



# An AI-driven Predictive Model for Pancreatic Cancer Patients Using Extreme Gradient Boosting

Aditya Chakraborty<sup>1</sup> · Chris P. Tsokos<sup>2</sup>

Received: 23 August 2022 / Accepted: 24 August 2023 / Published online: 11 September 2023  
© The Author(s) 2023

## Abstract

Pancreatic cancer is one of the deadliest carcinogenic diseases affecting people all over the world. The majority of patients are usually detected at Stage III or Stage IV, and the chances of survival are very low once detected at the late stages. This study focuses on building an efficient data-driven analytical predictive model based on the associated risk factors and identifying the most contributing factors influencing the survival times of patients diagnosed with pancreatic cancer using the XGBoost (eXtreme Gradient Boosting) algorithm. The grid-search mechanism was implemented to compute the optimum values of the hyper-parameters of the analytical model by minimizing the root mean square error (RMSE). The optimum hyperparameters of the final analytical model were selected by comparing the values with 243 competing models. To check the validity of the model, we compared the model's performance with ten *deep neural network* models, grown sequentially with different activation functions and optimizers. We also constructed an ensemble model using Gradient Boosting Machine (GBM). The proposed XGBoost model outperformed all competing models we considered with regard to root mean square error (RMSE). After developing the model, the individual risk factors were ranked according to their individual contribution to the response predictions, which is extremely important for pancreatic research organizations to spend their resources on the risk factors causing/influencing the particular type of cancer. The three most influencing risk factors affecting the survival of pancreatic cancer patients were found to be the age of the patient, current BMI, and cigarette smoking years with contributing percentages of 35.5%, 24.3%, and 14.93%, respectively. The predictive model is approximately 96.42% accurate in predicting the survival times of the patients diagnosed with pancreatic cancer and performs excellently on test data. The analytical methodology of developing the model can be utilized for prediction purposes. It can be utilized to predict the time to death related to a specific type of cancer, given a set of numeric, and non-numeric features.

---

This innovation is accepted for a US provisional patent with application number 63/416,414.

---

Extended author information available on the last page of the article

**Keywords** Pancreatic Cancer · Extreme Gradient Boosting · Boosted Regression Trees · Pancreatic Risk Factors · Grid Search Mechanism

## Abbreviations

XGBoost	Extreme Gradient Boosting
GBM	Gradient Boosting Machine
MAE	Mean absolute error
RMSE	Root mean square error
AF	Activation Function
BN	Batch Normalization
DL	Deep Learning

## 1 Introduction

Pancreatic cancer continues to be one of the significant health hazards, and highly devastating gastrointestinal cancer affecting people all over the globe [25]. “Pancreatic cancer incidence rates are nearly similar to mortality rates due to high fatality rates” [35]. “According to the current health science researchers, this disease causes approximately 30,000 deaths per year in the USA” [31]. It is the fourth principal reason for cancer death in the USA and leads to an estimated 227,000 deaths per year worldwide. The incidence and number of fatalities from pancreatic tumors have been continuously increasing, while the incidence and mortality from other prevalent cancers have been decreasing. Despite advancements in pancreatic cancer detection and care, it is estimated that approximately 4% of patients will survive five years following diagnosis. [47]. After the detection of pancreatic cancer, doctors usually perform some additional tests to understand better if the cancer has been spread or the spreading area of cancer. Different imaging tests, such as a PET (Positron Emission Tomography) scan, have proven helpful to doctors in order to identify the presence of cancerous growths. With these tests, doctors try to establish the cancer’s stage. Staging helps explicate how advanced the cancer is. It also assists doctors in deciding the treatment options and alternatives. The following is the description of the stages used in our data set according to the definition of the Surveillance, Epidemiology, and End Results (SEER) database.<sup>1</sup>

- **Localized:** No evidence that the malignancy has spread beyond the pancreas.
- **Regional:** Cancer has spread to neighboring structures or lymph nodes from the pancreas.
- **Distant:** Cancer has spread to other regions of the body, including the lungs, liver, and bones.

<sup>1</sup> The paper is out of my doctoral dissertation chapter 3, page number: 62–83, and it is a part of innovation which was accepted for the US Provisional Patent (application number: 63416414. TTO ref. 22A113PR). The permission to publish by the patent authority can be provided on request.

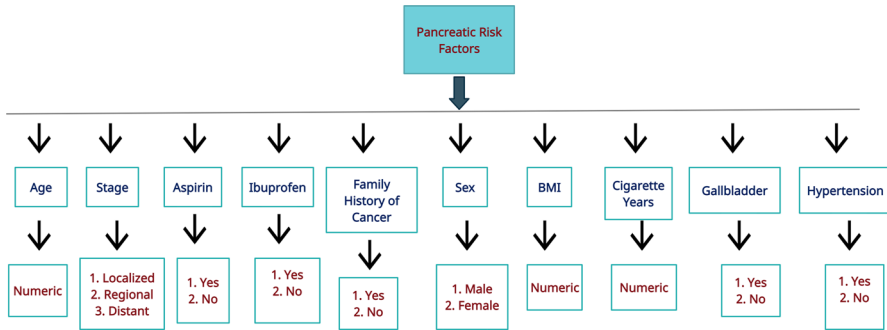
Risk factors for developing pancreatic cancer usually include family history, obesity, type 2 diabetes, and use of tobacco products. Even for the tiny proportion of patients who have a localized, resectable tumor, the prognosis remains poor, with only 20% surviving 5 years after surgery [37]. The outcome variable of our study is the survival time (in years). Although in most cases, pancreatic cancer remains incurable, researchers have concentrated on how to enhance the survival rates of individuals with pancreatic cancer. In our study, we developed a non-linear predictive model using Extreme Gradient Boosting (XGBoost) to estimate the survival time of patients diagnosed with pancreatic cancer. Given a set of risk factors (described in Section 2.1), our model predicts the survival of patients with a high degree of accuracy. We also compared our proposed model's accuracy (in terms of RMSE) with Gradient Boosting Machines (GBM) and different deep-learning models. In recent years, researchers are prone to using sophisticated data-driven machine-learning models, decision-making models, and deep-learning algorithms in applied research because of their high predictive power and learning abilities from data [9, 10, 16, 17, 27, 36, 44, 48]. There is an increased tendency in the studies published in recent years that applied semi-supervised ML techniques for modeling cancer survival which address both labeled and unlabeled data. [39]. Kourou, Exarchos, et al., 2015 [28] presented a detailed review of the most recent ML research methods applicable to cancer prediction/prognosis with case studies. Ahmad, Eshlaghy, et al., [2] used different ML and DL algorithms like Decision Tree (DT), Support Vector Machine (SVM), and Artificial Neural Network (ANN) and compared their performance to predict the recurrence of breast cancer using 10-fold cross-validation. Hayward, Alvarez, et al., [23] developed different predictive models for the clinical performance of pancreatic cancer patients based on machine learning methods. The predictive performance of machine learning (ML) is compared with linear and logistic regression techniques. According to their study, ML offers techniques for improved prediction of clinical performance, and thus, these techniques can be considered as valuable alternatives to the conventional multivariate regression methods in clinical research. Wang & Yoon [32] suggested an online gradient boosting (GAOGB) model based on a genetic algorithm for incremental breast cancer (BC) prognosis. Their proposed GAOGB model was evaluated on the SEER database in terms of accuracy, the area under the curve (AUC), sensitivity, specificity, retraining time, and variation at each iteration. Ma, Meng, et al., [33] suggested a classification model that uses the power of extreme gradient boosting (XGBoost) in complicated multi-omics data to focus on early-stage and late-stage malignancies separately. Their XGBoost model was applied to four types of cancer data downloaded from The Cancer Genome Atlas (CGA), and the model's performance was compared with other popular machine learning methods (ML) methods. The authors investigated the efficacy of XGBoost on the diagnostic categorization of malignancies in their study and found XGBoost as a robust predictive algorithm. Chen, Jia, et al., [15] proposed a non-parametric model for survival analysis that utilizes an ensemble

of regression trees to determine the variation of hazard functions with respect to the associated risk factors. The scientists used GBMCI (gradient boosting machine for concordance index) software to develop their model and tested its effectiveness against other conventional survival models using a large-scale breast cancer prognostic dataset. In their study, they found the GBMCI to be consistently outperforming other methods based on a number of covariate settings. Amjad, Maaz, et al. used the XGBoost algorithm to predict the pile-bearing capacity values and obtained the highest performance capability when compared with the other competing models like AdaBoost, Random Forest, decision tree, and Support Vector Machine [3]. In a study conducted in Beijing, the researchers proposed a hypertensive outcome prediction model combining the gain sequence forward tabu search feature selection (GSFTS-FS) and XGBoost by utilizing the data from patients with hypertension and obtained an accuracy of 94.6% with AUC of 0.956 [11]. Shi, Xiupeng, et al. used XGBoost for driving assessment and risk prediction and Key feature selection were done by gain-based importance ranking and recursive elimination [42]. Yang, Jian, et al. proposed a SMOTE-based Xgboost methodology for heart disease prediction and obtained 93.44% prediction accuracy. The relative importance of the features was accessed using the information gain [50]. Li, Hua, et al. applied the XGBoost algorithm to the personal credit evaluation problem based on big data and obtained the highest model performance based on four model evaluation matrices (Accuracy, Kappa, AUC, and KS) when compared with four popular machine learning models (logistic regression, decision tree, random forest, and Gradient Boosting Decision Tree). [30]

## 2 Materials and Methods

### 2.1 Data Description

The study data has been obtained from National Cancer Institute (NIH). The data contains information on patients diagnosed with pancreatic adenocarcinoma. We treated the survival time (in days) as the response in developing our model and considered cause-specific death (deaths due to pancreatic cancer) for each patient. Patient survival time is one of the most crucial factors in all cancer studies. It is critical to assess the severity of cancer since it helps to determine the prognosis and find the best treatment options. There were a total of 800 patients' information in our study after eliminating the missing observations for which several risk factors were missing. In our study, the response variable is the survival time of patients (in days). There are a total of *ten* risk factors used in our predictive analysis. Seven of those are categorical in nature, and three of them are numeric variables. The descriptions of the risk factors are as follows.



**Fig. 1** Pancreatic Cancer Data with Relevant Risk Factors

1. `panc_exitage` (Numeric) ( $X_1$ ): Age of diagnosis of the patient.
2. `Stage` (Categorical) ( $X_2$ ): Pancreatic Cancer Stages, categorized as a) localized, b) regional, and c) distant
3. `asp` (Categorical) ( $X_3$ ): Does the person use Aspirin Regularly?
4. `ibup` (Categorical) ( $X_4$ ): Does the person use Ibuprofen Regularly?
5. `fh_Cancer` (Categorical) ( $X_5$ ): The number of first-degree relatives with any type of cancer.
6. `Sex` (Categorical) ( $X_6$ ): Sex of the individual.
7. `BMI` (numeric) ( $X_7$ ): Current Body Mass Index (BMI) at Baseline (In lb/in2)
8. `Cigarette Years` (numeric) ( $X_8$ ): The total number of years the patient smoked.
9. `gallblad_f` (Categorical) ( $X_9$ ): Did the individual ever have gallbladder stones or inflammation?
10. `hyperten_f` (Categorical) ( $X_{10}$ ): Did the individual ever have high blood pressure?

A schematic diagram of the data used in our study with the description of risk factors is shown in Fig. 1 below.

As the above Fig. 1 illustrates, seven out of ten risk factors are categorical, having two or more categories. Before starting our analysis of the data, one important question is if there is any statistically significant difference between the survival times of male and female patients diagnosed with pancreatic cancer. To answer this question, we used the non-parametric Wilcoxon rank-sum test with continuity correction and obtained a p-value of .47, which suggests that there is no statistically significant difference between the true median survival times of patients from both genders at 5% level of significance [8]. Therefore, we performed our analysis by combining the information of males and females.

### 3 A Brief Overview of Gradient Boosting Machine (GBM) and Extreme Gradient Boosting (XGBoost)

In the literature of machine learning, “Boosting” is a collection of algorithms that transforms the ensemble of weak learners to strong learners iteratively. Boosting is an ensemble method for improving the model predictions of any given learning algorithm. Gradient boosting machines (GBM), as introduced by Friedman (2001) [20], are a prominent family of machine-learning (ML) algorithms that have demonstrated significant success in a wide range of applied and experimental fields. They are highly customizable to the specific requirement of the application and can be implemented with respect to different loss functions. In this section, we will go through the theoretical notions of gradient boosting briefly [38].

Let us assume the problem of classical supervised learning problem where we have  $n$  risk factors  $X = (x_1, x_2, \dots, x_n)$  and  $y$  as a continuous response variable. Given the data, training of the model is performed by obtaining the optimal model parameters  $\theta$  that best fit the training data  $x_i$  and response  $y_i$ . To train the model, we define the following objective function to quantify how well the model fits the training data.

$$O(\theta) = L(\theta) + \rho(\theta) \quad (1)$$

where  $L(\theta) = \sum_{i=1}^n (y_i - \hat{y}_i)^2$  is the training loss (mean square error) function that measures the predictive power of our model is with respect to the training data.  $\rho(\theta)$  is the regularization term that helps to prevent model overfitting and controls the complexity of the model.

#### 3.1 Decision Tree Ensembles

In our study, we use boosted decision tree ensemble method to train our model. Boosting combines a learning algorithm in an additive manner to achieve a strong learner from many sequentially connected weak learners. A decision tree’s major goal is to partition the input space variables into similar rectangular sections using a tree-based rule system. Each tree split corresponds to an if-then rule applied to a single input variable. A decision tree’s structure naturally stores and represents the interactions between predictor variables (risk factors). The number of splits, or equivalently, the *interaction depth*, is typically used to parameterize these trees. It is also possible to have one of the variables split numerous times in a row. A tree stump is a special example of a decision tree with just one split (i.e., a tree with two terminal nodes). As a result, if one wishes to fit an additive model using tree base-learners, the tree stumps can be used. Small trees and tree stumps produce remarkably accurate results in many real-world applications [49].

### 3.2 Model Structure

Mathematically, we can write our analytical model in the form:

$$\hat{y} = \hat{f}(x) = \sum_{i=1}^K \hat{f}_i(x), \quad \hat{f}_i \in \mathcal{F} \tag{2}$$

where  $\mathcal{F}$  is the collection of all possible regression trees,  $K$  is the number of regression trees, and  $\hat{f}_i$  are the additive functions (additive trees) in  $\mathcal{F}$ .

$f(x) = w_{q(x)}(q : \mathbb{R}^m \longrightarrow \{1, 2, \dots, T\}, w \in \mathbb{R}^T)$ . Here,  $q$  indicates the tree structure that maps an input to the relevant leaf index at which it finishes up. The number of leaves in the tree is denoted by  $T$ . Individual regression trees accommodate a continuous score on each of its leaves.  $w_i$  represents the score on  $i^{th}$  leaf. The tree structures of  $\hat{f}_i$  are intractable to learn at once. Hence, we use the following additive strategy. Let  $\hat{y}_i^{(t)}$  be the predicted value of the  $i$ th observation at step  $t$ . Then,

$$\begin{aligned} \hat{y}_i^{(0)} &= 0 \\ \hat{y}_i^{(1)} &= \hat{f}_1(x_i) = \hat{y}_i^{(0)} + \hat{f}_1(x_i) \\ \hat{y}_i^{(2)} &= \hat{f}_1(x_i) + \hat{f}_2(x_i) = \hat{y}_i^{(1)} + \hat{f}_2(x_i) \\ &\vdots \\ \hat{y}_i^{(t)} &= \sum_{j=1}^t \hat{f}_j(x_i) = \hat{y}_i^{(t-1)} + \hat{f}_t(x_i). \end{aligned} \tag{3}$$

Now we have introduced the model; our goal is to define an objective function mathematically and proceed to minimize it. From Equation (1) in Section (3), we have

$$O(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{j=1}^K \rho(\hat{f}_j), \tag{4}$$

where  $l(\cdot, \cdot)$  is a convex differentiable function that measures the difference between actual  $y_i$  and predicted  $\hat{y}_i$ .  $\rho(\hat{f}_j) = \gamma T + \frac{1}{2} \lambda (\|w\|)^2$ .  $T$  is the number of leaves in the tree.  $\gamma$  and  $\lambda$  are the model hyper-parameters. From Eqs. (3) and (4), at the  $t^{th}$  iteration, the objective function can be written as

$$\begin{aligned} O^{(t)} &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \rho(\hat{f}_i) \\ &= \sum_{i=1}^n l(y_i, (\hat{y}_i^{(t-1)} + \hat{f}_t(x_i))) + \sum_{i=1}^t \rho(\hat{f}_i) \end{aligned} \tag{5}$$

Since, we use the mean-square error loss function, the above equation takes the following form:

$$\begin{aligned}
 O^{(t)} &= \sum_{i=1}^n (y_i - \hat{y}_i^{(t-1)} + \hat{f}_t(x_i))^2 + \sum_{i=1}^t \rho(\hat{f}_i) \\
 &= \sum_{i=1}^n (y_i - \hat{y}_i^{(t-1)})^2 + \sum_{i=1}^n (\hat{f}_t(x_i))^2 \\
 &\quad - 2 \sum_{i=1}^n (y_i - \hat{y}_i^{(t-1)})\hat{f}_t(x_i) + \sum_{i=1}^t \rho(\hat{f}_i) \\
 &= \sum_{i=1}^n (y_i - \hat{y}_i^{(t-1)})^2 + \sum_{i=1}^n (\hat{f}_t(x_i))^2 \tag{6} \\
 &\quad - 2 \sum_{i=1}^n (y_i - \hat{y}_i^{(t-1)})\hat{f}_t(x_i) + \sum_{i=1}^{t-1} \rho(\hat{f}_i) + \rho(\hat{f}_t) \\
 &= \underbrace{-2 \sum_{i=1}^n (y_i - \hat{y}_i^{(t-1)})\hat{f}_t(x_i) + \sum_{i=1}^n (\hat{f}_t(x_i))^2 + \rho(\hat{f}_t)}_{\text{function of } t} + c
 \end{aligned}$$

where  $c = \sum_{i=1}^n (y_i - \hat{y}_i^{(t-1)})^2 + \sum_{i=1}^{t-1} \rho(\hat{f}_i)$  is a constant term (not a function of  $t$ ). From the above expression, the optimal weights of the leaf can be computed which minimizes the objective function. For details, see [12, 43]. In the next section, we discuss briefly the hyper-parameters for Gradient Boosted Machines (GBMs).

### 3.3 Model Tuning Gradient Boosted Machine (GBM)

Although GBMs are highly flexible, they can take significant time to tune and find the optimal combination of hyperparameters. If the learning algorithm is not applied properly with the optimal combination of the hyperparameters, the model is prone to overfitting the data; this suggests that it will predict the training data rather than the functional relationship between the risk factors and response variables. The following are the most typical hyperparameters seen in most GBM implementations:

#### 3.3.1 Number of Trees

It represents the total number of trees required to match the model. GBMs frequently necessitate a large number of trees. However, GBMs, unlike random forests, can overfit. Hence, the goal is to use cross-validation to estimate the appropriate number of trees that minimize the loss function of interest.



### 3.3.2 Depth of Trees

The complexity of the boosted ensemble is determined by the number of splits in each tree. It is in charge of the depth of the individual trees. Naturally, numbers range from 3 to 8; however, it is not uncommon to have a tree depth of 1 [19].

### 3.3.3 Shrinkage

The introduction of regularization by shrinkage is the traditional strategy to controlling model complexity. Shrinkage is employed in the context of GBMs to reduce or decrease the influence of each additionally fitted base learner. It decreases the number of incremental steps, penalizing the significance of each successive iteration. The idea behind this strategy is to take many modest steps to improve a model rather than taking a few enormous steps. If one of the boosting iterations is found to be incorrect, the adverse impact can be simply addressed in the following steps. The shrinking effect is usually denoted as the parameter  $\lambda \in (0, 1]$  and is applied to the final step in the gradient boosting algorithm [20, 24].

### 3.3.4 Subsampling

The subsampling approach has been demonstrated to increase the model's generalization features while minimizing the required computation resources [45]. The objective of this approach is to incorporate some unpredictability into the fitting procedure. Only a random subset of the training data is used to fit a consecutive base learner at each learning iteration. Frequently, training data is sampled without replacement (SWOR). Using less than 100% of the training observations implies the implementation of stochastic gradient descent (SGD). This helps to reduce overfitting and keep the loss function gradient from being trapped in a local minimum or plateau.

Extreme Gradient Boosting (XGBoost) performs in a similar mechanism as GBM using ensemble additive training. Both XGBoost and GBM follow the principle of gradient boosting. However, XGBoost uses some more regularized model parameters to reduce overfitting and obtain the bias-variance trade-off, which improves the performance of the model. For more theoretic and practical applications, see [13, 22, 41]. In the next section, we discuss the statistical data analysis and results.

## 4 Statistical Analysis and Results

One of the most important goals of our study is to predict the survival times of pancreatic cancer patients with the highest degree of accuracy. For that purpose, a number of machine learning (ML) and deep learning (DL) models have been tested

**Table 1** Hyper-parameters and Their Combinations in the Grid Search

Hyper-parameters	Value combination
Shrinkage (S)	(0.01, 0.1, 0.3)
Interaction.depth (I.D)	(2, 3, 5)
n.minobsinnode (N.M)	(5, 10)
bag.fraction (B.F)	(0.65, 0.8, 1)

and validated on our data. We used Feed forward Deep Learning Models [5, 18, 46] with different layers, optimizer, and activation functions [29]. The best deep learning model that we have obtained is a dense feed-forward network with RMSE **0.38** on the test data. However, our proposed XGBoost model does the prediction task with significantly lower RMSE **0.04** on test data.

As described in Section 2.1, in our data, we have seven categorical and three numeric risk factors. Usually, most of the ML and DL algorithms do not accept categorical/factor inputs. This implies that the categorical risk factors must be converted to a numerical form. However, in our case, 70% of the risk factors are non-numeric in nature. To overcome this problem, we used a sophisticated technique, termed as “one-hot-encoding” [40]. It is a tool to convert the categorical predictors to numeric in ML algorithms to do a better job in prediction. After we convert the risk factors to a numeric scale, we perform *Min-Max normalization* on the set of risk factors. Min-Max normalization is a tool used in ML tasks to adjust the predictors and responses when they are in different scales. Usually, it makes all the predictors fall into [0,1]. It is defined as follows:

$$y^* = \frac{y - \min(y)}{\max(y) - \min(y)} \quad (7)$$

where  $y$  and  $y^*$  are the original response value, and the normalized value of the response respectively. After training the XGBoost model, we can back-transform to get the original prediction of the response. In our data set, the minimum and maximum responses are 0.21 years and 21 years respectively. Hence,  $\min(y) = 0.21$  years,  $\max(y) = 21$  years, and  $\max(y) - \min(y) = (21 - 0.21) = 20.79$  years. Now, we can back transform (7) in the following manner:

$$\begin{aligned} y &= \min(y) + y^*[\max(y) - \min(y)] \\ &= 0.21 + 20.79y^* \end{aligned} \quad (8)$$

We also performed the z-score standardization with the data but, the min-max normalization provided better performance with XGBoost. After normalizing the data, we divided the data into 70% training and 30% test data. At first, we perform the GBM algorithm on the data. In order to find the best combination of hyperparameters, we performed *grid search* mechanism [7, 26] that iterates through every possible combination of hyperparameter values and enables us to select the

**Table 2** Top 10 Models with Hyper-parameters for GBM

S	I.D	N.M	B.F	O.T	min_RMSE
0.3	5	5	0.8	47	0.03217434
0.3	5	10	1	87	0.03354224
0.1	5	5	0.8	140	0.03358716
0.1	3	5	0.8	232	0.03376142
0.1	3	5	1	413	0.03376934
0.3	5	10	0.65	126	0.03377321
0.1	5	10	0.8	206	0.03380464
0.1	2	5	0.65	603	0.03382063
0.01	5	5	0.65	1000	0.03382830
0.3	3	10	1	76	0.03386993

most suitable combination. To perform a grid search, we create our grid of hyperparameter combinations. We searched across 54 models with varying learning rates (shrinkage), tree depth (interaction.depth), and the minimum number of observations allowed in the trees' terminal nodes (n.minobsinnode). We also introduced stochastic gradient descent (SGD) in the grid search (bag.fraction < 1). The following Table 1 shows the combinations of the hyperparameters (abbreviated by S, I.D, N.M, and B.F, respectively) we used for the grid search to obtain 54 models.

We loop through each hyperparameter combination and apply the grid search on 1,000 trees. After around 30 min, our grid search completes, and we the estimated hyper-parameters for all **54 models**. The following Table 2 shows **top ten** models (ascending order of RMSE ) with the particular choices of the hyper-parameters.

From the above table, we see that, while training the model, we obtain the minimum RMSE (**0.03217434**) for the following optimal values of the hyper-parameters in the model:

- shrinkage (S): 0.3
- interaction.depth (I.D): 5
- n.minobsinnode (N.M): 5
- bag.fraction (B.F): 0.8
- optimal\_trees (O.T): 47

Now we have the optimal values of the hyper-parameters, we utilize 5-fold cross-validation to train our model with the hyper-parameters. The RMSE we obtained in the test data set using GBM is **0.04222367**.

Now we proceed to perform the data analysis with XGBoost, which is more sophisticated than GBM and has more options to set the hyper-parameters to reduce overfitting. It has several hyperparameters options to train the model. We shall describe briefly the hyperparameters we used for training the model according to the definition given in the **R software** module [14].

**Table 3** Top 10 Models with Hyper-parameters for XGBoost

eta	M.D	MCW	SS	CSBT	OT	min_RMSE
0.05	7	1	0.8	0.8	158	0.0304000
0.05	7	3	1	0.8	182	0.0305060
0.01	7	1	0.8	0.65	713	0.0305134
0.05	7	3	0.8	0.8	141	0.0306156
0.05	7	3	1	0.8	134	0.0306568
0.01	7	1	0.65	0.65	762	0.0307100
0.01	7	1	0.8	0.65	725	0.0307280
0.05	7	1	0.65	0.8	174	0.0307378
0.01	7	1	0.65	0.8	725	0.0307526
0.01	7	1	1	0.8	816	0.0307682

- **nrounds**: Controls the maximum number of iterations.
- **eta**: Controls the learning rate, or how quickly the model learns data patterns.
- **max\_depth (MW)**: The depth of the tree is controlled by this variable. Typically, the greater the depth, the more complex the model grows, increasing the likelihood of overfitting.
- **min\_child\_weight (MCW)**: It denotes the smallest number of instances required in a child node in the context of a regression problem. It aids in preventing overfitting by avoiding potential feature interactions.
- **subsample (SS)**: It regulates the number of samples (observations) provided to a tree.
- **colsample\_bytree (CSBT)**: It controls the number of predictors given to a tree.

Similar to GBM, we perform a grid search with different combinations of hyper-parameters. We trained 243 different hyper-parameter combinations to model. The following Table 3 shows **top ten** models (ascending order of RMSE ) with the particular choices of the hyperparameters.

From the above table, we see that the minimum RMSE (**0.0304**) was achieved while training the data when

- eta = 0.05
- max\_depth (MD) = 7
- min\_child\_weigh (MCW) = 1
- subsample (SS) = 0.8
- colsample\_bytree (CSBT) = 0.8
- optimal\_trees (OT) = 158

Therefore, our final XGBoost ensemble model can be expressed as follows

$$\hat{y}^* = \hat{f}(x) = \sum_{i=1}^{158} \hat{f}_i(x), \quad \hat{f}_i \in \mathcal{F} \quad (9)$$

where  $\mathcal{F}$  is the collection of all possible regression trees and  $\hat{f}_i$  are the additive functions (additive trees) in  $\mathcal{F}$ . Our analytical model provides the best results with the optimal values of the six hyper-parameters mentioned above. With the optimal values of the hyper-parameters, we train our model with 5-fold cross-validation and obtained an RMSE of **0.04127676** in test data, which is better than what we obtained using GBM.

We can provide the algorithm to obtain the best analytical model with the optimal hyper-parameters in the following manner:

#### Algorithm for Obtaining Optimal Analytical Model

##### Input

- Input Vector:  $X = (x_1, x_2, \dots, x_n)$ .
- response  $y$  as output.
- Number of iteration  $T$  decided by the researcher.
- Mean Square Error Loss Function  $L(\theta) = \sum_{i=1}^n (y_i - \hat{y}_i)$ .
- Decision tree as base (weak) learner to be combined in the ensemble.

##### Algorithm

- for  $t = 1$  to  $T$  do
  1. Initially, a decision tree is fitted to the data:  $\hat{f}_1(x) = y$ .
  2. Next, the subsequent decision tree is fitted to the prior tree's residuals:  $d_1(x) = y - \hat{f}_1(x)$
  3. The latest tree is then added to the algorithm:  $\hat{f}_2(x) = \hat{f}_1(x) + d_1(x)$ .
  4. The succeeding decision tree is fitted to the residuals of  $\hat{f}_2$  :  $d_2(x) = y - \hat{f}_2(x)$ .
  5. The new tree is then added to our algorithm:  $\hat{f}_3(x) = \hat{f}_2 + d_2(x)$
  6. Use cross-validation while training the model to decide the stopping criteria of the training process.
  7. Create a hyper-parameter grid with some user provided values and perform grid search mechanism to find optimal combination of the hyper-parameters.
  8. The final analytical model is the sum of all the decision tree base learners with optimal values of the hyper-parameter along with the optimal number of trees  $T^*$ :  $\hat{f} = \sum_{i=1}^{T^*} \hat{f}_i$ .
- end.

#### 4.1 Validation of the Proposed Model

After developing our proposed analytical model, it is most important to validate the model so that we can implement it to obtain the best results. In developing the

**Table 4** Predicted and Actual Response

Predicted response	Actual response
1.5849055	1.7806254
2.1938655	2.0418507
2.3095083	2.0542900
2.5678812	2.1577326
2.1382802	2.3273000
3.5089106	3.7427615
3.2106355	3.3957704
2.4213239	2.5643014
1.2646362	1.6215333
1.5551881	1.8559159
2.1867148	2.4340161
2.9590347	3.2622116

model, we used 70% of the training data and obtained an RMSE of **0.034**. It is a usual tendency of a good model to have a predictive performance in the test data set close to the training data set. When we implement our model on the test data set, we obtained an RMSE of 0.0422, which is very close to what we have obtained in the training set, implying that our model performs well on the unseen/future data set. We can predict the survival times (in years) by back-transforming the scaled response using equation (8) from Section 4 and compare how good the prediction is. The following Table 4 shows the actual and estimated predictions of pancreatic survival times (in years).

From the above table, we see that the predictions are very close to the actual response.

To validate our prediction accuracy, we also performed Wilcoxon's rank-sum test with continuity correction to check if the actual and predicted responses are significantly different. The test produced a p-value of **0.5** ( $> 0.05$ ), implying that there is insufficient sample evidence to reject the null hypothesis that both actual and predicted responses are the same. Thus, the test suggests there is no significant difference between the actual and predicted responses at a 5% level of significance.

## 4.2 Comparison with Different Models

The XGBoost method performed really well and was about 96% accurate. We compared the proposed boosted regression tree (using XGBoost) model with different deep-learning models to validate its performance. Deep learning models are efficient with a large amount of data to train to address the complex structure of features. We used activation functions like rectified linear unit (ReLU), Exponential Linear Unit (ELU), scaled exponential linear units (SELU), and Hyperbolic Tangent (tanh) in different layers of the deep network and used optimizer like stochastic gradient descent (SGD), Root Mean Square Propagation (RMSprop), and Adam (derived from adaptive moment estimation). In some models, we introduced

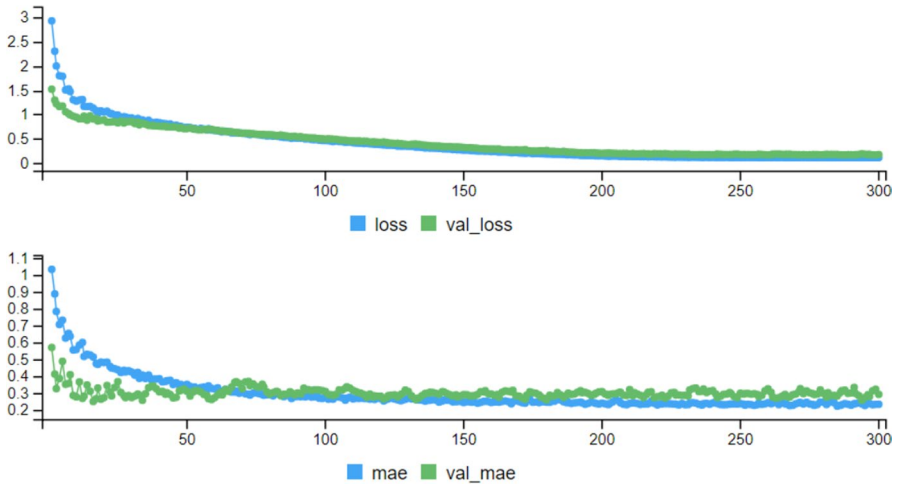


Fig. 2 RMSE and MAE of DL6 for Training and Validation Data

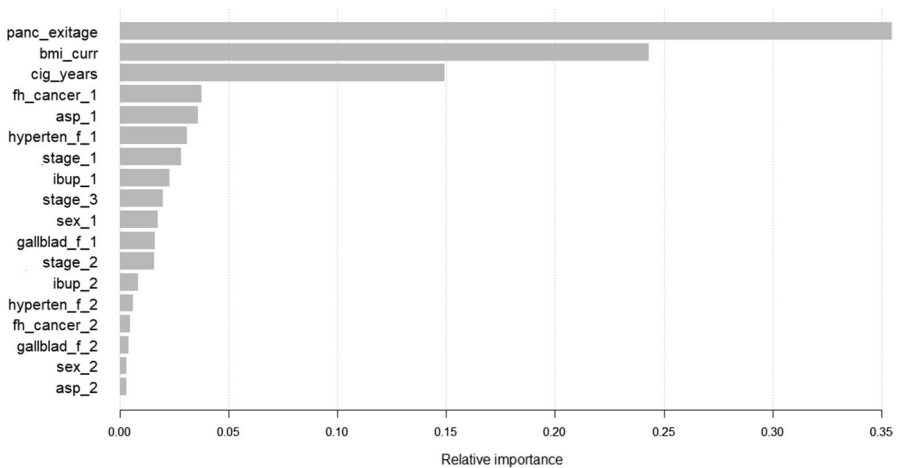
Table 5 Comparison of Different GBM & XGBoost Models in Terms of RMSE and MAE in Test Data

MODEL	RMSE	MAE
XGBoost	0.0412	0.034
GBM	0.0422	0.039

Table 6 Comparison of Different Deep Learning Models in Terms of RMSE and MAE in the Test Data

Model	Unit	AF	OPT	DROP	BN	RMSE	MAE
DL1	(100,90,50)	(tanh,tanh,relu)	RMSprop	yes	yes	0.381	0.26
DL2	(100,90,50)	(ReLU,ReLU,ReLU)	Adam	yes	yes	0.391	0.24
DL3	(100,90,50)	(ReLU,ReLU,ReLU)	SGD	yes	yes	0.9	0.255
DL4	(100,90,50)	(ReLU,ReLU,ReLU)	RMSprop	Yes	Yes	0.391	0.25
DL5	(100,90,50)	(ReLU,ReLU,ReLU)	Adam	No	No	0.39	0.26
<b>DL6</b>	<b>(100,90,50)</b>	<b>(tanh,tanh,tanh)</b>	<b>Adam</b>	<b>Yes</b>	<b>Yes</b>	<b>0.378</b>	<b>0.249</b>
DL7	(100,90,50)	(ELU,ELU,ReLU)	Adam	Yes	Yes	0.388	0.234
DL8	(100,90,50)	(ReLU,SELU,ELU)	Adam	Yes	Yes	0.385	0.232
DL9	(100,90,50)	(ReLU,ReLU,ReLU)	Adam	No	Yes	0.49	0.4
DL10	(100,90,50)	(ReLU,ReLU,ReLU)	Adam	No	No	0.51	0.3

dropouts and batch normalization, and in some models, we did not. Adding dropouts [21] and batch-normalization usually prevents overfitting in the networks and boosts the performance. The theoretical details and applications of the optimizer and activation functions can be found in [1, 6, 51]. Each of the models is trained



**Fig. 3** The Relative Importance of Risk Factors Used in the XGBoost Model

using 300 epochs and batch size = 32. Table 6 compares different deep learning models in terms of root mean square error (RMSE) and mean absolute error (MAE) in the test data. In the following table, the activation function, optimizer, dropout, and batch normalization are abbreviated as AF, OPT., DROP., and BN, respectively. We considered ten deep learning sequential models with three dense layers containing units 100, 90, and 50, respectively. As Table 6 illustrates, the best deep learning model (DL6) with minimum RMSE (.378) is the model where we use *tanh* activation function in each of the three hidden layers, use optimizer *Adam*, use *dropout* with *batch-normalization*. The following Fig. 2 illustrates the graph of RMSE and MAE of DL6 while training.

The following Table 5 compares the boosted regression tree model using GBM and XGBoost in terms of RMSE and MAE in test data.

As the above Table 5 illustrates, the XGBoost performs the best with the minimum RMSE.

### 4.3 Ranking of Risk Factors and Prediction of the Survival Time

Once we have found the best-performing model, it is important to rank the pancreatic risk factors according to their relative importance. We rank the contributing risk factor in survival time using the measure *Gain*,<sup>38</sup>. The gain denotes the relative impact of a certain risk factor to the model, which is computed by considering each predictor's contributions to each tree in the model. A higher value of this metric for a specific risk factor, compared to another risk factor, implies that the risk factor with a higher gain is more important for generating a prediction.

From Fig. 3, we see that the top five most contributing risk factors in the model are age, current bmi, the number of years a patient smoked cigarettes, people who have family history of cancer, and people who took aspirin on a regular basis.



**Table 7** Risk Factors and Their Percentage of Contribution to The Response

Risk Factors	% Contribution
panc_exitage	35.5
bmi_curr	24.3
cig_years	14.93
fh_cancer_1	3.76
asp_1	3.6
hyperten_f_1	3.1
stage_1	2.82
ibup_1	2.29
stage_3	1.96
sex_1	1.73
gallblad_f_1	1.6
stage_2	1.57
ibup_2	0.83
hyperten_f_2	0.61
fh_cancer_2	0.45
gallblad_f_2	0.4
sex_2	0.29
asp_2	0.28

Table 7 illustrates the percentage contributions of the risk factors to the response survival times.

From Table 7, see that the risk factors explain **96.42%** of the total variation of the response.

## 5 Conclusion

In cancer research, one of the most important aspects is to estimate the survival times of the patients. It results in improved management, more efficient use of resources, and the provision of specialized treatment alternatives. [4, 34]. It is imperative to investigate the clinical diagnosis and enhance the therapeutic/treatment strategy of pancreatic cancer. Pancreatic cancer is one of the deadliest cancer, and in most cases, detected in later stages (stage III /IV). Once a patient is diagnosed with pancreatic cancer, he/she or his/her family members would be interested in knowing how long is the expected/predicted survival. This question is usually asked by patients with a terminal illness to their doctors. However, it is impossible to provide the exact answer to these questions; doctors provide an answer which is mainly subjective. If we have a model based on real data that answer the questions given a particular choice of risk factors, it would be very helpful to doctors and medical professionals. Also, if we have some more relevant risk factors, we can incorporate those into this model. This would be very helpful for healthcare professionals and patients with terminal illnesses.

Given a collection of risk factors, we can build a questionnaire (attached in Appendix I) that can address the patient information who are diagnosed with pancreatic cancer. Based on their response, the estimate of the survival times can be obtained very accurately. To our knowledge, there is no such model that is as accurate as our predictive analytical model. In this study,

1. We developed a boosted ensemble regression tree model using XGBoost that is very accurate and performs well on test data sets, given a collection of risk factors (numeric and categorical).
2. We ranked all the risk factors according to their relative importance in the boosted model. This ranking provides the percentage of contribution of the individual risk factors to the response and survival time.
3. We compared the performance of the XGBoost model with the GBM model and other ten deep-learning sequential models with different activation functions and optimizers. The XGBoost model produced an RMSE and MAE of **0.0412** and **.034** which is the smallest on the test data compared to all of the other models.
4. The proposed analytical model can be implemented to any future data set containing information on different risk factors relating to the subject study to obtain very good predictive performance.

## Appendix

In the appendix, our version of the survey questionnaire for NIH pancreatic data is posted and we request the same type of information.

### Questionnaire—

1. panc\_exitage (Numeric) ( $X_1$ ): Age of diagnosis of the patient.
2. Stage (Categorical) ( $X_2$ ): Pancreatic Cancer Stages, categorized as a) localized, b) regional, and c) distant
3. Asp (Categorical) ( $X_3$ ): Does the person use Aspirin Regularly? “During the last 12 months, have you regularly used aspirin or aspirin containing products, such as Bayer, Bufferin or Anacin? (Please do not include aspirin-free products such as Tylenol and Panadol.)” 0=“No” 1=“Yes”
4. Ibuf (Categorical) ( $X_4$ ): Does the person use Ibuprofen Regularly? “During the last 12 months, have you regularly used ibuprofen-containing products, such as Advil, Nuprin, or Motrin?” 0=“No” 1=“Yes”
5. fh\_cancer (Categorical) ( $X_5$ ): The number of first-degree relatives with pancreatic cancer. Any first-degree relative with cancer. 0=“No” 1=“Yes”
6. Sex ( $X_6$ ): Sex of the individual. 1=“Male” 2=“Female”
7. BMI (numeric) ( $X_7$ ): Current Body Mass Index (BMI) at Baseline (In lb/in<sup>2</sup>)
8. Cigarette Years (numeric) ( $X_8$ ): The total number of years the patient smoked.
9. gallblad\_f (Categorical) ( $X_9$ ): Did the individual ever have gallbladder stones or inflammation? 0=“No” 1=“Yes”

10. hyperten\_f (Categorical)(X<sub>10</sub>): Did the individual ever have high blood pressure? 0=“No” 1=“Yes”

**Acknowledgements** This paper is based on the third chapter of my doctoral dissertation (<https://digitallcommons.usf.edu/etd/9316/>).

**Funding** Not Applicable

## Declarations

**Conflict of interest** There are no competing interests to declare.

**Consent for publication** Not Applicable

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Agostinelli, F., Hoffman, M., Sadowski, P., Baldi, P.: Learning activation functions to improve deep neural networks. (2014) arXiv preprint [arXiv:1412.6830](https://arxiv.org/abs/1412.6830)
2. Ahmad, L.G., Eshlaghy, A.T., Poorebrahimi, A., Ebrahimi, M., Razavi, A.R.: Using Three Machine Learning Techniques for Predicting Breast Cancer Recurrence. *J. Health Med. Inform.* **4**, 124 (2013). <https://doi.org/10.4172/2157-7420.1000124>
3. Amjad, M., et al.: Prediction of pile bearing capacity using XGBoost algorithm: modeling and performance evaluation. *Appl. Sci.* **12**(4), 2126 (2022)
4. Bal, M.S., Bodal, V.K., Kaur, J., Kaur, M., Sharma, S.: Patterns of Cancer: A Study of 500 Punjabi Patients. *Asian Pac. J. Cancer Prev.* **16**(12), 5107–10 (2015)
5. Bebis, G., Georgiopoulos, M.: Feed-forward neural networks. *IEEE Potentials* **13**(4), 27–31 (1994)
6. Bello, I., Zoph, B., Vasudevan, V., Le, Q. V.: Neural optimizer search with reinforcement learning. In *International Conference on Machine Learning* (pp. 459–468). PMLR (2017)
7. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *Journal of machine learning research*, **13**(2), (2012)
8. Chakraborty, A., Tsokos, C.: Survival Analysis for Pancreatic Cancer Patients using Cox-Proportional Hazard (CPH) Model. *Global J. Med. Res.* (2021). <https://doi.org/10.34257/GJMRFVOL21IS3PG29>
9. Chakraborty, A., Tsokos, C.P.: Parametric and Non-Parametric Survival Analysis of Patients with Acute Myeloid Leukemia (AML). *Open J. Appl. Sci.* **11**, 126–148 (2021). <https://doi.org/10.4236/ojapps.2021.111009>
10. Chakraborty, A., Tsokos, C.P.: A Real Data-Driven Analytical Model to Predict Happiness. *Sch. J. Phys. Math. Stat.* **8**(3), 45–61 (2021)
11. Chang, W., et al.: Prediction of hypertension outcomes based on gain sequence forward tabu search feature selection and xgboost. *Diagnostics* **11**(5), 792 (2021)
12. Chen, T., Guestrin, C.: 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, August)

13. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H.: Xgboost: extreme gradient boosting. R package version 0.4-2, **1**(4), (2015)
14. Chen, T., He, T., Benesty, M., Khotilovich, V.: Package “xgboost”. R version, 90 (2019)
15. Chen, Y., Jia, Z., Mercola, D., Xie, X.: A Gradient Boosting Algorithm for Survival Analysis via Direct Optimization of Concordance Index. *Comput. Math. Method. Med.* **2013**, 1–8 (2013). <https://doi.org/10.1155/2013/873595>
16. Cicchetti, D.: Neural networks and diagnosis in the clinical laboratory: state of the art. *Clin. Chem.* **38**, 9–10 (1992)
17. Cochran, A.J.: Prediction of outcome for patients with cutaneous melanoma. *Pigment Cell Res.* **10**, 162–167 (1997)
18. Fine, T.L.: Feedforward neural network methodology. Springer Science Business Media, USA (2006)
19. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning, vol. 1. Springer Series in Statistics New York, NY, USA (2001)
20. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. *Annal. Statist.* **29**(5), 1189–1232 (2001). <https://doi.org/10.1214/aos/1013203451>
21. Garbin, C., Zhu, X., Marques, O.: Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimed Tools Appl* 1–39 (2020)
22. Gómez-Ríos, A., Luengo, J., Herrera, F.: A Study on the Noise Label Influence in Boosting Algorithms: AdaBoost, GBM and XGBoost. *Hybrid Artif. Intell. Syst.* 268–280 (2017). [https://doi.org/10.1007/978-3-319-59650-1\\_23](https://doi.org/10.1007/978-3-319-59650-1_23)
23. Hayward, J., Alvarez, S.A., Ruiz, C., Sullivan, M., Tseng, J., Whalen, G.: Machine learning of clinical performance in a pancreatic cancer database. *Artif. Intell. Med.* **49**(3), 187–195 (2010). <https://doi.org/10.1016/j.artmed.2010.04.009>
24. Hothorn, T., Buhlmann, P., Kneib, T., Schmid, M., Hofner, B.: Model-based boosting 2.0. *J. Mach. Learn. Res.* **11**, 2109–2113 (2010)
25. Hu, J.-X., et al.: Pancreatic cancer: A review of epidemiology, trend, and risk factors. *World. J. Gastroenterol.* **27**(27), 4298 (2021)
26. Jiménez, Á.B., Lázaro, J.L., Dorronsoro, J.R.: Finding optimal model parameters by discrete grid search. In *Innovations in Hybrid Intelligent Systems* (pp. 120–127). Springer, Berlin, Heidelberg (2007)
27. Khan, M.A., et al.: Corporate vulnerability in the US and China during COVID-19: A machine learning approach. *J. Econ. Asymmet.* **27**, e00302 (2023)
28. Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V., Fotiadis, D.I.: Machine learning applications in cancer prognosis and prediction. *Comput. Struct. Biotechnol. J.* **13**, 8–17 (2015). <https://doi.org/10.1016/j.csbj.2014.11.005>
29. Leshno, M., Lin, V.Y., Pinkus, A., Schocken, S.: Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw.* **6**(6), 861–867 (1993)
30. Li, H., et al.: XGBoost model and its application to personal credit evaluation. *IEEE Intell. Syst.* **35**(3), 52–61 (2020)
31. Li, D., Xie, K., Wolff, R., Abbruzzese, J.L.: Pancreatic cancer. *Lancet* **363**(9414), 1049–1057 (2004). [https://doi.org/10.1016/s0140-6736\(04\)15841-8](https://doi.org/10.1016/s0140-6736(04)15841-8)
32. Lu, H., Wang, H., Yoon, S.W.: A Dynamic Gradient Boosting Machine Using Genetic Optimizer for Practical Breast Cancer Prognosis. *Expert Syst. Appl.* (2018). <https://doi.org/10.1016/j.eswa.2018.08.040>
33. Ma, B., Meng, F., Yan, G., Yan, H., Chai, B., Song, F.: Diagnostic classification of cancers using extreme gradient boosting algorithm and multi-omics data. *Comput. Biol. Med.* **103761**, (2020). <https://doi.org/10.1016/j.combiomed.2020.103761>
34. Mehrabani, D., Tabei, S., Heydari, S., Shamsina, S., Shokrpour, N., Amini, M., et al.: Cancer occurrence in Fars Province, Southern Iran. *Iran Red. Crescent. Med. J.* **10**(4), 314–22 (2008)
35. Michaud, D.S.: Epidemiology of pancreatic cancer. *Minerva Chir.* **59**(2), 99–111 (2004)
36. Mikhaylov, A., et al.: Integrated decision recommendation system using iteration-enhanced collaborative filtering, golden cut bipolar for analyzing the risk-based oil market spillovers. *Comput. Econ.* 1–34 (2022)
37. Mizrahi, J.D., et al.: Pancreatic cancer. *Lancet* **395**(10242), 2008–2020 (2020)
38. Natekin, A., Knoll, A.: Gradient boosting machines, a tutorial. *Front. Neurobot.* **7**, 21 (2013). <https://doi.org/10.3389/fnbot.2013.00021>

39. Park, K., Ali, A., Kim, D., An, Y., Kim, M.H.: Shin Robust predictive model for evaluating breast cancer survivability. *Engl. Appl. Artif. Intell* **26**, 2194–2205 (2013)
40. Seger, C.: An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing (2018)
41. Sheridan, R.P., Wang, W.M., Liaw, A., Ma, J., Gifford, E.M.: Extreme gradient boosting as a method for quantitative structure-activity relationships. *J. Chem. Inf. Model.* **56**(12), 2353–2360 (2016)
42. Shi, X., et al.: A feature learning approach based on XGBoost for driving assessment and risk prediction. *Accident Anal. Prevent.* **129**, 170–179 (2019)
43. Song, R., Chen, S., Deng, B., Li, L.: eXtreme gradient boosting for identifying individual users across different digital devices. In *International Conference on Web-Age Information Management* (pp. 43–54). Springer, Cham (2016)
44. Stødle, K., Flage, R., Guikema, S. D., Aven, T.: Data-driven predictive modelling in risk assessment: Challenges and directions for proper uncertainty representation. *Risk Anal.* (2023)
45. Sutton, C.D.: Classification and regression trees, bagging, and boosting. *Handb. Stat.* **24**, 303–329 (2005). [https://doi.org/10.1016/S0169-7161\(04\)24011-1](https://doi.org/10.1016/S0169-7161(04)24011-1)
46. Svozil, D., Kvasnicka, V., Pospichal, J.: Introduction to multi-layer feed-forward neural networks. *Chemom. Intell. Lab. Syst.* **39**(1), 43–62 (1997)
47. Vincent, A., Herman, J., Schulick, R., Hruban, R.H., Goggins, M.: Pancreatic cancer. *Lancet* **378**(9791), 607–620 (2011). [https://doi.org/10.1016/s0140-6736\(10\)62307-0](https://doi.org/10.1016/s0140-6736(10)62307-0)
48. Wang, J., et al.: A data-driven integrated framework for predictive probabilistic risk analytics of overhead contact lines based on dynamic Bayesian network. *Reliabil. Eng. Syst. Safety.* **235**, 109266 (2023)
49. Wenxin, J.: On weak base hypotheses and their implications for boosting regression and classification. *Ann. Stat.* **30**, 51–73 (2002)
50. Yang, J., Guan, J.: A heart disease prediction model based on feature optimization and smote-Xgboost algorithm. *Information* **13**(10), 475 (2022)
51. Zhang, Z.: Improved adam optimizer for deep neural networks. In: *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)* (pp. 1–2). IEEE (2018)

## Authors and Affiliations

Aditya Chakraborty<sup>1</sup>  · Chris P. Tsokos<sup>2</sup> 

✉ Aditya Chakraborty  
chakraa@evms.edu

Chris P. Tsokos  
ctsokos@usf.edu

<sup>1</sup> Eastern Virginia Medical School, Norfolk, USA

<sup>2</sup> University of South Florida, Tampa, USA