cancer Using CT Image Features. *IEEE Access* **2**, 1418-1426, doi:10.1109/ACCESS.2014.2373335 (2014).
12 Dietterich, T. G. Ensemble methods in machine learning. *Multiple classifier systems* **1857**, 1-15 (2000).
13 Raschka, S. *Python machine learning*. (Packt Publishing Ltd, 2015).
14 Cortes, C. & Vapnik, V. Support vector machine. *Machine learning* **20**, 273-297 (1995).