Predicción de mortalidad en pacientes de ICU. Physionet Challenge 2012.

López Velasco José Armando[1][31315734−3]

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México
armando.lopez@c3.unam.mx

Resumen En las últimas décadas, el sector salud ha ido mejorando con pasos agigantados en su tecnología y el cuidado de sus pacientes, sin embargo, aún nos encontramos lejos del ideal deseado para ésta área. Una solución reside en la Inteligencia Artificial aplicada a la salud.
El challenge "Predicting Mortality of ICU Patients - The PhysioNet Computing in Cardiology Challenge 2012"[1] sirvió para desarrollar modelos para predecir mortalidad de pacientes en ICU(Intensive Care Units) después de 48 horas de haber ingresado.
En este artículo se desarrollaron distintos modelos de aprendizaje máquina para predecir la mortalidad de un paciente. Exploré todas las variables y extraje las más útiles para perfeccionar el modelo, utilizando medidas de dispersión conocidas.
Entre la búsqueda de modelos para la tarea se exploraron modelos comunes de aprendizaje máquina y redes neuronales básicas, el modelo más prometedor obtuvo un SCORE1[2] de 0.4513 con una red neuronal y de 0.4615 con regresión logística y oversampling.

Keywords: Machine Learning · Neural Networks · Mortality prediction · Medicine · Healthcare · IA · ICU · Over Sampling.

1. Introducción

La última década la industria tecnológica ha tenido un enorme crecimiento en distintas áreas, una de ellas, y una de las más importantes, es la tecnología aplicada a la medicina, en los últimos años la cantidad de datos obtenidos de los pacientes ha aumentado considerablemente. Los aparatos más novedosos para el monitoreo de pacientes los podemos ver en practicamente cualquier hospital de primer mundo en salas de ICU, éstos se encuentran extrayendo datos cada segundo. Una utilidad inmediata a estos datos es la de monitorear pacientes y tratar de conocer su desenlace, de esta manera se pueden incrementar los esfuerzos realizados para salvar una vida.

[1] https://physionet.org/content/challenge-2012/1.0.0/
[2] Métrica utilizada para rankear a los participantes

Actualmente existen métricas conocidas que ayudan al personal médico a tener una idea de la gravedad de un paciente respecto a una escala, dos de las más conocidas son SAPS y SOFA [6] [5], sin embargo, estas métricas tienen una exactitud relativamente baja, ya que se hace conforme a puntaje dependiendo de distintos niveles. Es por esto que se han desarrollado distintos modelos matemáticos para predicción de mortalidad.

2. Estado del arte

En la actualidad existen distintos algoritmos de aprendizaje máquina que son utilizados para realizar ésta tarea, aunque la mayoría no resultan ser un algoritmo específico para mortalidad, existen diversos que han presentado un gran potencial en realizar ésta tarea. Muchos de éstos están basados en árboles con distintas variantes, por ejemplo:

- Basados en ensambles Bayesianos
- Basados en redes neuronales
- Basados en ensambles de modelos aprendizaje máquina

Los clasificadores basados en árboles usando como marco de referencia un modelo Bayesiano[2] tiene bastantes ventajas, una de ellas es el alto rendimiento computacional que poseen ya que no necesitan entrenar muchos parámetros ya que se modelan las distribuciones de manera explícita.
Cada árbol selecciona un subconjunto de datos utilizando dos regresiones que los dividen. Estas observaciones son dadas un número aleatorio de veces al modelo así eligirá una variable y el árbol agregará además una contribución, esto ayuda para los valores faltantes, un problema muy común con los datos obtenidos de ICU. La clasificación se hace mediante votación de las hojas, así la probabilidad de clase está dada por la suma de cada una de las menciones.

$$Y = \sum_{j=i}^{m} g(x; T_j, M_j) + \epsilon, \epsilon \sim N(0, \sigma^2), \tag{1}$$

Donde para cada árbol binario de regresión T_j y su nodo terminal asociado de parámetros M_g, $g(x; T_j.M_j)$ es la función que asigna cada parámetro de las hojas de cada árbol (μ_{ij}) tal que $\mu_{ij} \in M_j$

Si bien, no es un algoritmo nuevo, por la naturaleza de los datos, es un algoritmo que está teniendo un florecimiento para esta tarea.

2.1. Trabajos relacionados

En el 2013 Mitchell[8] realizó una investigación sobre la predicción de tasas de mortalidad, utilizó un índice modelado como una Gaussiana de normal inversa para poder predecir la tasa de mortalidad. En el 2017 Li, Yuenan[7] predice tasas de mortalidad respecto a contaminación, el modelo utilizado fue una regresión

ponderada. En el 2015 Wang, G.[9] predice las tasas de mortalidad después de cierta cirugía utilizando siete distintas técnicas de aprendizaje automático, entre ellos, una red neuronal de back propagation, función de base radial(RBFn), Extreme Learning Machine (EML), Regularized ELM, máquinas de soporte vectorial y K vecinos más cercanos. Entre sus resultados, lo más interesante fue que RELM resultó con el mejor accuracy para su problema con un valor de 0.8 y una gran velocidad de aprendizaje.

Algunos algoritmos mencionados en la literatura mencionan como buenos clasificadores para ésta tarea a la vieja cofiable regresión logística, Gradient Boosted Trees, y técnicas de aprendizaje profundo, en las cuales no profundizaremos.

3. Datos

Los datos fueron presentados para un challenge en el 2012 por Physionet [3] llamado "Predicting Mortality of ICU Patients - The PhysioNet Computing in Cardiology Challenge 2012"[4]

Los datos usados fueron obtenidos de 12,000 pacientes de ICU, todos los pacientes son adultos que fueron admitidos por distintas enfermedades/razones, entre ellas, problemas: cardiacos, médicos, quirúrgicos, traumas. Cada paciente sobrevivió y estuvo en ICU por al menos 48 horas. No se excluyeron pacientes con DNR(Orden de no resucitar) o con CMO (sólo medidas confortables).

Contamos con tres grupos A, B, C. A y B son dados para entrenamiento y validación, cada uno con 4,000 registros, el resto (C) es utilizado por los evaluadores para realizar la prueba. Se cuentan con dos tipos de archivos. Set y Outcomes.

3.1. Outcomes

En este archivo encontramos un archivo csv que contiene las siguientes variables:

- RecordID
- SAPS-I score [6]
- SOFA score [5]
- Length of stay (days): número de días que pasó el paciente desde que ingresó a ICU hasta el final de la hospitalización.
- Survival (days): El número de días entre la admisión y el deceso.
- In-hospital death (0: sobreviviente, or 1: murió en el hospital)

Para comprender *Survival* es mejor hacerlo de la siguiente manera:

$$\text{Survival Length of stay} \rightarrow Survivor$$
$$\text{Survival} = \text{-1} \rightarrow Survivor$$
$$2 \leq Survival \leq Lengthofstay \rightarrow In - hospitaldeath$$

[3] https://physionet.org/

3.2. Set A-B

Estos datos son obtenidos a lo largo del tiempo, por lo que cada paciente posee un archivo .txt con su RecordID, dentro encontraremos por cada renglón dos datos, el primero será la hora y el segundo la varaible con el valor. Todas las variables estáticas son registradas en la hora '00:00' y son las siguientes:

- RecordID
- Age
- Gender (0: mujer, 1: hombre)
- Height
- ICUType
- Medical o Surgicla ICU
- Weight

Después de que aparezcan éstas variables aparecerán las que cambian con el tiempo (variables dinámicas) las cuales son:

- Albumin (g/dL)
- ALP [Alkaline phosphatase (IU/L)]
- ALT [Alanine transaminase (IU/L)]
- AST [Aspartate transaminase (IU/L)]
- Bilirubin (mg/dL)
- BUN [Blood urea nitrogen (mg/dL)]
- Cholesterol (mg/dL)
- Creatinine [Serum creatinine (mg/dL)]
- DiasABP [Invasive diastolic arterial blood pressure (mmHg)]
- FiO2 [Fractional inspired O2 (0-1)]
- GCS [Glasgow Coma Score (3-15)]
- Glucose [Serum glucose (mg/dL)]
- HCO3 [Serum bicarbonate (mmol/L)]
- HCT [Hematocrit (
- HR [Heart rate (bpm)]
- K [Serum potassium (mEq/L)]
- Lactate (mmol/L)
- Mg [Serum magnesium (mmol/L)]
- MAP [Invasive mean arterial blood pressure (mmHg)]
- MechVent [Mechanical ventilation respiration (0:false, or 1:true)]
- Na [Serum sodium (mEq/L)]
- NIDiasABP [Noninvasive diastolic arterial blood pressure (mmHg)]
- NIMAP [Noninvasive mean arterial blood pressure (mmHg)]
- NISysABP [Noninvasive systolic arterial blood pressure (mmHg)]
- PaCO2 [partial pressure of arterial CO2 (mmHg)]
- PaO2 [Partial pressure of arterial O2 (mmHg)]
- pH [Arterial pH (0-14)]
- Platelets (cells/nL)
- RespRate [Respiration rate (bpm)]
- SaO2 [O2 saturation in hemoglobin (
- SysABP [Invasive systolic arterial blood pressure (mmHg)]
- Temp [Temperature (C)]
- TropI [Troponin-I (g/L)]
- TropT [Troponin-T (g/L)]
- Urine [Urine output (mL)]
- WBC [White blood cell count (cells/nL)]
- Weight (kg)*

4. Preprocesamiento

4.1. Análisis

Primero conozcamos la distribución de nuestros datos de los conjuntos A y B.

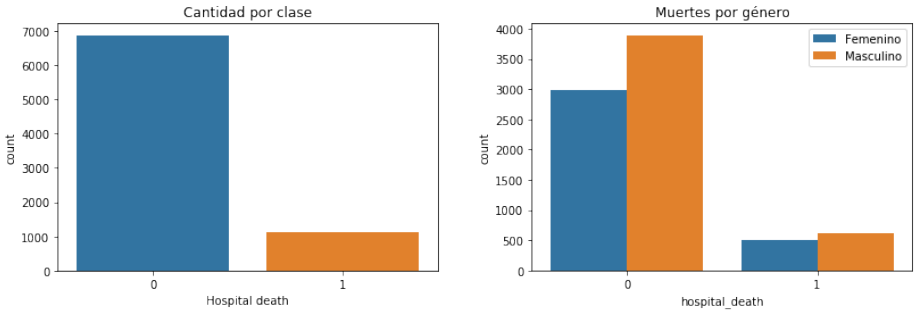

Figura 1. Distribución de mortalidad, por clase (izquierda) y por clase por sexo (derecha)

Como podemos observar en la **Figura 1** existe un claro desbalance de clases, esto provocará que el principal reto de nuestro modelo sea una clasificación acertada que no se incline por la clase mayoritaria, en este caso, no-muerte. Respecto a la cantidad de personas por género que mueren, podemos decir que están balanceados respecto al sexo, respecto a los que sobreviven, existe una notable diferencia, en este caso los pacientes masculinos tienen mayor índice de supervivencia, sin embargo la diferencia no es muy representativa como para tomarlo en cuenta.

Podemos notar en la **Figura 2** que la distribución de edades es asimétrica a la derecha, por lo que la mayoría de nuestros pacientes tienen una edad avanzada, con una media de 64.4 y mediana de 67 años.

4.2. Manipulación de los datos

Para alimentar el modelo se tomaron las primeras 48 horas de estancia de cada paciente, y en lugar de trabajar con la serie de tiempo opté por utilizar medidas de disperción conocidas.

Sin embargo, existe una gran cantidad elementos faltantes, en total, de las 42 variables eliminamos aquellas que tuvieran más del 25 % de elementos faltantes. Una vez obtenidas estas variables, procedemos a sacar las medidas. Para cada variable dinámica obtendremos lo siguiente:

- Mínimo

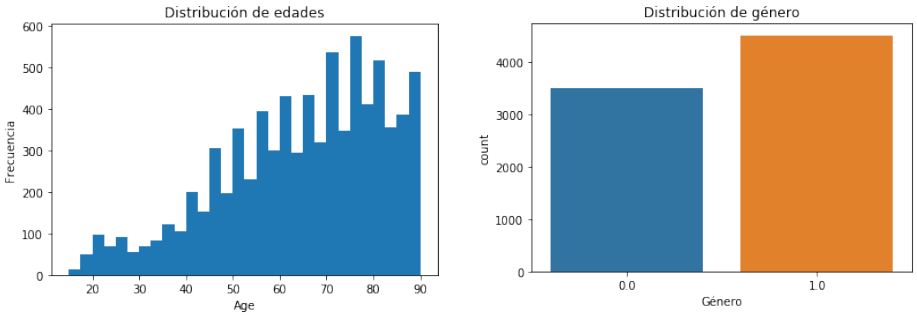

Figura 2. Distribución de edades y género, en ese orden (izquierda a derecha)

- Máximo
- Media
- Primero
- Último
- Diferencia o rango
- Varianza

Adicionalmente agregamos las variables estáticas tomadas en la hora 00:00 como la edad, género, etc. Posteriormente agregamos los dos descriptores SAPS y SOFA.

4.3. Llenando nulos

Para llenar la gran cantidad de nulos utilicé la técnica de imputar con la media, la cual es considerada una de las mejores técnicas en este tipo de datos [3].

Posterior a esto utilizamos Standard Scaler [4] para tener todos nuestros valores con media 0 y desviación de 1.

5. Modelos

Realicé dos stacks de modelos, debido a que contaba con un desbalance de clases tan alto (aprox 14 % con el target de interés) decidí resolver el problema con dos enfoques. El primero fue usar los datos tal cual ya están procesados (lo llamaré: Desbalanceado) y el segundo fue utilizar una técnica de Oversampling.

En todos los modelos utilicé CrossValidation con 5 dobleces, con un tamaño de validación del 20 %. La prueba se realiza con el conjunto C que no ha visto el modelo el cual consiste de 4 mil registros.

Los modelos utilizados serán:

[4] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

- Random Forest
- Logistic Regression
- Light Gradient Boost
- XGBoost Classifier
- AdaBoost

5.1. Random Forest

La elección de este modelo es porque es el modelo básico para realizar predicción de mortalidad, y justo es por eso que algunos de los algoritmos que utilizaré son derivaciones de Random Forest, que en realidad son árboles de decisión.

5.2. Logistic Regression

En la literatura que encontré al resolver este problema encontré que la mayoría de los clasificadores de mortalidad eran desarrollados o comparados contra una regresión logística. Para la expliación matemática del algoritmo puede viajar a la liga al pie de página. [5]

5.3. Light Gradient Boost

Este algoritmo está basado en los árboles de decisión, la mayoría de los modelos basados en árboles se cultivan o crecen pro profundiad, es decir, en cada iteración el árbol puede ser un nivel al menos más profundo. LGBM primero escogerá la hoja con pérdida máxima para crecer el accuracy manteniendo la hoja fija así logra bajar la pérdida y creciendo a los lados. [6] Además LGBM asegura que entrenará más rápido, con menor consumo de memoria y con un nivel de accuracy más alto.

5.4. XGBoost

XGBoost implementa algoritmos que son altamente flexibles, XGB es en su interior un árbol en paralelo con boosting, esto mediante la optimización del gradiente. Es la versión "no ligera"de LGBM. [7]

5.5. AdaBoost

El algoritmo Adaptative-Boosting es un ensamble o meta- algoritmo. Adaboost es adaptativo dado que se modifican a favor de instancias clasificadas erróneamente, de esta manera logra accuracys más altos. Sin embargo, parte de su mayor beneficio es un problema, ya que si tenemos valores atípicos es muy sensible a tener malos resultados, justo por esta adaptabilidad.

[5] https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc
[6] https://lightgbm.readthedocs.io/en/latest/Features.html
[7] https://xgboost.readthedocs.io/en/latest/index.html

6. Score para Test

El conjunto Test será evaluado con una métrica especial, debido al marcado desbalance, algunas métricas no son tan confiables ya que al obtener preferencia por una clase, podemos tener valores de hasta 0.86 en accuracy, auc, etc. Es por esto que para el challenge se pide usar la siguiente métrica:

$$Score1 = min(Se, P^+) \tag{2}$$

Dónde:

$$Se = TP/(TP + FN) \tag{3}$$

$$P^+ = TP/(TP + FP) \tag{4}$$

Que básicamente son exhaustividad y precisión respectivamente. Al utilizar el mínimo, aseguramos que al haber preferencia por una clase, la otra automáticamente será menor.

6.1. Modelo Desbalanceado

Para estos modelos utilicé los datos como se terminaron de procesar, en la siguiente tabla podemos ver los valores de AUC para entrenamiento y validación. Como podemos observar, el ganador fue **XGBoost con un Score de 0.4424**

Modelo	Train AUC	Val AUC	Se	P	SCORE1
RandomForest	0.8395	0.5520	0.0393	0.6969	0.0393
XGBoostClf	0.8386	0.7387	0.4855	0.4424	**0.4424**
LogisticRegression	0.8341	0.5847	0.1402	0.6721	0.1402
LGBMC	0.8390	**0.7658**	0.6188	0.4234	0.4234
AdaBoost	0.8343	0.5483	0.0701	0.6949	0.0701

con los siguientes parámetros:
XGBClassifier(colsample_bytree=0.8, eval_metric='auc', gamma=10, learning_rate=0.15, max_depth=4, reg_alpha=0.5, reg_lambda=2, scale_pos_weight=6.239819004524887, seed=442, subsample=0.8) Con la curva ROC que se muestra en **Fig 3**

6.2. Red Neuronal para el caso desbalanceado

Adicionalmente realizé una red neuronal (no entraré en detalles teóricos), la mejor después de más de 16,000 pruebas fue una red secuencial cuya entrada son las 166 variables preprocesadas, una capa oculta densa de 64 neuronas, un Dropout de 0.2 anterior a la capa de salida, ambas con activación sigmoide, función de pérdida de entropía cruzada binaria, optimizada con Adam, con pesos de clase={0:0.35, 1:0.65}, y un tamaño de bache de 32. En el cual obtuvimos los siguientes valores en Test:

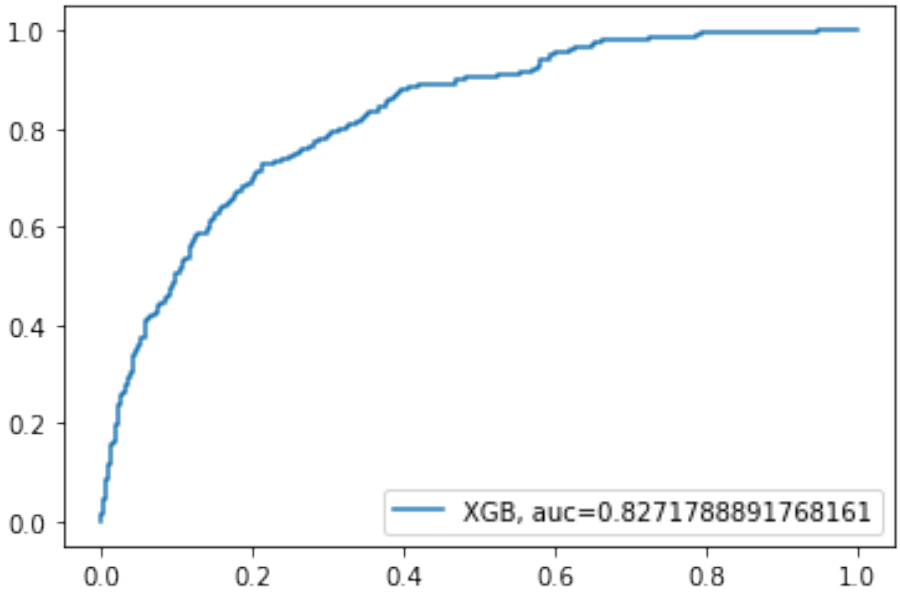

Figura 3. Curva ROC de validación para XGBoost

Se= 0.4513
P=0.4714
Score1= 0.4513

Como podemos ver, con la red obtuvimos el mejor resultado, el cual fue de **Score1= 0.4513** que está por arriba de XGBoost por casi 0.01

7. Modelo con Over Sampling (modelo ganador)

Para esta segunda tanda de modelos utilicé una técnica para balancear las clases, la cual fue Over Sampling con SMOTE.

SMOTE [1] es una técnica para balancear las clases la cual, a diferencia de duplicar datos, crea unos nuevos que estén cercanos a los que ya conocemos que son verídicos, de esta manera crea vecindades al rededor de cada clase uniendo con una linea cerca del mismo espacio y creando ese nuevo punto. De esta manera aseguramos que los valores creados serán lo más parecidos posibles a los que ya tenemos sin que esten repetidos. Los resultados fueron:

Como podemos ver, LogisticRegression es el nuevo máximo con un **SCORE1=0.4615** lo cual lo deja por arriba de todos los modelos.

(l)3-5		Test values		
Modelo	Train AUC	Se	P	SCORE1
RandomForest	0.9492	0.4889	0.3778	0.3778
XGBoostClf	0.9742	0.9555	0.1603	0.1603
LogisticRegression	0.8630	0.4615	0.4963	***0.4615***
LGBMC	0.9737	0.9487	0.1481	0.1481
AdaBoost	0.9565	0.7607	0.1994	0.1994

8. Conclusiones

Por falta de tiempo, la búsqueda de parámetros y arquitectura para la red neuronal no se pudo lograr, sin embargo considero que para un trabajo futuro podría lograr un resultado mejor que el de la regresión.

Fuera de eso, este es el board de la competencia del 2012:

Participant	Score
Alistair Johnson, Nic Dunkley, Louis Mayaud, Athanasios Tsanas, Andrew Kramer, Gari Clifford	0.5353
Luca Citi, Riccardo Barbieri	0.5345
Srinivasan Vairavan, Larry Eshelman, Syed Haider, Abigail Flower, Adam Seiver	0.5009
Martin Macas, Michal Huptych, Jakub Kuzilek	0.4928
Henian Xia, Brian Daley, Adam Petrie, Xiaopeng Zhao	0.4923
Steven L Hamilton, James R Hamilton	0.4872
Natalia M Arzeno, Joyce C Ho, Cheng H Lee	0.4821
Chih-Chun Chia, Gyemin Lee, Zahi Karam, Alexander Van Esbroeck, Sean McMillan, Ilan Rubinfeld, Zeeshan Syed	0.4564
Alexandros Pantelopoulos	0.4544
Deep Bera, Mithun Manjnath Nayak	0.4513

Figura 4. Top ten de competidores.

El Score obtenido con la regresión que fue de **0.4615** nos ubicaría en octavo lugar, lo cual considero que es un resultado bastante aceptable.

Respecto a si mejoramos o no los predictores anteriores (SAPS y SOFA) podemos concluir que se mejoró un 67 % ya que SAPS-I obtiene un SCORE de 0.3125.

Referencias

1. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, Jun 2002.
2. H. A. Chipman, E. I. George, and R. E. McCulloch. Bart: Bayesian additive regression trees. *Ann. Appl. Stat.*, 4(1):266–298, 03 2010.
3. K. C. Dewi, W. F. Mustika, and H. Murfi. Ensemble learning for predicting mortality rates affected by air quality. *Journal of Physics: Conference Series*, 1192:012021, mar 2019.
4. A. L. G. L. H. J. I. P. M. R. M. J. M. G. P. C. Goldberger, A. and H. Stanley. *PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220.* https://physionet.org/, 2000.
5. K. M. L. J. C. K. H. S. H. C. L. Y. K. Y. Kim YH, Yeo JH. Performance Assessment of the SOFA, APACHE II Scoring System, and SAPS II in Intensive Care Unit Organophosphate Poisoned Patients. *JKMS*, 28(12), 12 2013.
6. J.-R. Le Gall, S. Lemeshow, and F. Saulnier. A New Simplified Acute Physiology Score (SAPS II) Based on a European/North American Multicenter Study. *JAMA*, 270(24):2957–2963, 12 1993.
7. L. J. Li Y, Chen Z. *How many people died due to PM2.5 and where the mortality risks increased? A case study in Beijing.* IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2017.
8. A. R. M. Mitchell D, Brockett P and M. K. *Modeling and forecasting mortality rates Insurance: Mathematics and Economics.* 2013.
9. D. Z. Wang G, Lam K M and C. K. *Prediction of mortality after radical cystectomy for bladder cancer by machine learning techniques.* Computers in Biology and Medicine, 2015.