Predicting Ovarian Cancer with Machine Learning: Integrating Clinical and Genetic Data

Ismael Gómez-Talal
Dept. Sig. Theory Comms.
Universidad Rey Juan Carlos
Fuenlabrada (Madrid), Spain
ORCID 0000-0003-4673-8193

Arantzazu Barquín
Unit of Gyn., Genitour. Skin Tum.
Hospital HM Sanchinarro
Madrid, Spain
ORCID 0000-0002-3701-0347

Luis Bote-Curiel
Dept. Sig. Theory Comms.
Universidad Rey Juan Carlos
Fuenlabrada (Madrid), Spain
ORCID 0000-0001-8845-2834

Mónica Yagüe-Fernández
Unit of Gyn., Genitour. Skin Tum.
Hospital HM Sanchinarro
Madrid, Spain
ORCID 0000-0001-9438-5703

José Luis Rojo-Álvarez
Dept. Sig. Theory Comms.
Universidad Rey Juan Carlos
Fuenlabrada (Madrid), Spain
ORCID 0000-0003-0426-8912

Jesús García-Donás
Unit of Gyn., Genitour. Skin Tum.
Hospital HM Sanchinarro
Madrid, Spain
ORCID 0000-0001-7731-3601

***Abstract*—Ovarian cancer (OC) is a deadly disease that affects a large number of women worldwide. Machine Learning (ML) models can help in the early detection of this disease, however, the use of these models may be limited by their lack of interpretability and the difficulty to evaluate their performance. In this work, five types of datasets were used, employing clinical features, different types of coding genomic features, and combining both. The use of interpretable ML (IML) models (one linear and one nonlinear model) provided us with better interpretability of the five feature sets. Following this study, nine binary classification models were compared, and the Accuracy, Recall, and Area Under the Curve were analyzed. The results showed that ML models employed the combination of clinical features and genomes with the coding of the position of genes in patients significantly improves the prediction. We demonstrated that the inclusion of different preprocessed patient data and especially through the information provided by IML models, can help clinicians to understand the disease better and make informed treatment decisions.**

***Index Terms*—Interpretable Machine Learning, Genetic Data, Ovarian Cancer, Classification, Healthcare, Manifold Learning**

I. Introduction

Ovarian cancer (OC) represents one of the most serious types of cancer in our society, with an incidence of about 225,000 women in the world, considered as the gynecological tumor with the worst survival prognosis (140,000 exitus per year) [1]. One of the main problems of OC is the difficult detection in the early stages, however, the advent of Machine Learning (ML) is expected to provide improvement and accuracy in diagnosis. The main applications of ML in OC have been mainly using medical images, such as ultrasound, computed tomography, or magnetic resonance imaging. One of the studies using 2D imaging based on clinical data obtained an accuracy of 78 % in predicting survival in advanced OC [2]. Currently, data science is used in the field of OC research in an applied way, such as in pathology diagnosis and determination of the response to treatment and malignant properties of OC tumors under study. However, one of the main barriers today is data collection, and generally, most studies focus mainly on transcriptomic and proteomic profiling. Neural networks have also been used for identifying a subset of proteins and catabolic pathways with clinical features in the diagnosis [3].

In healthcare and other sectors, datasets contain a series of characteristics (variables) of different types such as metric variables, categorical variables, factual variables, and text. However, many types of datasets use the same type of data and require time-consuming preprocessing. For example, in this work there are in the dataset the collection of the genomes of each patient where relevant information appears, as the field of the position that occupies that genome in the patient. The treatment and time spent in the preprocessing of the data can result in a considerable improvement of the classification and regression models because the model must sometimes have the information collected in a way that maximizes the greatest linear combination of the relationships between the variables.

In this work, the first contribution is the proposal of different possibilities of oncological data processing where different possibilities of dataset of clinical data are proposed. In addition, we propose different types of genomic data coding and the combination of clinical and genomic features. The first study is framed in the study of the use of algorithms based on Interpretable ML (IML) to visualize the samples of the patients through their projection of low-dimensional latent spaces, and the behavior of each of the cases and the class latent distribution to subsequently improve their prediction. In the last part, a set of supervised ML models are proposed in order to evaluate and study the performance of each of the models according to the metrics that evaluate the binary classification. Therefore, our objective is to define a binary classification model that obtains better performance based on a set of metrics based on hyperparameter adjustment and data preprocessing, by combining the data and coding the genomic features of the patients.

II. Material and Methods

A. Manifolds Learning Models

The two Manifold Learning models used to study the latent spaces of the projections are the (linear model of) principal component analysis (PCA) and the (nonlinear model of) Uniform Manifold Approximation and Projection (UMAP). PCA is a dimensionality reduction method that aims to project a set of samples into a lower dimensional latent space, in this particular work, into a three-dimensional latent space, while preserving as many key features of the entire data set as possible. The mathematical formulation of PCA is based on given a dataset X, where there is a number of observations k and with p variables, the covariance matrix S given X ($S = \text{cov}(X)$) is

$$S = \frac{X^T X}{k-1}, \tag{1}$$

The eigenvectors of this matrix are obtained by solving

$$S\mathbf{v} = \lambda \mathbf{v}, \tag{2}$$

where $\mathbf{v}$ is an eigenvector and λ is the eigenvalue associated with that eigenvector. The projection of the observations X on the latent space given by the subspace of the n principal components (in this case $n = 3$) can be expressed as follows

$$\mathbf{X}_L = X\mathbf{W}, \tag{3}$$

where $\mathbf{W}$ is the matrix of the n selected eigenvectors, and each column corresponds to an eigenvector (ordered from higher to lower eigenvalue) [4].

The second Manifold Learning model is UMAP, a dimensionality reduction method, but in this case representing a nonlinear mapping where the objective is to maintain the local structure of the input data samples in a low-dimensional latent space. Given the dataset X, the method builds a neighborhood graph, where each observation is connected to its nearest neighbors according to a measure of distance between data samples. Then, an optimization algorithm calculates the low-dimensional projected vectors, grouped in matrix Y, based on the minimization of an objective cost function given by

$$C = \sum_{i \neq j} v_{ij} \log\left(\frac{v_{ij}}{w_{ij}}\right) + (1 - v_{ij}) \log\left(\frac{1 - v_{ij}}{1 - w_{ij}}\right) \tag{4}$$

where w_{ij} are the similarities between the pairs of points within the output space Y used the t-Student distribution with one degree of freedom based on the Euclidean distance, and they are defined as

$$w_{ij} = \left(1 + \|y_i - y_j\|_2^2\right)^{-1}, \tag{5}$$

and variable v_{ij} is the Gaussian pair similarity but with respect to the Euclidean distance in the space of X, i.e., the distance between x_i and x_j, defined as follows

$$v_{ij} = \exp\left(-\|x_i - x_j\|_2^2 / 2\sigma_i^2\right). \tag{6}$$

The cost function is optimized by stochastic gradient descent which iteratively updates the Y projection in terms of the function gradient at each point [5].

B. Models of Supervised Learning Classification

We used the following battery of ML classifiers.

(1) The Logistic Regression (LR) Model is a binary classifier based on the logistic function (also known as sigmoid) where the input is the linear combination of the features and returns a probability between or and 1. Specifically, the implemented model is based on the Sklearn library with the established hyperparameters of the canceled penalty and using the Newton-Cholesky resolution, being this model the optimal one used in the grid. The solvers used in the grid are the Newton-Cholesky model itself, the Broyden-Fletcher-Goldfarb-Shannon based optimization model, the solver for sparse logistic regression, and the stochastic mean gradient descent [6], [7].

(2) The LR Least Absolute Shrinkage and Selection Operator (LASSO) Model is a variant of logistic regression that uses the $L1$ penalty within the cost function to penalize the coefficients of the binary classification model. This type of penalty is used to identify the most important variables in the classification model. In this work, the hyperparameter was used with the coordinate descent algorithm being the optimal option within the grid [6], [8].

(3) The LR Ridge Model is a variant of the first LR model, but with the $L2$ penalty where the objective is the minimization of the Ridge LR cost function by estimating coefficients maximizing the likelihood of the data matrix subject to the sum of squares constraint. This means that all variables contribute to the model prediction, but their effect is reduced proportionally to their magnitude [9]. The hyperparameter used in this model is the solver based on the Broyden-Fletcher-Goldfarb-Shannon algorithm being the optimal one used in the grid based on the performance metrics of the model [6].

(4) The LR ElasticNet Model employs the LR variant using the combination of both l1 and l2 penalties within the cost function to regularize the model and avoid overfitting. The combination of both penalties allows to control the trade-off between feature selection which makes the ElasticNet classification model useful in problems where it is required to select relevant features and avoid overfitting [10].

(5) The Decision Tree Model is based on a decision tree in order to train a model that can predict the class to which a sample belongs based on the characteristics. The model belongs to the Scikit-learn library where the hyperparameters have been adjusted, where the chosen in the grid is the Shannon entropy criterion in order to divide the nodes of the tree that minimizes the log loss [6], [11].

(6) The Random Forest Model is a variant of the previous Decision-Tree Classifier supervised algorithm, but in this case multiple trees are combined and a prediction is generated based on the majority of the votes of the individual trees. The process of building this model starts with the random selection of a sample of the k-fold training data. Subsequently, a decision tree is constructed for that sample using the Random Subspace method that selects a random subset of features for each tree [12]. The optimal hyperparameter analogously to the previous model has been fitted to Shannon's entropy method for the decision criterion of random trees [6].

(7) The Support Vector Classification (SVC) Model uses support vector machines (SVM) to separate two classes. This algorithm uses a separating hyperplane which is optimal for maximizing the distance between the nearest data samples of the two classes, i.e., the maximum margin or separability between them. Specifically, the classification model used has been fitted with a linear kernel, this being the optimum in the grid selections with the given metrics [6], [13].

(8) The XGBoost Model is a variant of the boosting algorithm that uses decision trees as a base estimator. Specifically, the hyperparameter used is the default one, since in the grid selection there is the option based on decision trees and the linear version, but the most optimal in this problem is the one based on decision trees. In each iteration of the classification model, a new decision tree is fitted to the residuals of the previous model [6], [14].

(9) The Gradient Boost Classifier Model also uses boosting techniques, but it combines several weak models (generally decision trees, although other options exist). Specifically, the model selected by the grid is by means of the logarithmic loss hyperparameter at a value of 0.01, which implies that the minimization of the loss function is optimized to improve the performance of the model. Another of the adjusted hyperparameters is the squared error option, where it adjusts the quadratic loss function to evaluate the quality of the classification model [6], [15].

C. 5-fold Validation Dataset and Metrics

In this work we used the K-fold cross-validation technique, with a value of $k = 5$, to obtain a more robust estimate of the performance of each of the proposed classification models. In this case the data set is divided into 5 equal parts (called folds), where in each iteration one of the folds is used as the test set and the remaining 4 folds as the training set, recurrently in 5 iterations of the loop. This cross-validation technique is widely accepted and it is considered as a practice for evaluating model performance, since its advantages include reducing the risk of overfitting, identifying generalization problems, and providing a more accurate estimate of model performance (by averaging evaluation metrics) [16].

The following metrics were used for validation: Accuracy (ACC) is a metric that measures the proportion of correct predictions made by the model out of the total predictions; Recall is the metric also known as sensitivity or true positive rate commonly used in binary classification to evaluate the performance of the model in correctly identifying positive samples [17]; Area Under the Cruve (AUC) is normally used in binary classification as it represents the value of the area under the ROC (Receiver Operating Characteristic) curve, which is the two-dimensional representation of the true positive rate relative to the false positive rate for different decision thresholds. The AUC value varies between 0 and 1 where a value of 0.5 indicates a random model performance (analogous to iterating a coin flip in prediction). This metric represents an advantage over ACC or recall, as it provides an evaluation of the model globally, regardless of the cutoff point [18].

D. Dataset Description

The database used in this work was created thanks to the Innovation Oncology Laboratory of the Gynecological, Genitourinary, and Skin Cancer Department at Clara Campal Comprehensive Cancer Center (Hospital HM Sanchinarro, España). This institution has been tracking biomarkers, particularly impacting healthcare since 2013. The requirements for inclusion are based primarily on age (considering those over 18 years of age) as the disease state. The collection of these patients passed filtering, where 54 were molecularly characterized by next-generation sequencing, which was applied in two phases: one of the phases is by whole exome sequencing (WES), and the second option used predesigned gene panels (Onco80).

It should be noted that the target training datasets will focus on five: (1) where the first is a clinical dataset, (2) a dataset containing a preprocessing with the coding of the genomes of each patient (in an accounted form for each patient), (3) the coding of the genomes of each patient with the assigned genome position in each patient, (4) the combination of the clinical data and the first mentioned coding of the genomes for each patient and (5) the combination of the clinical data with the second type of coding of the genomes given the position. The clinical dataset contains the row-wise information of the 54 patients where the fields of BRCA status, age at diagnosis, histology (with the first principal component), stage, type of primary surgery, whether the patient has received interval surgery, type of interval surgery, adjuvant, and finally, disease progression (corresponding to the binary classification label into platinum sensitive and platinum resistant) are included.

The first input dataset for the models consists of the patient clinical data (CD), where the categorical variables go through the one-hot encoding preprocessing process to facilitate model training and coding. The second dataset (GD1) corresponding to genomic data requires further preprocessing. The initial data is based on the set of genes of each patient, wherein each row appear the information of the genes of the same patient and with a different extension for each patient. In this case, the preprocessing is based on counting the genes considering each of the patients iteratively in order to count by columns the genes in each row of patients, where in the case that the gene does not appear in that patient's mutation, it is counted as null. The third dataset (defined as GD2) consists of the second encoding of the genome data, where the genome count is considered with the addition of the genome position type restriction. Therefore, the accounting of each type of gene is filtered with each patient and given that type of position in the gene. The fourth training dataset (CGD1) consists of the aggregation of the CD with the data from the first type of gene coding (GD1). However, it is necessary to homogenize the information in the two datasets, as is the case of dimensionality and the integration of the same set of patients in the two datasets. Starting from the original set of 54 patients, in each of the data sets, there are some discrepancies with the patient data where patients appear in one set that only exists in others.

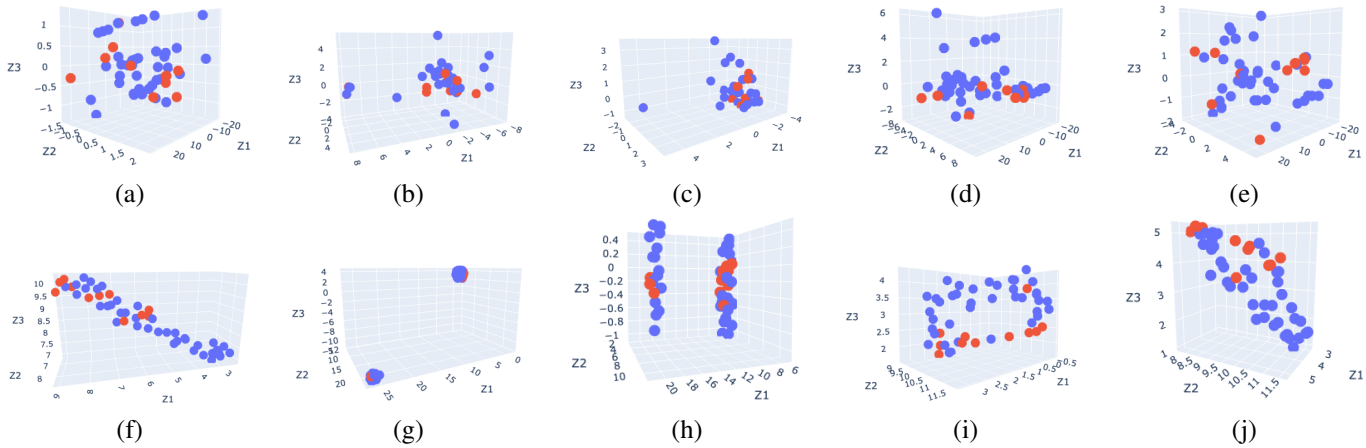

Fig. 1. Latent spaces. The first row corresponds to the PCA model with the CD database (a), the GD1 database (b), the GD2 data (c), the CGD1 data (d), and the CGD2 data (e). The second row corresponds to the UMAP model with the CD (f), the GS1 data (g), the GD2 data (h), the CGD1 data (i), and the CGD2 data (j).

In the case of the CD, three patients appear not included in the genomic data set. In the case of the genomic data set, one patient appears that is not recorded in the CD. Therefore, the data set comprises 51 patients combined with the two datasets. The fifth dataset (CGD2) combines the CD data with the GD2 data, i.e., analogous to the previous dataset, but with the aggregation of the second type of gene coding.

III. Experiments

This section is divided into two main parts, where the first one is based on the study of the IML models in each of the data sets and observing the separability of the classes in each of the projections of the samples in the latent space. The second part focuses on binary classification using nine different models (the ones discussed in Subsection II-B) and comparing the performance of each model by comparing the accuracy, recall, and AUC metrics on the five datasets.

A. Manifold Model Study

Figure 1 shows the set of experiments of the two IML models (the linear PCA model and the nonlinear UMAP model) with the five datasets. In the PCA model, it is observed that with the CD (in *Figure 1* (a)), there is no separability between classes. In contrast, with the gene data (GD1 and GD2), a group of samples appears in the latent space (in *Figure 1* (b) and (c)), but with little separability. On the other hand, with the combination of both data (CGD1 and CGD2) in *Figure 1* (d) and (e), it improves with few changes. However, it is observed that the resistant platinum class (in red color) differs outside the latent space concerning the sensitive platinum class (in blue color).

In the case of the nonlinear model, the behavior of the projections in the latent space changes. In the case of the CD (in *Figure 1* (f)), it is observed that the sensitive platinum class (blue color) appears with high separability in the right zone (the positive infinity of the Z1 axis) of the subfigure and the resistant platinum class (red color) appears in the left zone (the negative infinity of the Z1 axis). However, there are areas of the latent space where both classes converge. With the GD1 (in *Figure 1* (g)), it is observed that the UMAP model generates two groups, but where both classes exist in each group, therefore the classes do not differ in the first gene coding. *Figure 1* (h) shows that, analogously to the previous one, two groups of samples are formed in the latent space, but both classes appear in each group; however, this type of coding considerably improves the interpretability of the samples in the latent space. In the case of the combination of the data, in the CGD1 (in the figure), it is observed that two sets of manifolds appear, where in the zone towards the positive infinity of the Z2 axis, the sensitive platinum class is grouped (blue color) and in the zone towards the negative infinity of the Z2 axis the resistant platinum class is grouped (red color). In the case of the second combination (CGD2), it is observed that the grouping of the manifold changes slightly and, above all, denotes that the classes are more clustered and in a linear trend. In the case of the sensitive platinum class, many of the projections are grouped. In the resistant platinum class (red color), a large part is grouped towards the positive infinity of the Z3 axis, and some samples flow in the intermediate zones with the sensitive platinum classes (with little separability).

B. Evaluation of Binary Classification Models

In the classification models, the chosen metrics are the accuracy, recall, and AUC values obtained by averaging the values stored in each k-fold iteration for the five partitions of the test data as listed in *Table I*. The results denote the conclusions obtained with the manifolds, where the datasets with clinical and genomic data combinations offer better performance in the classification model. Specifically, using the CGD1, two models stand out for their performance: the LASSO LR model and the Decision Tree model. The latter stands out above all for having a better AUC score. However, due to the final application of this model, which is the classification of the

TABLE I
RESULTS OF THE ACCURACY, RECALL, AND AUC METRICS WITH THE CLASSIFICATION MODELS USED THE 5-FOLD VALIDATION METHOD WITH THE FIVE DATASETS.

	Dataset CD			Dataset GD1			Dataset GD2			Dataset CGD1			Dataset CGD2		
Model	ACC	Recall	AUC	ACC	Recall	AUC	ACC	Recall	AUC	ACC	Recall	AUC	ACC	Recall	AUC
LR	0.76	0.84	0.67	0.72	0.88	0.52	0.78	0.95	0.56	0.82	0.87	0.74	0.88	0.94	0.80
LR LASSO	0.78	1.0	0.5	0.78	0.95	0.56	0.74	0.95	0.47	**0.88**	**0.91**	**0.80**	**0.88**	**0.91**	**0.82**
LR Ridge	0.82	0.97	0.60	0.82	1.0	0.58	0.78	0.97	0.53	0.86	0.88	0.80	0.88	0.94	0.80
LR ElasticNet	0.78	1.0	0.5	0.78	1.0	0.50	0.78	1.0	0.50	0.78	1.0	0.50	0.82	0.94	0.68
Decision Tree	0.78	0.81	0.74	0.67	0.80	0.45	0.68	0.81	0.51	**0.86**	**0.88**	**0.81**	0.84	0.88	0.79
Random Forest	0.78	0.89	0.66	0.76	0.97	0.48	0.78	1.0	0.50	0.86	0.94	0.77	0.86	0.94	0.77
SVC	0.78	1.0	0.5	0.78	1.0	0.50	0.78	1.0	0.50	0.84	0.88	0.77	**0.86**	**0.89**	**0.83**
XGBoost	0.80	0.94	0.65	0.74	0.95	0.47	0.76	0.95	0.52	0.86	0.94	0.77	0.84	0.91	0.75
GBC	0.78	0.86	0.61	0.74	0.93	0.49	0.76	0.97	0.48	0.84	0.89	0.79	0.86	0.91	0.79

type of platinum used in the progression of OC disease, and because the cost of false positive rate scores is critical, the recall score is essential. Therefore, in this case, the LASSO LR model for this particular health application is the most appropriate considering the equivalence of scores on the other metrics. In the case of the second CGD2 combination, models with better performance than in the previous case are obtained, where the SVC model has the best AUC value. However, again the model that seems to have the best performance is the LR LASSO, although it should be noted that the LR model and the LR with Ridge optimization have high Recall values with the same accuracy score. However, the LASSO model has a better AUC value, which generally denotes better performance. In addition, these two previous models do not stand out in the CGD1, which denotes that the performance of these models does not have a robust behavior.

IV. CONCLUSION

The IML model experiments show that the nonlinear UMAP model performs better in the generalized form on each dataset than the linear model based on PCA dimensionality reduction. As for the UMAP model, there are differences between each of the five datasets. The CD has some separability, and the genomic data with the second type of combination (GD2) offers a higher interpretability of the samples in the class projections. The CGD2 (the combination of the two) offers a higher linear combination than the first combination (CGD1). Regarding the classification model, the best-performing model is the LR LASSO model considering the CGD2 considering the critical criterion of recall metric due to the high cost of false positive rate within the prognosis of OC disease.

ACKNOWLEDGMENT

This work was partially supported by research Grant PID2019-104356RB-C42, -C43, and PID2019-106623RB-C41 funded by MCIN / AEI /10.13039/501100011033 and by Andres Poveda GEICO Translational Research Grant 2021.

REFERENCES

[1] M. Moschetta, A. George, S. Kaye, and S. Banerjee, "Brca somatic mutations and epigenetic brca modifications in serous ovarian cancer," *Annals of Oncology*, vol. 27, no. 8, pp. 1449–1455, 2016.

[2] T. Rutherford, J. Orr Jr, E. Grendys Jr, R. Edwards, T. C. Krivak, R. Holloway, R. G. Moore, L. Puls, T. Tillmanns, J. C. Schink, *et al.*, "A prospective study evaluating the clinical relevance of a chemoresponse assay for treatment of patients with persistent or recurrent ovarian cancer," *Gynecologic Oncology*, vol. 131, no. 2, pp. 362–367, 2013.

[3] K.-H. Yu, V. Hu, F. Wang, U. A. Matulonis, G. L. Mutter, J. A. Golden, and I. S. Kohane, "Deciphering serous ovarian carcinoma histopathology and platinum response by convolutional neural networks," *BMC medicine*, vol. 18, no. 1, pp. 1–14, 2020.

[4] I. T. Jolliffe, *Principal component analysis for special types of data*. Springer, 2002.

[5] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7] N. Simon, J. Friedman, and T. Hastie, "A blockwise descent algorithm for group-penalized multiresponse and multinomial regression," *arXiv preprint arXiv:1311.6529*, 2013.

[8] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.

[9] S. L. Cessie and J. V. Houwelingen, "Ridge estimators in logistic regression," *Journal of the Royal Statistical Society Series C: Applied Statistics*, vol. 41, no. 1, pp. 191–201, 1992.

[10] Q. Li and N. Lin, "The bayesian elastic net," 2010.

[11] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees. wadsworth int," *Group*, vol. 37, no. 15, pp. 237–251, 1984.

[12] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[13] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[14] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[15] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[16] J. G. Moreno-Torres, J. A. Sáez, and F. Herrera, "Study on the impact of partition-induced dataset shift on k-fold cross-validation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1304–1312, 2012.

[17] J. Lever, "Classification evaluation: It is important to understand both what a classification metric expresses and what it hides," *Nature methods*, vol. 13, no. 8, pp. 603–605, 2016.

[18] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, 2006.