Estimation of vital status of patients with ovarian cancer using Machine Learning models

Kontilenia Maria Kotsifakou
National Technical University of Athens
Data Science and Machine Learning
03400174
kontileniakotsifakou@mail.ntua.gr

***Abstract*—The objective of this paper is to test different Machine Learning models in order to accurately predict the vital status of patients with high grade serous ovarian cancer. The dataset used contains information about 488 patients for 11 ovarian cancer related attributes and their vital status, while the trained models implement the classifiers K-Nearest-Neighbors, Support Vector Machine, Logistic Regression, Random Forest and XGBoost. The methodologies used are K-Nearest-Neighbors for filling the missing values, PCA and variance threshold for attribute selection, Min-Max scaling and Z-Score for normalization, 5-fold Cross Validation for the validation of the models and Grid Search for hyperparameter selection. The performance of the models is evaluated using the metrics accuracy and Area Under the Curve (AUC) while precision, recall, F1-Score were merely examined. The best classifier regarding the accuracy is Logistic Regression with the score of 77.3%, and regarding the AUC is XGBoost with the score of 73.47%.**

***Index Terms*—Machine Learning, Ovarian cancer**

I. Introduction

Cancer is one of the deadliest diseases in the world [6]. The aforementioned study is focused on ovarian cancer, which is ranked fifth in cancer deaths among women, while it is estimated that approximately 1 in 78 women are at risk of developing ovarian cancer at some point in their lives [1]. The term "silent" is commonly used to describe ovarian cancer since a significant proportion of patients are diagnosed with high grade serous ovarian cancer at a very advanced stage [12]. The basic treatment for ovarian cancer is cytoreduction surgery followed by chemotherapy with the use of platinum [5]. The seriousness of the disease along with the high mortality rate have led to extensive research regarding the factors that play a significant role in the cancer's outcome [1].

Initially Statistical Analysis was the primary method for analyzing ovarian cancer related data with numerous studies having been conducted [11] [3] [5]. Methods of Statistical Analysis, which most commonly include Cox regression or proportional hazards regression, were used for predicting important attributes of ovarian cancer such as the patient's survival and the platinum sensitivity. Some instances include Teramukai and al. [11] who created a model with the use of Cox regression that predicts the Overall Survival of patients with advanced epithelial ovarian cancer and Gerestein and al. [3] who created a similar model that additionally to the Overall Survival predicts the Progression Free Survival.

However, the use of Machine Learning during recent years leads to better and more accurate predictions as tested in several research papers [7]. For instance Paik and al. [7] predicted the survival outcome of patients with ovarian cancer with the use of the Cox regression and the Gradient Boosting classifier, which is a Machine Learning model, with the later approach proving significantly better results. Recently, numerous research papers have been conducted with the use of Machine Learning algorithms that replace the traditional Statistical Analysis and make more accurate predictions [4] [2] [8] [9] [10].

The purpose of this paper is to implement different machine models and predict the vital status of patients with ovarian cancer. The dataset used is part of the Cancer Genome Atlas project and more specifically contains information about 488 patients for 11 attributes regarding ovarian cancer (e.g. tumor stage) that form the vital status of them, which constitutes the class [12]. The classifiers implemented are K-Nearest Neighbors, Support Vector Machine, Logistic Regression, Random Forest and XGBoost. The output of the models is the predicted vital status of the patients, which is either Living or Deceased, while their performance is assessed with the most commonly used evaluation metrics for that particular task.

II. Related work

Numerous research papers have been conducted recently that implement Machine Learning models to predict important attributes of ovarian cancer outcome such as the Overall survival and the platinum sensitivity. To begin with, Hwangbo and al. [4] with the use of a very similar dataset implemented the classifiers Logistic Regression, Random Forest, Support Vector Machine and Deep Neural Networks in order to predict platinum sensitivity in patients with advanced ovarian cancer. In the research presented the Logistic Regression model performed the best [4]. Moreover, Arezzo and al. [2] with the use of a similar dataset implemented three Machine Learning algorithms that were Logistic Regression, Random Forests and K-Nearest Neighbors in order to predict the Progression Free Survival of patients with ovarian cancer. The best model in this case was Random Forest.

Furthermore, Sorayaie Azar and al. [10] implemented six machine algorithms in order to predict the survival of patients with ovarian cancer. More specifically the classifiers used were the K-Nearest Neighbors, Support Vector Machine, Decision Tree, Random Forest, Adaptive Boosting, and Extreme Gradient Boosting with the best classifiers being Random Forest and Extreme Gradient Boosting. In another recent approach, Piedimonte and al. [8] with the use of the Random Forest classifier created a model that predicts the outcome of the primary cytoreductive surgery, which achieved great performance.

Last but not least, it is important to mention the work of Sidey-Gibbons and al [9] that tested seven machine learning algorithms, including Logistic Regression, General additive model, Regression Trees, Gradient boosting trees, Multivariate adaptive regression splines, Support Vector Machines and Neural Networks, in order to predict if a woman with ovarian cancer will die in the next six months. Despite the fact that the dataset used in their analysis was structured differently including forms filled by patients, one of the best performing models was the Gradient boosting trees [9]. An early assumption that can be made is that the classifiers that include Trees or Random Forests perform better in predicting ovarian cancer's important attributes, which is something that is going to be tested in this study.

III. Dataset and Features

The dataset consists of 488 rows of data and 13 columns. The first column represents the BCR Patient Barcode, which is simply a unique ID for each patient and thus it is not taken into consideration in the analysis. The third column represents the Vital Status, which is the class label with the two distinct categories Living or Deceased. The remaining 11 columns represent features related to ovarian cancer. There is a more extensive analysis of those features in Table I, where additive information about the name, the domain of values, the type, the missing or null values and a brief description of them are illustrated.

In regard to preprocessing of the dataset, a lot of methods and techniques are used and tested. To begin with, the categorical variables are transformed to integer (one hot) encoding in order to be handled more efficiently by Machine Learning algorithms. A similar approach is followed for ordinal attributes, with the difference that the order is preserved. The appropriate mapping is made using the LabelEncoder function from sklearn.preprocessing library.

Furthermore, the dataset is balanced regarding the class labels and therefore oversampling or undersampling techniques are not performed. Regarding the numerous missing values of the dataset the KNNImputer function is used from the impute module of the sklearn library. KNNImputer replaces missing values present in the observations by finding the nearest neighbors with the Euclidean distance matrix. The rows that contain missing values in the class label are ignored.

Moreover, for the dataset's feature reduction two different approaches are tested: Feature Selection and Feature Extraction. The reduction of features is vital as the number of features is considered relatively high regarding the number of rows. Feature selection refers to the elimination of features based on criteria without transforming their values, while feature extraction refers to the transformation of features into new ones in a smaller space.

A feature selection technique that is tested is the variance threshold, where features with low variance are discarded from the dataset. More specifically, a variance upper threshold can be set and thus features that possess smaller than the threshold variance are removed. A feature extraction method that is tested is the Principal Component Analysis (PCA) where the features are analyzed into principal components and the training is done in completely new, linearly uncorrelated features of lower dimensionality (Figure 1).

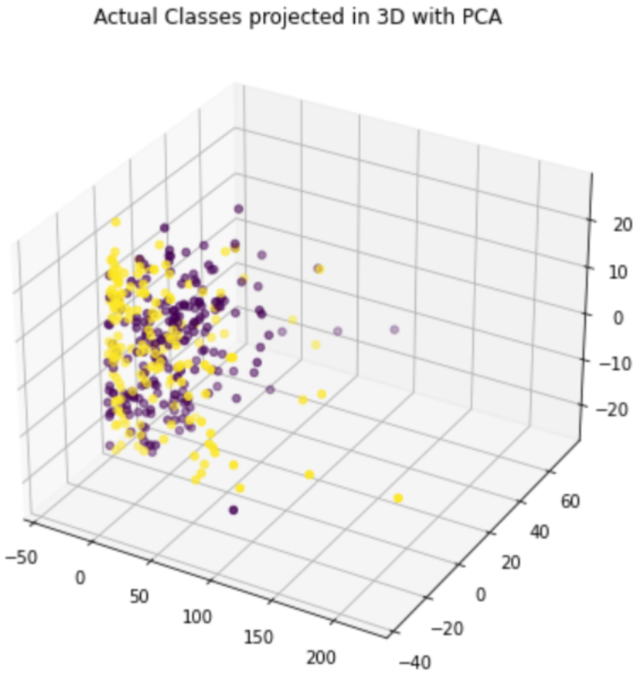

Fig. 1. Visualization of data using PCA in 3 dimensions

Last but not least, for the normalization of features the two widely used normalization techniques are tested, which are the Min-Max and the Z-Score. The Min-Max normalization scales the features linearly in the [0,1] interval by subtracting with the min value and dividing with the min-max difference while the Z-Score normalization normalizes the features to have a mean value of 0 and a standard deviation of 1, like the normal distribution (Equations 1, 2). Normalization techniques are implemented only as part of feature selection because at PCA the features are initially normalized with the Z-Score.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

$$z = \frac{X - \mu}{\sigma} \tag{2}$$

All the methods used for the replacement of missing values, feature selection and normalization of features are based on the most suitable techniques that are used in related research papers [7] [4] [2] [8] [9] [10].

TABLE I
REPRESENTATION ANALYSIS OF EVERY FEATURE

Name	**Domain of Values**	**Type**	**Null / Missing Values**	**Short description**
Vital Status	1.Living, 2.Deceased	Categorical	5	Class label
Age at Diagnosis	[30.5 - 87.47] μ = 59.8 σ = 11.48	Numerical (float)	11	Age of the initial cancer diagnosis
Tumor Stage	1.IIA, 2.IIB, 3.IIC 4.IIIA, 5.IIIB, 6.IIC, 7.IV	Ordinal	4	The Stage of Cancer (IIA $<$ IIB $<$ IIC $<$ IIIA $<$ IIIB $<$ IIC $<$ IV) Broader categories are stages: II, III and IV
Tumor Grade	1.G2, 2.G3	Ordinal	11	Aggressiveness of cancer (G2<G3)
Tumor Residual Disease	1.No Macroscopic Disease, 2.1-10 mm 3.11-20mm, 4.>20mm	Ordinal	56	The size of the residual disease after primary surgery
Primary Therapy Outcome success	1.Complete Response, 2.Partial Response 3.Stable Disease 4.Progressive Disease	Categorical	93	Response to treatment after primary surgery and chemotherapy
Person NeoPlasm Cancer Status	1.With tumor, 2.Tumor Free	Categorical	56	Last known status of disease
Overall Survival (OS)	[0.13 - 179.18] μ = 36.94 σ = 24.97	Numerical (Float)	6	Interval from primary surgery to the date of last known contact or death
Progression Free Status	1.Recurred/Progressed, 2.Disease Free	Categorical	2	Progression: spread of cancer Recurrence: return of cancer after a time period
Progression Free Survival (PFS)	[0.3 - 179.18] μ = 18.23 σ = 16.4	Numerical (Float)	93	Interval from primary surgery to the date of progression or recurrence or last known contact (not recurred)
Platinum Free Interval	[-3.7 - 106.9] μ = 13.09 σ = 17.02	Numerical (Float)	141	Interval from last platinum treatment to the date of progression or recurrence or last known contact (not recurred)
Platinum Status	1.Sensitive 2.Resistant 3.Too early	Categorical	143	Cancer resistance to platinum treatment (too early to judge in some cases)

IV. METHODS

A. Grid Search - Pipelines in machine learning

The majority of machine learning models contain parameters that can be adjusted to vary how the model learns. Hyperparameter tuning is a process using pipelines and grid search in machine learning that automate the search of best fitted parameter values in each model.

A machine learning pipeline is the end-to-end construct that orchestrates the flow of data into, and output from, a machine learning model (or set of multiple models). It includes raw data input, features, outputs, the machine learning model and model parameters, and prediction outputs.

For example, the logistic regression model, from sklearn, has a parameter C that controls regularization,which affects the complexity of the model. One method is to try out different values and then pick the value that gives the best score. This technique is known as a grid search. If there is a list of possible values to select for two or more parameters, all combinations of the sets of values would be evaluated thus forming a grid of values.

For each hyperparameter value of the grid, the average of the estimator should be calculated in all folds of the cross-validation based on the selected metric, for example F1, and the best combination of parameters should be selected. The specific search strategy for the optimal hyperparameters is the exhaustive grid search and is obviously very computationally expensive.

Briefly, hyperparameter optimization requires:

- an estimator (a classifier)
- the scope of the hyperparameters
- a way to search for their possible value combinations (grid search)
- a cross-validation scheme (5-fold as suggested in related work)
- a performance metric (or score) (accuracy or AUC as suggested in related work)

In the specific analysis, five pipelines and grid search are developed, each one corresponding to a classifier model and its parameters and hyperparameters.

B. Classifiers

There are two kinds of classifiers the parametric ones and the non parametric ones. In general parametric classifiers are simpler, faster in train/test phases and need less training data. On the other hand, they generally have a smaller capacity, i.e. they can separate the classes into problems of relatively smaller dimensions, while the requirement of the real data to follow an exact distribution is very strong and not practically verified. Conversely, non-parametric classifiers are slower to train, generally have larger space/memory requirements and need more data but have more capacity, can learn harder problems, and perform better on larger datasets.

According to related work, K-nearest neighbors (k-NN), Support Vector Machines (SVM), Random Forest (RF), Logistic Regression and Boosting algorithms are tested in similar tasks and datasets.

k-NN belongs to non parametric algorithms. Its principle of operation its the following. For a new sample to be classified, its k nearest neighbors is/are computed (in the n-dimensional input feature space) based on some distance function, usually Euclidean distance 2.

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \ldots + (x_n - x'_n)^2}$$

Fig. 2. Euclidean Distance

The class of the new sample will be the class of the majority of k neighbors (we choose odd k in general), either simply calculated (sum) or (inversely) weighted by each neighbor's distance.

K-NN has practically no training phase. However, to classify a new sample in the test phase, its distance has to be compared to each sample in the train set. This means that all training samples are necessary for classification. This means that k-NN is more demanding in both space (storing all samples) and time (computing all distances for each new sample).

The k variable of k-NN is a hyperparameter of the classifier. Another hyperparameter for example is the distance function. Hyperparameters are choices made by the system designer, and the optimal values are evaluated empirically on data.

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. However, it is more efficient for sorting. The aim of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends on the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2D plane.

The SVM kernel is a function that takes a low-dimensional input space and transforms it into a higher-dimensional space, in fact it transforms a non-separable problem into a separable problem. It is mainly useful in nonlinear separation problems. The kernel does some very complex data transformations and then figures out the process of separating the data based on the labels or outputs defined. There are three types of kernel: Linear, Rbf and Sigmoid.

- The linear kernel is used when the data is linearly separable (or nearly linearly separable), so it can be separated using a single line. It is one of the most common kernels used. It is mainly used when there are a large number of features in a particular data set.
- When the data set is non-linear, it is recommended to use kernel functions such as RBF. For a linearly separable dataset (linear dataset) one could use a linear kernel function (kernel = ”linear”). RBF kernels are the most generalized form of kernels and are one of the most widely used kernels due to its similarity to the Gaussian distribution. The RBF kernel function for two points x1 and x2 calculates the similarity or how close they are to each other. This kernel can be represented mathematically as in Figure 3.

$$K(X_1, X_2) = e^{-\frac{||X_1 - X_2||^2}{2\sigma^2}}$$

Fig. 3. The RBF Kernel function for two points x1 and x2

- Sigmoid kernel is mainly preferred for neural networks. This kernel function is similar to a two-layer perceptron model of the neural network, which acts as an activation function for neurons.

Logistic regression is a fundamental classification technique. It belongs to the group of linear classifiers and is somewhat similar to polynomial and linear regression. It is fast and relatively uncomplicated. Although it's essentially a method for binary classification, it can also be applied to multiclass problems.

It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable, so the outcome must be a categorical or discrete value. Is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

In Logistic regression, instead of fitting a regression line, an ”S” shaped logistic function is fitted. This sigmoid function is used to map predictions and their probabilities. The sigmoid function converts any real value to a range between 0 and 1. Moreover, if the output of the sigmoid function (estimated probability) is greater than a predefined threshold on the graph, the model predicts that the instance belongs to that class. If the estimated probability is less than the predefined threshold, the model predicts that the instance does not belong to the class.

The sigmoid function is referred to as an activation function for logistic regression and is defined as in Figure 4, where e is the base of natural logarithms and value is the numerical value one wishes to transform.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Fig. 4. Sigmoid function

The Figure 5 represents logistic regression, where x is the input value, y is predicted output, b_0 is bias or intercept term and b1 is coefficient for input(x).

$$y = \frac{e^{(b_0 + b_1X)}}{1 + e^{(b_0 + b_1X)}}$$

Fig. 5. Logistic Regression

This equation is similar to linear regression, where the input values are combined linearly to predict an output value using weights or coefficient values. However, unlike linear regression, the output value modeled here is a binary value (0 or 1) rather than a numeric value.

Additionally, Random Forests is another supervised ensemble technique, using the combination of predictions of other classifiers; random trees. Every one of the decision tree classifiers is trained in a randomly selected subset of the training set, and then the votes from different decision trees are collected in order for the Random Forest Classifier to decide the final prediction. It seems like Random Forests pulling together the decision tree algorithm efforts. Taking the teamwork of many trees thus improving the performance of a single random tree. It can operate in classification and regression tasks. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.

The main challenge of Random Forest is to find the important features that radically define the root node and every level of the tree. The latter procedure is known as feature selection. Both in Decision Trees and therefore in Random Forests there are two popular measures: Gini index 7 and Information Gain based on Entropy 6. Hence, the feature selection method is one of the hyperparameters of the Random Forest.

$$Entropy = \sum_{i=1}^{n} -p(c_i)log_2(p(c_i))$$

Fig. 6. Entropy

$$Gini = 1 - \sum_{i=1}^{n} p^2(c_i)$$

Fig. 7. Gini Index

Moreover, one significant hyperparameter for Random Forest is the depth of the decision trees. In particular, trees that are grown very deep tend to learn highly irregular patterns: they overfit their training sets, i.e. have low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

One more recently involved classifier that is known to outperform Random Forests is Gradient Boosting. XGBoost is an open-source implementation of gradient-boosting decision trees. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for “Extreme Gradient Boosting” and it has become one of the most popular and widely used machine learning algorithms due to its ability to handle large datasets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression.

The advantageous characteristics making XGboosting ideal for the particular dataset of the current study are the following:

- Parallelized tree building: XGBoost approaches the process of sequential tree building using parallelized implementation.
- Tree pruning: XGBoost grows the tree up to max_{depth} and then prune backward until the improvement in loss function is below a threshold.
- Cache awareness and out of core computing: XGBoost has been designed to efficiently reduce computing time and allocate an optimal usage of memory resources.
- Regularization: The biggest advantage of XGBoost is regularization, allowing control overfitting by introducing L1/L2 penalties on the weights and biases of each tree.
- Handles missing value: This algorithm has important features of handling missing values by learning the best direction for missing values. The missing values are treated to combine a sparsity-aware split finding algorithm to handle different types of sparsity patterns in data.
- Built-in cross validation: The algorithm comes with a built-in cross validation method at each iteration, taking away the need to explicitly program this search and to specify the exact number of boosting iterations required in a single run.

V. Experiments, results and discussion

To start with, our code is available at [13], where all the following experiments and results are illustrated more extensively. As previously mentioned, all classifiers went through grid search for hyper-parameter optimization. Moreover, we try to optimize data prepossessing according to every one of the classifiers. Explicit details can be found on Tables II,III about best models on accuracy and AUC, respectively.

We ran all the experiments twice in order to optimize scaler. It seems that no conclusion can be made for finding the best of min-max and standard scaler as they were been chosen equally at almost every scenario. The results concerning K-nn Imputer for filling the missing values shows that $k = 16$ was slightly more popular option between best category of the models. Despite of the fact that only 4 PCA components can explain the 0.912 of total info of the dataset 8, the grid search illustrate a quite undeniable preference to 15 PCA components even between the two final best models. Similar preference seems to be found on variance threshold of feature selection, in the threshold value of 0.1.

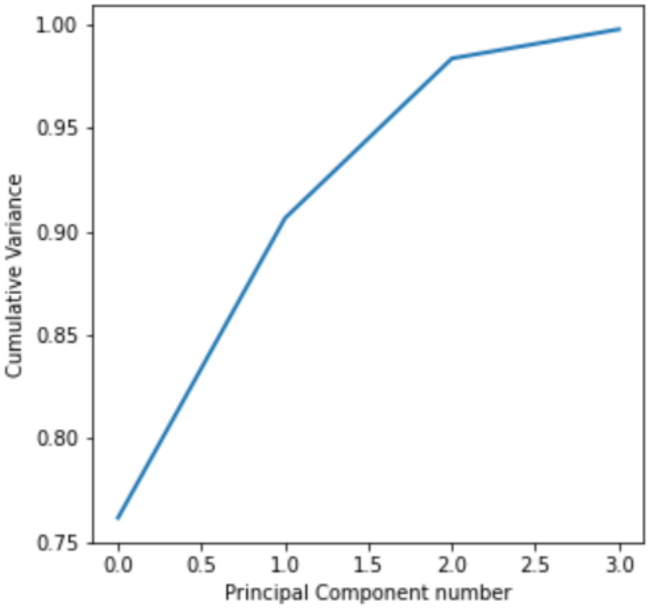

Fig. 8. Information gain from 4 PCA components

For every k-NN classifier, k hyperparameter varied between 4 and 19 which is a large fluctuation. The experiments allowed grid search to choose between 4,9,14,19. Nevertheless, this particular classifier seems to have less accuracy than the rest, but the second best AUC score. The hyperparameter of SVM was kernel (linear, polynomial and RBF). There was no preferences on that based on the results. It only worth to mention that min-max scaler was the optimal for that model.

Logistic Regression lyperparameters also varies in best models shown. C values were in $1.e-04, 1.e-02, 1.e+00, 1.e+02, 1.e+04$. It had the best accuracy score hence it is one of the suggested models for predicting vital status the particular dataset. The confusion matrix is presented in Figure 9. Random Forest Classifier had two hyperparameters as mentioned in previous sections. The best model based on AUC was XGBoost having 3 hyperparameters. The ROC curve is shown in Figure 10.

VI. Conclusion/Future Work

The best classifiers for predicting vital status of women with ovarian cancer are Logistic Regression based on accuracy and XBoost based on AUC. It is important to mention that the difficulties of the particular task were the numerous missing values and the bulk of the dataset. Having that little of data makes it way more possible for every model to overfit. XGBoost, solves the aforementioned problems in theory and our expiraments illustrates that this is true in our dataset too. XGBoost had the higher accuracy and the best AUC.

Further examination should be done, in order to compare more Gradient Boosting implementations. Another interesting idea is to pre-train a deep learning model in general and much larger dataset with similar type of cancers and try to fine-tune a model by freezing the pretrain model and using only a simple linear output layer for ovarian vital status prediction.

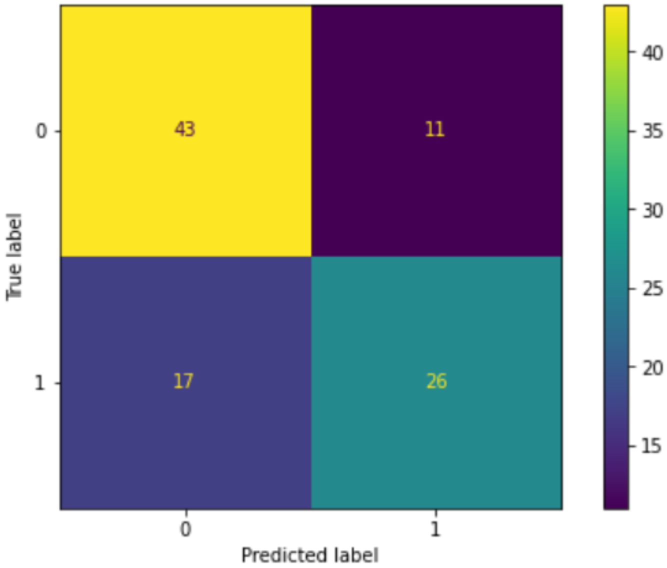

Fig. 9. Confussion matrix for Logistic Regression

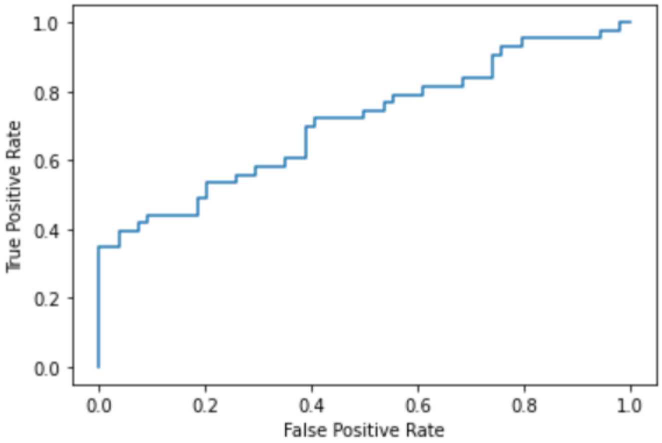

Fig. 10. ROC Curve for XGBoost

References

[1] American Cancer Society (2023) Key Statistics for Ovarian Cancer, Ovarian Cancer Statistics — How Common is Ovarian Cancer. Available at: https://www.cancer.org/cancer/ovarian-cancer/about/key-statistics.html

[2] Arezzo, F. et al. (2022) "A machine learning approach applied to gynecological ultrasound to predict progression-free survival in ovarian cancer patients," Archives of Gynecology and Obstetrics [Preprint]. Available at: https://doi.org/10.21203/rs.3.rs-1382403/v1.

[3] Gerestein, C.G. et al. (2009) "The prediction of progression-free and overall survival in women with an advanced stage of epithelial ovarian carcinoma," BJOG: An International Journal of Obstetrics & Gynaecology, 116(3), pp. 372–380. Available at: https://doi.org/10.1111/j.1471-0528.2008.02033.x.

[4] Hwangbo, S. et al. (2021) "Development of machine learning models to predict platinum sensitivity of high-grade serous ovarian carcinoma," Cancers, 13(8), p. 1875. Available at: https://doi.org/10.3390/cancers13081875.

TABLE II
BEST MODELS HYPERPARAMETERS FOR OPTIMIZING ACCURACY

Classifier	**Accuracy based % (scaler)**	**Parameters (based on accuracy)**
K-NN	70.1 (Min-max)	k: 4, k-imputer: 8, pca comp: 15,variance thres: 0.1
SVM	74.2 (Min-max)	kernel’: ’rbf’, k-imputer: 4, pca comp: 5, variance thres: 0.1
Logistic Regression	**77.31** (Zscore)	C: 1.0, penalty: ’l1’, k-imputer: 4, pca comp: 15, variance thres: 0
Random Forest	74.22 (Zscore)	criterion: ’entropy’, max depth: 7, k-imputer: 16, pca comp: 15, variance thres: 0.1
XBoost	75.25 (Zscore)	gamma: 0.5, max depth: 3, subsample: 1, k-imputer: 16, pca comp: 15, variance thres: 0

TABLE III
BEST MODELS HYPERPARAMETERS FOR OPTIMIZING AUC

Classifier	**AUC based % (scaler)**	**Parameters (based on AUC)**
K-NN	70.7 (Z-score)	k: 19, k-imputer: 12, pca comp: 5,variance thres: 0.1
SVM	70.9 (Min-max)	kernel’: linear’, k-imputer: 4, pca comp: 15, variance thres: 0.1
Logistic Regression	67.7 (Zscore)	C: 0.01, penalty: ’l2’, k-imputer: 12, pca comp: 15, variance thres: 0.1
Random Forest	67.22 (Min-Max)	criterion: ’entropy’, max depth’: 7, k-imputer: 16, pca comp: 15, variance thres: 0.1
XBoost	**73.49** (Min-Max)	gamma: 2, max depth’: 3, subsample: 0.8, k-imputer: 16, pca comp: 15, variance thres: 0

[5] Mankoo, P.K. et al. (2011) “Time to recurrence and survival in serous ovarian tumors predicted from integrated genomic profiles,” PLoS ONE, 6(11). Available at: https://doi.org/10.1371/journal.pone.0024709.

[6] Mlakar, I. et al. (2021) “Patients-centered survivorship care plan after cancer treatments based on big data and Artificial Intelligence Technologies (persist): A multicenter study protocol to evaluate efficacy of digital tools supporting cancer survivors,” BMC Medical Informatics and Decision Making, 21(1). Available at: https://doi.org/10.1186/s12911-021-01603-w.

[7] Paik, E.S. et al. (2019) “Prediction of survival outcomes in patients with epithelial ovarian cancer using machine learning methods,” Journal of Gynecologic Oncology, 30(4). Available at: https://doi.org/10.3802/jgo.2019.30.e65.

[8] Piedimonte, S. et al. (2022) “Using a machine learning algorithm to predict outcome of primary cytoreductive surgery in Advanced ovarian cancer,” Journal of Surgical Oncology, 127(3), pp. 465–472. Available at: https://doi.org/10.1002/jso.27137.

[9] Sidey-Gibbons, C.J. et al. (2022) “Predicting 180-day mortality for women with ovarian cancer using machine learning and patient-reported outcome data,” Scientific Reports, 12(1). Available at: https://doi.org/10.1038/s41598-022-22614-1.

[10] Sorayaie Azar, A. et al. (2022) “Application of machine learning techniques for predicting survival in ovarian cancer,” BMC Medical Informatics and Decision Making, 22(1). Available at: https://doi.org/10.1186/s12911-022-02087-y. .

[11] Teramukai, S. et al. (2007) “PIEPOC: A new prognostic index for Advanced Epithelial Ovarian Cancer—Japan multinational trial organization OC01-01,” Journal of Clinical Oncology, 25(22), pp. 3302–3306. Available at: https://doi.org/10.1200/jco.2007.11.0114

[12] The Cancer Genome Atlas Research Network (2011) “Integrated genomic analyses of ovarian carcinoma,” Nature, 474(7353), pp. 609–615. Available at: https://doi.org/10.1038/nature10166.

[13] https://github.com/Kontilenia/Machine-Learning.git