

Personalized Prognostic Models for Oncology: A Machine Learning Approach

David Dooling¹, Angela Kim¹, Barbara McAneny¹, Jennifer Webster¹

1 Innovative Oncology Business Solutions, Albuquerque, NM, USA

* ddooling@innovativeobs.com

Abstract

We have applied a little-known data transformation to subsets of the Surveillance, Epidemiology, and End Results (SEER) publically available data of the National Cancer Institute (NCI) to make it suitable input to standard machine learning classifiers. This transformation properly treats the right-censored data in the SEER data and the resulting Random Forest and Multi-Layer Perceptron models predict full survival curves. Treating the 6, 12, and 60 months points of the resulting survival curves as 3 binary classifiers, the 18 resulting classifiers have AUC values ranging from .765 to .885. Further evidence that the models have generalized well from the training data is provided by the extremely high levels of agreement between the random forest and neural network models predictions on the 6, 12, and 60 month binary classifiers.

Introduction

Opportunities are emerging in many industries today to develop and deploy services that cater to individual needs and preferences. Music aficionados can create their own radio stations from Pandora [1], bibliophiles can receive book recommendations from goodreads.com [2], and Google will provide directions between any two points with warnings of delays in real-time, as well as allowing users to choose mode of transportation [3]. These individualized services leverage large databases to learn and extract information relevant to individuals. A class of techniques that transforms data into actionable information goes by the name of Machine Learning (ML) [4]. ML has recently become a popular method to answer questions and solve problems that are too complex to solve via traditional methods.

The primary objective of this study is to show how ML models can be trained using publically available data to produce personalized survival prognosis curves. The methods presented below can be applied to any type of temporal outcome data, including survival, cost, complication and toxicity data. Traditionally, cancer survival curves have been estimated using Kaplan-Meier methods [5]. Kaplan-Meier methodology also uses large datasets to make predictions, but the resulting curves are summaries for a population and not necessarily relevant or particularly accurate for any given individual. This property of Kaplan-Meier methods is exacerbated when dealing with heterogeneous populations [6]. This capability to provide individualized survival curve prognoses is a direct result of recent advances in computing power and ML algorithms. Similar methodology is becoming commonplace in many industries. These techniques are now infiltrating the healthcare industry.

The Surveillance, Epidemiology, and End Results (SEER) Program of the National Cancer Institute (NCI) program is the most recognized authoritative source of information on cancer incidence and survival in the United States and is the primary data source for this study. SEER currently collects and publishes cancer incidence and survival data from population-based cancer registries covering approximately 28 percent of the US population. The SEER Program has been collecting data since 1973. Intuitively researchers feel confident that this data will surface information crucial to patients and providers, including the relationships between the collected data (demographics, staging, treatment and disease characteristics) and survival outcomes. Though these relationships evade capture by traditional methods, it is possible to surface them with two machine learning techniques known as random forests and neural networks.

One challenge of the SEER data that is shared by many survival datasets is the inclusion of censored data. Observations are labeled censored when the survival information is incomplete. The SEER data contains the number of months each patient survived, as well as the vital status. Traditional methods to deal effectively with this kind of "right-censored data" include Kaplan-Meier curves and Cox Proportional Hazard models [5].

Previous work applying machine learning methods to subsets of the SEER data include creative attempts to deal with the problems presented by right-censored data. Shin et al. [7] use semi-supervised learning techniques to predict 5 year survival, essentially imputing values for SEER records where the survival information is censored at a value less than 5 years. Zolbanin et al. [8] remove all records corresponding to patients who were living but censored within the 60 month study window. This treatment biases the predictions and leads to overly pessimistic predictions.

Previous work applying machine learning methods based on decision trees to survival data in general have a long history, starting with Gordon et al. [9]. A summary of more recent developments concerning survival trees is provided by Bou-Hamad et al. [10]. These methods focus on altering the splitting criteria used in decision tree growth to account for the censoring, and use Kaplan-Meier methods at the resulting nodes for prediction purposes. These methods do not generalize to non-tree-based machine learning algorithms, though Ishwaran et al. have extended the methodology to random survival forests, ensembles of survival trees [11].

Instead of modifying existing learning algorithms, we focus attention on the input data. This approach allows us to take advantage of powerful and rapidly improving machine learning derived discrete classifiers without modification. The essential idea is to recast the problem as a discrete classification problem (predicting the likelihood that a patient is alive in any given month) instead of a regression problem (predicting survival months). Treating months after diagnosis as a discrete feature, the SEER data (or any other right-censored data) can be transformed to make predictions for the hazard function (the probability of dying in the next month, given that the patient has not yet died). The survival function can then be derived from the hazard function.

Materials and Methods

Data preparation and preprocessing

For this study we use the publically available 1973-2012 SEER incidence data files corresponding to colon, breast and lung cancer. These files are listed in subsection Raw SEER datafiles in the Supporting Information Appendix. A great deal of data munging is necessary before using these SEER incidence files as input into machine learning algorithms. The input data was recoded and reshaped to comply with the requirements

of the analysis program. Details are included in subsection Data Preparation Details in the Supporting Information Appendix. Biefly, we transformed the location variables that are given as categorical State and County code pairs to (latitude, longitude, elevation) triples using the Google Maps API, as well as one-hot encoded all categorical variables.

In the SEER data, there is a record for each primary tumor. It multiple records exist for a given patient, only the first chronologically was included. The full set of conditions defining the subsets of the SEER data used in this study is included in the appendix Supporting Information.

Before applying machine learning models trained with these datasets, we describe in detail a method that takes full advantage of all the data, including the right-censored data, and which involves a simple and intuitive transformation, culminating in the full set of features and target variable listed in the appendix Supporting Information.

Transformation of Censored Data for Machine Learning

In this section we describe an inuitive way to transform right-censored data appropriately so that it may be used as input to machine learning algorithms that learn the hazard fuction. The full details of this transformation, and a large inspiration for this study, can be found in this blog post [12].

The key observation is to note that the hazard function at any given time point can be directly learned via standard machine learning methods. It can be rewritten as

$$\lambda(\mathbf{X}_i, t_j) = P(Y = t_j | Y \geq t_j, \mathbf{X}_i), \quad (1)$$

the probability that, if someone has survived up until month t_j , they will die in that month. j runs from 0 to 107, and \mathbf{X}_i corresponds to the single row corresponding to patient i in the original untransformed dataset. 107 months was the maximum value of survival months in all three of the cancer datasets, and is a consequence of the data subsets chosen for this study. Y represents the true, uncensored number of survival months of the patient. What is actually provided in the SEER data is the related variable `SURVIVAL MONTHS` T (how long each subject was in the study), and whether they exited by dying or being censored (D), `VITAL STATUS RECODE` . D is a Boolean variable, so $D = 1$ if $T = Y$, and $D = 0$ if $T < Y$.

It follows directly from equation 1 that

$$P(Y = t_j | \mathbf{X}_i) = \lambda(\mathbf{X}_i, t_j) \prod_{k=1}^{j-1} (1 - \lambda(\mathbf{X}_i, t_k)) \quad (2)$$

Knowing $P(Y = t_j | \mathbf{X}_i)$ for all t_j gives the full probablity distribution of dying at time Y [12]. The survival function is then readily derived from this distribution as

$$S(\mathbf{X}_i, t_k) = 1 - CDF(\mathbf{X}_i, t_k) \quad (3)$$

where $CDF(\mathbf{X}_i, t_k) = \sum_{j=1}^k P(Y = t_j | \mathbf{X}_i)$ is the cumulative density function corresponding to the probability mass function in equation [13].

Treating T as just another covariate is the key to the transformation. Each datapoint in the hidden classification problem is the combination of an \mathbf{X}_i in the orginal dataset plus some month t_j , and the classification problem is "did point \mathbf{X}_i die in month t_j ." We will call this new variable D_{ij} (`newtarget`). We can transform our original data set into a new one, with one row for each month that each \mathbf{X}_i is in the sample; train a standard classifier on this new dataset with D_{ij} as the target, and derive a survival model from the orginal dataset. Psuedocode for this transformation is found in the appendix Supporting Information.

Explicit examples will help make this transformation clear. The untransformed records represented in Table (1) are transformed to the multiple records shown in Table (2).

Table 1. Example of four columns in the untransformed dataset.

	cs_tumor_size	year_of_birth	survival_months	vital_status_recode_Death
newindex				
205	60	1951	3	1
306	40	1950	3	0

Table 2. Example of four columns in the transformed dataset.

	cs_tumor_size	year_of_birth	month	newtarget
newindex				
205	60	1951	0	0
205	60	1951	1	0
205	60	1951	2	0
205	60	1951	3	1
306	40	1950	0	0
306	40	1950	1	0
306	40	1950	2	0
306	40	1950	3	0

One obvious side effect of this transformation is that it increases the length of the dataset. For this study, the original, untransformed colon cancer DataFrame has shape (113072, 103), and the total transformed colon cancer DataFrame has shape (4165251, 103). Similarly, the original, untransformed lung cancer DataFrame has shape (177089, 115), and the total transformed lung cancer DataFrame has shape (3079931, 115). The biggest increase in dataset size occurred with the breast cancer data, which is a consequence of the relatively high survival rates in breast cancer. A subject who is censored with a recorded survival months of 48 will contribute 49 rows to the transformed dataset. The original, untransformed breast cancer DataFrame has shape (329949, 67), and the total transformed breast cancer DataFrame has shape (15085711, 67). Training machine learning algorithms on such large datasets, even after splitting into training and testing sets described below, requires large RAM. All computations for this study were performed on a Dell XPS 8700 Desktop with 32GB of RAM. The training times involved in the classification task of learning the hazard function $\lambda(\mathbf{X}_i, t_j)$ for the chosen model parameters were on the order of a few hours or less, but the evaluation of the AUC performance metrics associated with the 6, 12, and 60 month binary survival classifiers took more than 24 hours for the random forest models. These AUC performance metrics provided the feedback mechanism to adjust the model hyperparameters.

Training and Test Partitions

The datasets were split into training and test sets at the patient level, with 97% of patients assigned to the training set, and the remaining 3% of patients assigned to the test set. All records corresponding to a given patient were assigned exclusively to either the training or test set. This choice of an unusually low percentage of data in the test set was made for two reasons. The performance metrics described in sec Performance

Metrics for the given choice of training and test partition of the data took well over 24 hours for the random forest models; choosing the convention 80/20 split would have resulted in a prohibitively long times for the training-performance metric feedback loop. Because of the large size of the data set, this choice of training and test partition still leads to an acceptably large test set for the purposes of model evaluation. An additional characteristic of this transformed data that requires careful treatment involves balancing. The transformation results in many new records with the target variable `newtarget == 0`. The training and test sets must be chosen such that the ratio of the number of records with `newtarget == 0` to that of the number of records with `newtarget == 1` is the same in the training and test datasets. This ratio turns out to be ≈ 396 for the breast cancer data, ≈ 99 for the colon cancer data, and ≈ 22.75 for the lung cancer data. The shapes of the training and testing datasets for breast cancer used in this study are (14936862, 67) and (148849, 67), respectively. For lung cancer, the corresponding datasets have shapes (2988768, 115) and (91163, 115). Finally, for colon cancer the partition into training and test datasets of the transformed data have the shapes (3958008, 103) and (207243, 103). Multiple rows correspond to the same test patient in these datasets. The colon cancer test dataset represents 5654 distinct subjects; the breast cancer test dataset represents 3300 distinct subjects; and the lung test dataset contains data for 5313 distinct subjects.

The models described below are trained to learn the values of `newtarget`, which is a binary variable: a value of '0' indicating that the subject is still alive at the given month, while a value of '1' indicates that the patient died at that particular value of `months`. The random forests and neural networks described below are binary classifiers with the target `newtarget`. Both the random forests and neural networks are capable of not only performing strict class prediction, i.e. predicting whether `newtarget` is '0' or '1', but are also able to predict the *probability* of `newtarget` being '0' or '1', and are thus able to learn the hazard function.

Prediction Models

With the datasets transformed as described above, we are now able to use them to train and evaluate machine learning classifiers. The classifier models described in this section are learning the hazard function: given all of the data given in the appendix Supporting Information for each cancer type, which includes the field `months` (the months after diagnosis), the models predict the target variable `newtarget`, which is a binary class label equal to 1 if the subject died in that month and 0 otherwise.

From the hazard function for each unique patient, we can construct the survival function as in Equation 3. The relevant python code is available at the github repository containing supplemental material for this study [14]. For each subject i , all input data minus `months` and `newtarget` is represented by \mathbf{X}_i . After the classifier models have trained with target `newtarget` on the (very large) training set, each subject's survival function is computed in the corresponding (much smaller) test set. These functions are computed by using the model to predict $\lambda(\mathbf{X}_i, t_j)$ for j running from 0 to 107 months, and \mathbf{X}_i corresponds to the single row corresponding to subject i in the original untransformed dataset. 107 months was the maximum value of survival months in all three of the cancer datasets, and is a consequence of the data subsets chosen for this study.

Decision Trees and Random Forests *Decision tree* classifiers are attractive models because they can be interpreted easily. Like the name decision tree suggests, we can think of this model as breaking down our data by making decisions based on asking

a series of questions. Based on the features in our training set, the decision tree model learns a series of questions to infer the class labels of the samples.

Random forests have gained huge popularity in applications of machine learning during the last decade due to their good classification performance, scalability, and ease of use. Intuitively, a random forest can be considered as an *ensemble of decision trees*. The idea behind ensemble learning is to combine *weak learners* to build a more robust model, a *strong learner*, that has a better generalization error and is less susceptible to overfitting.

The goal behind *ensemble methods* is to combine different classifiers into a meta-classifier that has a better generalization performance than each individual classifier alone. For example, assuming that we collected predictions from 10 experts, ensemble methods would allow us to strategically combine these predictions by the 10 experts to come up with a prediction that is more accurate and robust than the predictions by each individual expert. The individual decision trees that make an ensemble are called base learners, and as long as the error rate of each base learner is less than .50, the combined random forest will benefit from the affects of combining predictions to achieve a far greater accuracy.

A big advantage of random forests is that honing in on suitable hyperparameter values (the number of trees in the forest, the depth of each decision tree, the specific measure of information gain used to choose the node splitting, etc) is not very difficult. The ensemble method is robust to noise from the individual decision trees, which helps to prevent overfitting (memorizing the training dataset targets instead of generalizing from learned rules to perform successfully on unseen data). The only parameter that has a clearly noticeable effect on performance is the number of trees to include in the forest; in general, the more trees the better the performance, but there is a price to pay in terms of computational cost. The number of trees for the forests trained in this study was relatively small, 20 trees for breast cancer and 25 for both the lung and colon cancer models. We have used the Python scikit-learn implementation of the Random Forest machine learning classifier [15]. Random Forests are frequent winners of the Kaggle machine learning competitions [16]. The model parameters for each cancer type are given in the appendix Supporting Information.

Multi-Layer Perceptron Neural Networks Neural networks are a biologically-inspired programming paradigm that enable computers to learn from observational data [17]. Neural networks are a hot topic not only in academic research, but also in big technology companies such as Facebook, Microsoft, and Google who invest heavily in artificial neural networks and deep learning research. As of today, complex neural networks powered by deep learning algorithms are considered to be the state-of-the-art when it comes to complex problem solving such as image and voice recognition. In addition, the pharmaceutical industry recently started to use deep learning techniques for drug discovery and toxicity prediction, and research has shown that these novel techniques substantially exceed the performance of traditional methods for virtual screening [18].

We have used the Multi-Layer Perceptron Neural Network (MLP neural network) implementation Keras developed at MIT. Keras was initially developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System) [19]. Keras is a minimalist, highly modular neural networks library, written in Python and capable of running on top of either TensorFlow or Theano. The model architecture for each cancer type are given in the appendix Supporting Information.

Results

In order to evaluate the performance of the models, we first construct three binary classifiers corresponding to whether or not a subject survived 6, 12, or 60 months after diagnosis. We iterate over all distinct patient indices in the test set, compute the predicted survival function, and capture the values corresponding to 6, 12, and 60 months. If the survival function evaluated at 6 months is greater than or equal to .5 for a given patient, then the 6 months binary classifier predicts that that patient will be alive 6 months after diagnosis. Similarly, if the survival function evaluated at 12 months is less than .5, then the 12 months binary classifier predicts that that subject will be dead 12 months after diagnosis. Figure (1) illustrates the method; in this case the 6-month and 12-month classifiers predict survival, while the 60-month classifier predicts death.

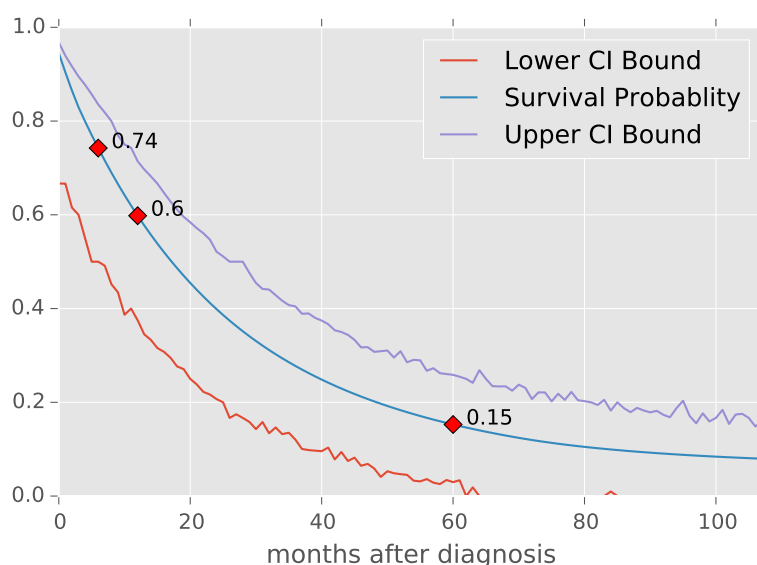


Figure 1. Example of the construction of the binary classifiers for 6, 12, and 60 months survival. A patient's hazard curve $\lambda(\mathbf{X}_i, t_j)$ is predicted by the model for times out to 107 months. The survival curve is then readily computed as in Equation (3). For this example, the 6-month and 12-month classifiers predict survival, while the 60-month classifier predicts death.

Because of censoring it is necessary to apply some Boolean filters to the data in order to correctly assess the resulting classifiers. To construct AUC curves for the 6 month classifier, we restrict ourselves to considering subjects in the test data where either of the following mutually exclusive conditions holds:

- `survival_months >= 6 AND vital_status_recode == 0`
- `vital_status_recode == 1`

That is, we restrict ourselves to subsets of the data where we know for certain whether or not the subject survived at least 6 months. Similarly for the 12 and 60 months survival classifiers.

Survival Curve Error Estimates The following bootstrap method was used to calculate the upper and lower bounds corresponding to 95% confidence intervals. From

equation 3, we can obtain the cumulative distribution function (CDF) associated with each individual survival curve. We then sample from this CDF in a way that reflects the underlying data used to produce the model. The training data used to create the model has an underlying distribution of survival months. In the transformed training dataset, each subject contributes as many rows as the number of survival months plus one (patients with zero survival months still contribute one row to the training data). A patient that survived 50 months contributes 51 "points" to the training of the model. If all patients lived out to 107 months, the model would contain less uncertainty. This observation leads to the following algorithm for determining the error estimates to the predicted survival curves:

- compute the CDF associated with the survival curve
- use the underlying training data CDF of survival months to choose the number of points to draw from the survival curve CDF, and compute a new survival curve
- Repeat the previous step 10,000 times and collect the curves into a list. Changing the number of curves affects how smooth the upper and lower bounds are, but does not affect the interval size between for each month.
- extract for each month from the list of curves the .975 and .025 percentiles to record the values for the upper and lower curves

The process is analogous to the following hypothetical situation. Imagine a patient going to an expert to get a survival prognosis. After collecting data on the patient and keeping records, the expert predicts the central, single survival curve. The patient then seeks multiple "second opinions." These second opinions are generated not from independent examinations of the patient, but by outside experts sampling from the data already collected by the first expert. Then the predictions of 95% of these 10,000 experts all fall within the band determined by the upper and lower curves.

Performance Metrics

AUC scores The AUC scores for each of the 18 different binary classifiers are listed in Table (3). The lowest AUC in Table 3 is .765, corresponding to the lung neural network model predictions for 6 months survival, while the highest AUC in Table 3 is .885, corresponding to the breast random forest model predictions for 12 months survival.

Table 3. AUC values for the Random Forest and Neural Networks model binary classifiers derived from the full survival curve predictions; see text for details. The number of subjects that were used in the calculation of a given AUC score are given in parenthesis after the score.

Model	6 Months AUC	12 Months AUC	60 Months AUC
Breast RF	.846 (3035)	.885 (2797)	.844 (1392)
Breast NN	.855 (3035)	.867 (2797)	.836 (1392)
Colon RF	.804 (5281)	.806 (5003)	.828 (3232)
Colon NN	.797 (5281)	.804 (5003)	.841 (3232)
Lung RF	.772 (5019)	.796 (4860)	.874 (4143)
Lung NN	.765 (5019)	.796 (4860)	.875 (4143)

Model Agreement An additional means of validating the predictions of these models is by comparing their predictions to each other for the same set of input data. Table 4 shows the strong agreement between the random forest and neural network classifiers for each cancer type. Python code showing how the values in Table 4 are

computed is available in the files `NewPatientBreastCF.html` ,
`NewPatientColonCF.html` , and `NewPatientLung.html` in the GitHub repository
containing supplemental material for this study [14]. Table 4 is computed as follows.
For each cancer type (breast,colon, and lung), do the following:

- use the corresponding Random Forest and Neural Network models to compute the survival curves for all of the test subjects
- extract the values of the survival curve evaluated for 6, 12, and 60 months for both models
- if both models predict less than .5 or both models predict greater than or equal to .5, that counts as agreement
- otherwise, the models disagree

Fig. (2) shows scatter plots of the neural network and random forest predictions for 6, 12, and 60 month survival prediction for each cancer type. The correlations of the neural network and random forest predictions are listed in Table (5). Event though breast cancer 6 month survival has the highest percentage agreement between the two classifiers (over 99%), it also has the lowest correlation (.676). This seeming contradiction is a result of the high 6-month survival rate for breast cancer, and that the random forest and neural network models seem to treat this case differently. The neural network models in general appear to be less pessimistic in general than the random forest model predictions.

The high level of agreement between two models lends confidence to the notion that they have both learned from the training data and are generalizing well.

Table 4. Percentage agreement for the Random Forest and Neural Network classifiers for 6, 12, and 60 month survival predictions on the test data for each cancer type.

Cancer Type	% agreement 6 months	% agreement 12 months	% agreement 60 months
Colon	.981	.971	.915
Breast	.994	.984	.938
Lung	.861	.883	.900

Table 5. Correlations between the Random Forest and Neural Network classifiers for 6, 12, and 60 month survival predictions on the test data for each cancer type. The corresponding scatter plots are shown in Fig. (2).

Cancer Type	6 months	12 months	60 months
Lung	.914	.946	.921
Colon	.900	.923	.935
Breast	.676	.72	.785

Survival Curve Prediction Apps

The six models have their full hyperparameter and architecture presented in the appendix section Supporting Information. Python code for all six model training and evaluation is available at the github repository containing supplemental material for this study [14].

Using the popular Flask microframework for web applications [20], we have made web applications corresponding to the six models. The list of web applications below will allow readers to freely experiment with the models.

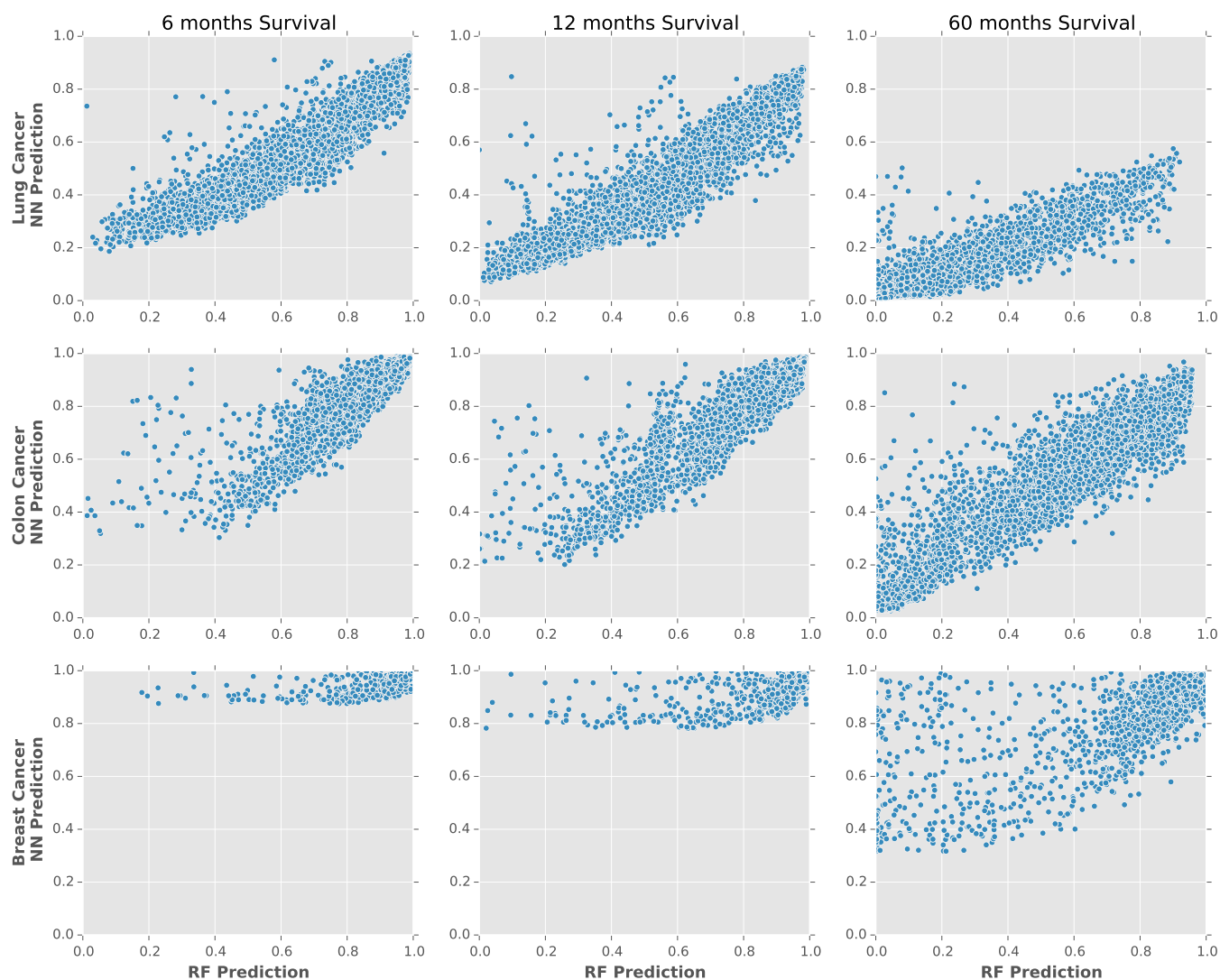


Figure 2. Scatter plots showing the correlations between the MLP model's prediction and RF model's prediction for the probability of surviving at least 6, 12, and 60 months for the lung, colon and breast cancer test data.

1. breast cancer 326
 - (a) random forest: 327
<https://github.com/doolingdavid/breast-cancer-rf-errors.git> 328
 - (b) neural network: 329
<https://github.com/doolingdavid/breast-cancer-nn-errors.git> 330
2. lung cancer 331
 - (a) random forest: 332
<https://github.com/doolingdavid/lung-cancer-rf-errors.git> 333
 - (b) neural network: 334
<https://github.com/doolingdavid/lung-cancer-nn-errors.git> 335
3. colon cancer 336
 - (a) random forest: 337
<https://github.com/doolingdavid/colon-cancer-rf-errors.git> 338
 - (b) neural network: 339
<https://github.com/doolingdavid/colon-cancer-nn-errors.git> 340

After downloading the .zip file associate with one of the above web applications, and assuming python is installed on your system, you can launch the application by running 341

```
>python hello.py 342
```

and pointing the browser to the local server: `http://127.0.0.1:5000` , or 343

`http://localhost:5000` . 344

These machine learning models are used to predict survival curves for a given set of input data. The resulting survival curves predict the probability that a patient with the given input data will survive at least up to month x . 345

For example, using the Colon Cancer neural network app, and inputting the values listed in Table (6) results in the survival curve depicted in Figure (3); the predicted probabilities of living at least 6, 12, and 60 months are .89, .83, and .50, respectively. 346

Table 6. Example input data to the Colon Cancer neural network app

<https://github.com/doolingdavid/colon-cancer-nn-errors.git>.

Variable	Value
What is the tumor size (mm)	300
What is the patient's address?	boston massachusetts
Grade	moderately differentiated
Histology	adenomas and adenocarcinomas
Laterality	not a paired site
Marital Status at Dx	Single, never married
Month of Diagnosis	Jan
How many primaries	1
Race_ethnicity	White
seer_historic_stage_a	Regional
Gender	Male
spanish_hispanic_origin	Non-spanish/Non-hispanic
Year of Birth	1940
Year of Diagnosis	2010

Changing the data in Table 6 so that the address field is changed from Boston, Massachusetts to Denver, Colorado but keeping all other variables are unchanged results 352

Colon Cancer Survival Curve Prediction

Prediction:

1. Probability of Surviving 6 months is **0.897**
2. Probability of Surviving 12 months is **0.831**
3. Probability of Surviving 60 months is **0.504**

Predicted Survival Curve from Model

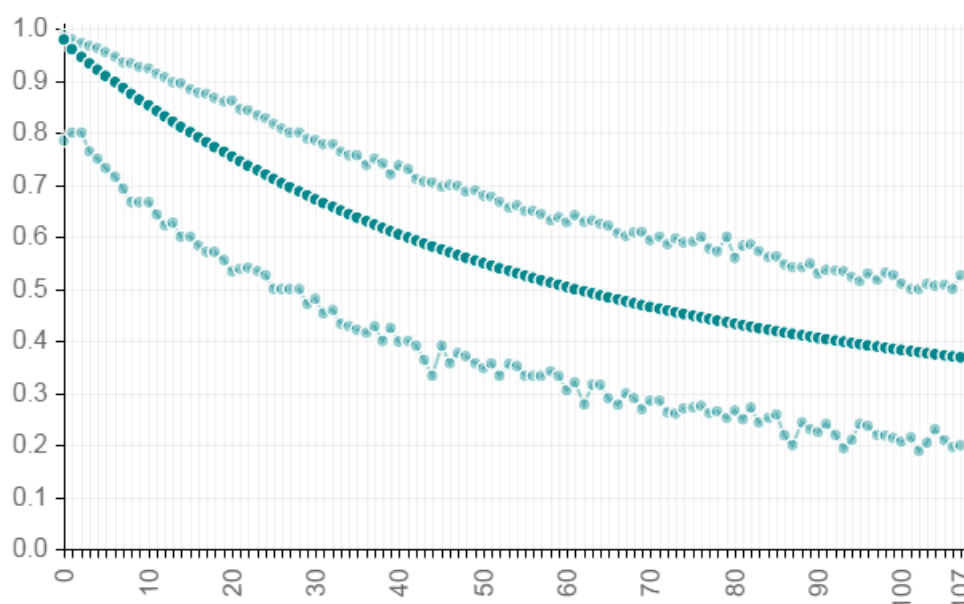


Figure 3. Colon Cancer Survival Curve predicted from the data in Table (6) using the neural network web app
<https://github.com/doolingdavid/colon-cancer-nn-errors.git>.

in the predicted probabilities of living at least 6, 12, and 60 months: .945, .902, .665. Behind the scenes, the apps use the input to the address field to make a call to the Google Maps API to convert the address into a latitude, longitude and elevation. These probabilities are noticeably higher and reflect the documented effects of both longitude and elevation on cancer treatment and prognosis in the United States [21].

A similar example of how changing the inputs to the models affects the predicted survival curves in interesting ways can be seen with the random forest model for lung cancer. Changing the data in Table 7 by toggling between the male/female, and married/single four possible permutations results in the following prediction probabilities for 6, 12, and 60 month survival:

- male/married: .53, .27, .01
- male/single: .35, .18, .009
- female/married: .55, .31, .01
- female/single: .50, .27, .01

Inputting the same combinations of data into the lung cancer neural network app

Table 7. Example input data to the Lung Cancer random forest app
<https://github.com/doolingdavid/lung-cancer-rf-errors.git>.

Variable	Value
What is the tumor size (mm)	500
What is the patient's address?	newark new jersey
Grade	well differentiated
Histology	acinar cell neoplasms
Laterality	bilateral involvement, lateral origin unknown; stated to be single primary
Marital Status at Dx	Married including common law
Month of Diagnosis	Jan
How many primaries	1
Race_ethnicity	White
seer_historic_stage_a	Distant
Gender	Female
spanish_hispanic_origin	Non-spanish/Non-hispanic
Year of Birth	1970
Year of Diagnosis	2011

<https://github.com/doolingdavid/lung-cancer-nn-errors.git> yields the following probabilities:

- male/married: .42, .24, .04
- male/single: .40, .22, .03
- female/married: .44, .26, .04
- female/single: .42, .24, .04

It is interesting to note that both the random forest and neural network lung cancer models predict greater 6 month survival rates for married people, with a slightly greater benefit for males than females. The effect is greater in the random forest model, but is also visible in the neural network model.

Discussion

The purpose of this study has been twofold; to develop a general methodology of data transformation to survival data with censored observations so that machine learning algorithms can be applied and to apply the methodology to create models of personalized survival curve prognosis. To help further refine the methodology, we would like to apply it to different survival datasets [22], not necessarily within the healthcare domain. In particular, the methods presented in this paper do not take into account time varying features. For example, the `cs_tumor_size` variable that has been a part of this study is kept fixed at the value measured at diagnosis for all records corresponding to a given subject. Clearly, the actual tumor size varies along with time and a sophisticated model can be developed to take this into account, given available datasets.

The SEER database has been linked with claims data in the SEER-Medicare Linked Database [23]. This linkage allows for the identification of additional clinical data for each record in the SEER database and allows for an enrichment of the models presented in this study, and is an avenue for further investigation.

An additional avenue of research concerns the broad concept of causality. As demonstrated in section Survival Curve Prediction Apps, there appears to be a

correlation between marital status and survival prognosis. Does this mean that if a single person in Boston, Massachusetts is diagnosed with cancer, that they should immediately get married and move to Denver? Of course not. But personal discussions with providers has confirmed for one of the authors (D.D.) that married males tend to be much more diligent in following instructions than their single counterparts. What appears to be in effect is that some of the SEER data is providing an identifiable signature of underlying causes not directly represented by the data. Latent variables not directly seen in the data are still providing echos of patterns in the data and the sheer volume allows us to see glimpses of these patterns. Marital status is in some instances a surrogate for the presence of a strong social structure and support group surrounding a patient, which presence presumably leads to more desirable survival prognosis. The daunting and exciting task of teasing out actual causality relationships within machine learning contexts has been pioneered by Judea Pearl of the University of California, Los Angeles ¹ and seems particularly relevant and applicable to censored survival data. Combining the methodology presented in this study with that of the pioneering work of Judea Pearl on causality will be a fruitful avenue for future research.

Supporting Information

Raw SEER datafiles

- incidence\yr1973_2012.seer9\COLRECT.txt
- incidence\yr1973_2012.seer9\BREAST.txt
- incidence\yr1973_2012.seer9\RESPIR.txt
- incidence\yr1992_2012.sj_la_rg_ak\COLRECT.txt
- incidence\yr1992_2012.sj_la_rg_ak\BREAST.txt
- incidence\yr1992_2012.sj_la_rg_ak\RESPIR.txt
- incidence\yr2000_2012.ca_ky_lo_nj_ga\COLRECT.txt
- incidence\yr2000_2012.ca_ky_lo_nj_ga\BREAST.txt
- incidence\yr2000_2012.ca_ky_lo_nj_ga\RESPIR.txt
- incidence\yr2005_lo_2nd_half\COLRECT.txt
- incidence\yr2005_lo_2nd_half\BREAST.txt
- incidence\yr2005_lo_2nd_half\RESPIR.txt

Data Preparation Details

A preprocessing step common to each of the three cancer types studied involves the SEER STATE-COUNTY RECODE variable. The STATE-COUNTY RECODE field is a state-county combination where the first two characters represent the state FIPS code and the last three digits represent the FIPS county code. The FIPS code is a five-digit Federal Information Processing Standard (FIPS) code which uniquely identifies counties and county equivalents in the United States, certain U.S. possessions, and certain freely associated states. This particular field illustrates an important characteristic of machine learning, that is, the difference between *categorical features* and *numeric features*. All input into a machine learning algorithm must be numeric, but real numbers carry with them the usually extremely useful property known as the well-ordering property. Machine learning algorithms use the well-ordering property of the real numbers to learn. But if one is tasked with encoding a categorical feature into suitable numeric format for machine learning, it is necessary to do so in a way that removes the well-ordering property. Categorical variables are commonly encoded using one-hot encoding, in which

¹Judea Pearl homepage at the University of California, Los Angeles, http://bayes.cs.ucla.edu/jp_home.html, accessed 11 Jan 2016.

the explanatory variable is encoded using one binary feature for each of the variable's possible values [24].

One-hot encoding needs to be applied to all of the nominal categorical variables in the SEER data that we wish to include in our predictive models. In particular, in order to include the geophgraphical information contained in the SEER categorical variable `STATE-COUNTY RECODE`, it becomes necessary to create a new feature variable for each of the distinct (state,county) pairs in the data. In the United States, there are approximately 3,000 counties. Clearly, transforming the `STATE-COUNTY RECODE` data representation into distinct (state_county) columns will explode the dataset to become wider than is optimal for machine learning. Adding extra columns to your dataset, making it wider, requires more data rows (making it taller) in order for machine learning algorithms to effectively learn [24]. Because one-hot coding `STATE-COUNTY RECODE` would cause such drastic shape changes in our data, we wish to avoid doing so. Fortunately, this variable, though given as a categorical variable, is actually a recode for three ordinal variables. There is an ordering among the (state_county) columns, namely longitude, latitude, and elevation. We can transform the data in `STATE-COUNTY RECODE` into three new numerical columns: `lat`, `lng`, and `elevation`.

For example, Table (8) shows how five entries of `STATE-COUNTY RECODE` corresponding to counties within New Mexico can be represented by the `elevation`, `lat`, and `lng` features.

Table 8. Example of the transformation of STATE-COUNTY RECODE to elevation, lat, and lng.

STATE-COUNTY RECODE	address	elevation	lat	lng
35001	Bernalillo+county+NM	5207.579772	35.017785	-106.629130
35003	Catron+county+NM	8089.242628	34.151517	-108.427605
35005	Chaves+county+NM	3559.931671	33.475739	-104.472330
35006	Cibola+county+NM	6443.415570	35.094756	-107.858387
35007	Colfax+county+NM	6147.749089	36.579976	-104.472330

It is a simple exercise to construct the full lookup table from the SEER `STATE-COUNTY RECODE` variable to the corresponding three values `elevation`, `lat`, and `lng`. We use the publically available datafile from the United States Census Bureau [25] to map the state FIPS and county FIPS codes to query strings like those in the `address` field in Table (8). It is then possible to programmatically query the Google Maps Geocoding API for the latitude and longitude [26], and the Google Maps Elevation API for the corresponding elevation [27]. An added benefit of this shift from the single categorical variable `STATE-COUNTY RECODE` to the three continuous numerical variables `lat`, `lng`, and `elevation` is that input into the web applications described later are not restricted to the states and counties covered in the SEER registries; in fact, the input to the models can be any address you would enter into Google Maps and calls to the Google Maps Geocoding API and the Google Maps Elevation API provide the conversion from the address string to the input variables `lat`, `lng`, and `elevation`. The full lookup table analogous to Table (8) is available from a GitHub repository containing supplemental information for this study [14].

Data Subsets

The four COLRECT.txt files were imported into a pandas DataFrame object. This data was then filtered according to the conditions in Table (9). The RESPIR.txt and

BREAST.txt files were imported into separate dataframes in similar fashion and filtered according to the conditions in Table (10) and Table (11), respectively. The SEER variable CS TUMOR SIZE records the tumor size in millimeters if known. But if not known, CS TUMOR SIZE is given as '999', to indicate that the tumor size is "Unknown; size not stated; not stated in pateint record." In this study, we discard those records, as indicated in Tables (11, 9, 10).

Table 9. Filters applied to the Colon Cancer data.

Column	Filter
SEQUENCE NUMBER-CENTRAL	≠ "Unspecified"
AGE AT DIAGNOSIS	≠ "Unknown age"
BIRTHDATE-YEAR	≠ "Unknown year of birth"
YEAR OF DIAGNOSIS	≥ 2004
SURVIVAL MONTHS FLAG	= "1"
CS TUMOR SIZE EXT/EVAL	≠ ""
CS TUMOR SIZE	≠ 999
SEER RECORD NUMBER	= 1
PRIMARY SITE	= "LARGE INTESTINE, (EXCL. APPENDIX)"
SEQUENCE NUMBER-CENTRAL	= 0

Table 10. Filters applied to the Lung Cancer data.

Column	Filter
SEQUENCE NUMBER-CENTRAL	≠ "Unspecified"
AGE AT DIAGNOSIS	≠ "Unknown age"
BIRTHDATE-YEAR	≠ "Unknown year of birth"
YEAR OF DIAGNOSIS	≥ 2004
SURVIVAL MONTHS FLAG	= "1"
CS TUMOR SIZE EXT/EVAL	≠ ""
CS TUMOR SIZE	≠ 999
SEER RECORD NUMBER	= 1
PRIMARY SITE	= "LUNG & BRONCHUS"
SEQUENCE NUMBER-CENTRAL	= 0

The following categorical features were one-hot encoded for each of the three datasets:

- SEX ,
- MARITAL STATUS AT DX ,
- RACE/ETHNICITY ,
- SPANISH/HISPANIC ORIGIN ,
- GRADE ,
- PRIMARY SITE ,
- LATERALITY ,
- SEER HISTORIC STAGE A ,
- HISTOLOGY RECODE--BROAD GROUPINGS ,

Table 11. Filters applied to the Breast Cancer data.

Column	Filter
SEQUENCE NUMBER-CENTRAL	≠ "Unspecified"
AGE AT DIAGNOSIS	≠ "Unknown age"
BIRTHDATE-YEAR	≠ "Unknown year of birth"
YEAR OF DIAGNOSIS	≥ 2004
SURVIVAL MONTHS FLAG	= "1"
CS TUMOR SIZE EXT/EVAL	≠ " "
CS TUMOR SIZE	≠ 999
SEER RECORD NUMBER	= 1
SEQUENCE NUMBER-CENTRAL	= 0

- MONTH OF DIAGNOSIS , 498
- VITAL STATUS RECODE , 499

and the STATE-COUNTY RECODE variable was dropped and replaced with the 500
elevation , lat , and lng variables for all three datasets as illustrated in Table (8). 501

Colon Cancer Feature Selection 502

The feature set used as input into both the Random Forest and Neural Network models, 503
after the transformation described in section Transformation of Censored Data for 504
Machine Learning is given below and also available in full detail in the file 505
NewPatientColonML.html . 506

- cs.tumor_size 507
- elevation 508
- grade_cell type not determined 509
- grade_moderately differentiated 510
- grade_poorly differentiated 511
- grade_undifferentiated; anaplastic 512
- grade_well differentiated 513
- histology_recode_broad.groupings_acinar cell neoplasms 514
- histology_recode_broad.groupings_adenomas and adenocarcinomas 515
- histology_recode_broad.groupings_blood vessel tumors 516
- histology_recode_broad.groupings_complex epithelial neoplasms 517
- histology_recode_broad.groupings_complex mixed and stromal neoplasms 518
- histology_recode_broad.groupings_cystic, mucinous and serous neoplasms 519
- histology_recode_broad.groupings_ductal and lobular neoplasms 520
- histology_recode_broad.groupings_epithelial neoplasms, NOS 521
- histology_recode_broad.groupings_fibromatous neoplasms 522
- histology_recode_broad.groupings_germ cell neoplasms 523
- histology_recode_broad.groupings_lipomatous neoplasms 524
- histology_recode_broad.groupings_miscellaneous bone tumors 525
- histology_recode_broad.groupings_myomatous neoplasms 526
- histology_recode_broad.groupings_neuroepitheliomatous neoplasms 527
- histology_recode_broad.groupings_nevi and melanomas 528
- histology_recode_broad.groupings_paragangliomas and glumus tumors 529
- histology_recode_broad.groupings_soft tissue tumors and sarcomas, NOS 530

• histology_recode_broad_groupings_squamous cell neoplasms	531
• histology_recode_broad_groupings_synovial-like neoplasms	532
• histology_recode_broad_groupings_transistional cell papillomas and carcinomas	533
• histology_recode_broad_groupings_unspecified neoplasms	534
• lat	535
• laterality_Left: origin of primary	536
• laterality_Not a paired site	537
• laterality_Only one side involved, right or left origin unspecified	538
• laterality_Paired site, but no information concerning laterality; midline tumor	539
• laterality_Right: origin of primary	540
• lng	541
• marital_status_at_dx_Divorced	542
• marital_status_at_dx_Married (including common law)	543
• marital_status_at_dx_Separated	544
• marital_status_at_dx_Single (never married)	545
• marital_status_at_dx_Unknown	546
• marital_status_at_dx_Unmarried or domestic partner	547
• marital_status_at_dx_Widowed	548
• month_of_diagnosis_Apr	549
• month_of_diagnosis_Aug	550
• month_of_diagnosis_Dec	551
• month_of_diagnosis_Feb	552
• month_of_diagnosis_Jan	553
• month_of_diagnosis_Jul	554
• month_of_diagnosis_Jun	555
• month_of_diagnosis_Mar	556
• month_of_diagnosis_May	557
• month_of_diagnosis_Nov	558
• month_of_diagnosis_Oct	559
• month_of_diagnosis_Sep	560
• number_of primaries	561
• race_ethnicity_Amerian Indian, Aleutian, Alaskan Native or Eskimo	562
• race_ethnicity_Asian Indian	563
• race_ethnicity_Asian Indian or Pakistani	564
• race_ethnicity_Black	565
• race_ethnicity_Chinese	566
• race_ethnicity_Fiji Islander	567
• race_ethnicity_Filipino	568
• race_ethnicity_Guamanian	569
• race_ethnicity_Hawaiian	570
• race_ethnicity_Hmong	571
• race_ethnicity_Japanese	572
• race_ethnicity_Kampuchean	573
• race_ethnicity_Korean	574
• race_ethnicity_Laotian	575
• race_ethnicity_Melanesian	576
• race_ethnicity_Micronesian	577
• race_ethnicity_New Guinean	578
• race_ethnicity_Other	579
• race_ethnicity_Other Asian	580
• race_ethnicity_Pacific Islander	581
• race_ethnicity_Pakistani	582

- race_ethnicity_Polynesian 583
- race_ethnicity_Samoan 584
- race_ethnicity_Thai 585
- race_ethnicity_Tongan 586
- race_ethnicity_Unknown 587
- race_ethnicity_Vietnamese 588
- race_ethnicity_White 589
- seer_historic_stage_a_Distant 590
- seer_historic_stage_a_In situ 591
- seer_historic_stage_a_Localized 592
- seer_historic_stage_a_Regional 593
- seer_historic_stage_a_Unstaged 594
- sex_Female 595
- spanish_hispanic_origin_Cuban 596
- spanish_hispanic_origin_Dominican Republic 597
- spanish_hispanic_origin_Mexican 598
- spanish_hispanic_origin_Non-Spanish/Non-hispanic 599
- spanish_hispanic_origin_Other specified Spanish/Hispanic origin (excludes Dominican Repuclic) 600
- spanish_hispanic_origin_Puerto Rican 601
- spanish_hispanic_origin_South or Central American (except Brazil) 602
- spanish_hispanic_origin_Spanish surname only 603
- spanish_hispanic_origin_Spanish, NOS; Hispanic, NOS; Latino, NOS 604
- spanish_hispanic_origin_Uknown whether Spanish/Hispanic or not 605
- year_of_birth 606
- year_of_diagnosis 607
- month 608

and `newtarget` is the target variable, indicating whether or not the subject died in month given by the value of the `month` variable. 610

Lung Cancer Feature Selection 612

The feature set used as input into both the Random Forest and Neural Network models, after the transformation described in section Transformation of Censored Data for Machine Learning is given below and also available in full detail in the file `NewPatientLungML.html` . 613
614
615
616

- cs_tumor_size 617
- elevation 618
- grade_cell type not determined 619
- grade_moderately differentiated 620
- grade_poorly differentiated 621
- grade_undifferentiated; anaplastic 622
- grade_well differentiated 623
- histology_recode_broad_groupings_acinar cell neoplasms 624
- histology_recode_broad_groupings_adenomas and adenocarcinomas 625
- histology_recode_broad_groupings_blood vessel tumors 626
- histology_recode_broad_groupings_complex epithelial neoplasms 627
- histology_recode_broad_groupings_complex mixed and stromal neoplasms 628
- histology_recode_broad_groupings_cystic, mucinous and serous neoplasms 629
- histology_recode_broad_groupings_ductal and lobular neoplasms 630
- histology_recode_broad_groupings_epithelial neoplasms, NOS 631

• histology_recode_broad_groupings_fibroepithelial neoplasms	632
• histology_recode_broad_groupings_fibromatous neoplasms	633
• histology_recode_broad_groupings_germ cell neoplasms	634
• histology_recode_broad_groupings_gliomas	635
• histology_recode_broad_groupings_granular cell tumors & alveolar soft part sarcomas	636
• histology_recode_broad_groupings_lipomatous neoplasms	637
• histology_recode_broad_groupings_miscellaneous bone tumors	638
• histology_recode_broad_groupings_miscellaneous tumors	639
• histology_recode_broad_groupings_mucoepidermoid neoplasms	640
• histology_recode_broad_groupings_myomatous neoplasms	641
• histology_recode_broad_groupings_myxomatous neoplasms	642
• histology_recode_broad_groupings_nerve sheath tumors	643
• histology_recode_broad_groupings_neuroepitheliomatous neoplasms	644
• histology_recode_broad_groupings_nevi and melanomas	645
• histology_recode_broad_groupings_osseous and chondromatous neoplasms	646
• histology_recode_broad_groupings_paragangliomas and glomus tumors	647
• histology_recode_broad_groupings_soft tissue tumors and sarcomas, NOS	648
• histology_recode_broad_groupings_squamous cell neoplasms	649
• histology_recode_broad_groupings_synovial-like neoplasms	650
• histology_recode_broad_groupings_thymic epithelial neoplasms	651
• histology_recode_broad_groupings_transitional cell papillomas and carcinomas	652
• histology_recode_broad_groupings_trophoblastic neoplasms	653
• histology_recode_broad_groupings_unspecified neoplasms	654
• lat	655
• laterality_Bilateral involvement, lateral origin unknown; stated to be single primary	656
• laterality_Left: origin of primary	657
• laterality_Not a paired site	658
• laterality_Only one side involved, right or left origin unspecified	659
• laterality_Paired site, but no information concerning laterality; midline tumor	660
• laterality_Right: origin of primary	661
• lng	662
• marital_status_at_dx_Divorced	663
• marital_status_at_dx_Married (including common law)	664
• marital_status_at_dx_Separated	665
• marital_status_at_dx_Single (never married)	666
• marital_status_at_dx_Unknown	667
• marital_status_at_dx_Unmarried or domestic partner	668
• marital_status_at_dx_Widowed	669
• month_of_diagnosis_Apr	670
• month_of_diagnosis_Aug	671
• month_of_diagnosis_Dec	672
• month_of_diagnosis_Feb	673
• month_of_diagnosis_Jan	674
• month_of_diagnosis_Jul	675
• month_of_diagnosis_Jun	676
• month_of_diagnosis_Mar	677
• month_of_diagnosis_May	678
• month_of_diagnosis_Nov	679
• month_of_diagnosis_Oct	680
• month_of_diagnosis_Sep	681

- number_of primaries 684
- race_ethnicity_Amerian Indian, Aleutian, Alaskan Native or Eskimo 685
- race_ethnicity_Asian Indian 686
- race_ethnicity_Asian Indian or Pakistani 687
- race_ethnicity_Black 688
- race_ethnicity_Chamorroan 689
- race_ethnicity_Chinese 690
- race_ethnicity_Fiji Islander 691
- race_ethnicity_Filipino 692
- race_ethnicity_Guamanian 693
- race_ethnicity_Hawaiian 694
- race_ethnicity_Hmong 695
- race_ethnicity_Japanese 696
- race_ethnicity_Kampuchean 697
- race_ethnicity_Korean 698
- race_ethnicity_Laotian 699
- race_ethnicity_Melanesian 700
- race_ethnicity_Micronesian 701
- race_ethnicity_New Guinean 702
- race_ethnicity_Other 703
- race_ethnicity_Other Asian 704
- race_ethnicity_Pacific Islander 705
- race_ethnicity_Pakistani 706
- race_ethnicity_Polynesian 707
- race_ethnicity_Samoan 708
- race_ethnicity_Thai 709
- race_ethnicity_Tongan 710
- race_ethnicity_Unknown 711
- race_ethnicity_Vietnamese 712
- race_ethnicity_White 713
- seer_historic_stage_a_Distant 714
- seer_historic_stage_a_In situ 715
- seer_historic_stage_a_Localized 716
- seer_historic_stage_a_Regional 717
- seer_historic_stage_a_Unstaged 718
- sex_Female 719
- spanish_hispanic_origin_Cuban 720
- spanish_hispanic_origin_Dominican Republic 721
- spanish_hispanic_origin_Mexican 722
- spanish_hispanic_origin_Non-Spanish/Non-hispanic 723
- spanish_hispanic_origin_Other specified Spanish/Hispanic origin (excludes Dominican Repuclic) 724
- spanish_hispanic_origin_Puerto Rican 726
- spanish_hispanic_origin_South or Central American (except Brazil) 727
- spanish_hispanic_origin_Spanish surname only 728
- spanish_hispanic_origin_Spanish, NOS; Hispanic, NOS; Latino, NOS 729
- spanish_hispanic_origin_Uknown whether Spanish/Hispanic or not 730
- year_of_birth 731
- year_of_diagnosis 732
- month 733

and **newtarget** is the target variable, indicating whether or not the subject died in month given by the value of the **month** variable.

Breast Cancer Feature Selection

The feature set used as input into both the Random Forest and Neural Network models, after the transformation described in section Transformation of Censored Data for Machine Learning is given below and also available in full detail in the file `NewPatientBreastML.html`.

- `cs.tumor_size`
- `elevation`
- `grade_moderately` differentiated
- `grade_poorly` differentiated
- `grade_ndifferentiated`; anaplastic
- `grade_well` differentiated
- `histology_recode_broad_groupings_adenomas` and adenocarcinomas
- `histology_recode_broad_groupings_adnexal` and skin appendage neoplasms
- `histology_recode_broad_groupings_basal` cell neoplasms
- `histology_recode_broad_groupings_complex` epithelial neoplasms
- `histology_recode_broad_groupings_cystic`, mucinous and serous neoplasms
- `histology_recode_broad_groupings_ductal` and lobular neoplasms
- `histology_recode_broad_groupings_epithelial` neoplasms, NOS
- `histology_recode_broad_groupings_nerve` sheath tumors
- `histology_recode_broad_groupings_unspecified` neoplasms
- `lat`
- `laterality_Bilateral` involvement, lateral origin unknown; stated to be single primary
- `laterality_Paired` site, but no information concerning laterality; midline tumor
- `laterality_Right`: origin of primary
- `lng`
- `marital_stats_at_dx_Divorced`
- `marital_stats_at_dx_Married` (including common law)
- `marital_stats_at_dx_Separated`
- `marital_stats_at_dx_Single` (never married)
- `marital_stats_at_dx_Unknown`
- `marital_stats_at_dx_Unmarried` or domestic partner
- `marital_stats_at_dx_Widowed`
- `month_of_diagnosis_Apr`
- `month_of_diagnosis_Aug`
- `month_of_diagnosis_Dec`
- `month_of_diagnosis_Feb`
- `month_of_diagnosis_Jan`
- `month_of_diagnosis_Jul`
- `month_of_diagnosis_Jun`
- `month_of_diagnosis_Mar`
- `month_of_diagnosis_May`
- `month_of_diagnosis_Nov`
- `month_of_diagnosis_Oct`
- `month_of_diagnosis_Sep`
- `race_ethnicity_Amerian` Indian, Aleutian, Alaskan Native or Eskimo
- `race_ethnicity_Asian` Indian
- `race_ethnicity_Black`
- `race_ethnicity_Chinese`
- `race_ethnicity_Japanese`
- `race_ethnicity_Melanesian`

- race_ethnicity_Other
- race_ethnicity_Other Asian
- race_ethnicity_Pacific Islander
- race_ethnicity_Thai
- race_ethnicity_Unknown
- race_ethnicity_Vietnamese
- race_ethnicity_White
- seer_historic_stage_a_Distant
- seer_historic_stage_a_In sit
- seer_historic_stage_a_Localized
- seer_historic_stage_a_Unstaged
- sex_Female
- spanish_hispanic_origin_Cuban
- spanish_hispanic_origin_Mexican
- spanish_hispanic_origin_Non-Spanish/Non-hispanic
- spanish_hispanic_origin.Other specified Spanish/Hispanic origin (excldes Dominican Republic)
- spanish_hispanic_origin.Spanish surname only
- spanish_hispanic_origin.Spanish, NOS; Hispanic, NOS; Latino, NOS
- year_of_birth
- year_of_diagnosis
- month

and `newtarget` is the target variable, indicating whether or not the subject died in month given by the value of the `month` variable.

Pseudocode for the Data Transformation

```
def train(X, T, D)
    // X, T, D are the original dataset
    X' = []
    D' = []

    // the transformation
    for each index i in X:
        for t=1 to T[i]:
            new_D = (0 if t < T[i], else D[i])
            append new_D to D'
            new_X = (X[i], t)
            append new_X to X'

    return a decision tree trained on (X', D')
```

```
def pmf(h, X)
    // X is a single datapoint
    // returns an array A where A[i] = P(Y = i | X)
    A = []
    p_so_far = 1 // this is p(T >= t | X)
    for t = 1 to (the last month where h has any data):
        // h knows p(T = t | T >= t, X), we call this p_cur
        p_cur = h's prediction for (X, t)
        append (p_so_far * p_cur) to A
        p_so_far *= (1 - p_cur)
```

Breast Random Forest Model Hyperparameters

```
f = RandomForestClassifier(n_estimators=20,min_samples_split=3,
                           max_depth = 15,
                           max_features = .8,
                           n_jobs=5,verbose=2,random_state=33)
```

Colon Random Forest Model Hyperparameters

```
rf = RandomForestClassifier(n_estimators=25,min_samples_split=3,
                           max_depth = 10,
                           max_features = .5,
                           n_jobs=5,verbose=2,random_state=3)
```

Lung Random Forest Model Hyperparameters

```
rf = RandomForestClassifier(n_estimators=25,min_samples_split=3,
                           max_depth = 11,
                           max_features = .8,
                           n_jobs=5,verbose=2,random_state=3)
```

Breast Neural Network Model Architecture

The architecture of the Keras multilayer perceptron neural network model trained on the breast cancer data is given explicitly below:

```
modelbreast = Sequential()
modelbreast.add(Dense(114, input_shape=(66,) ,init='normal'))
modelbreast.add(Activation('relu'))
modelbreast.add(Dropout(0.05))
modelbreast.add(Dense(50, init='normal'))
modelbreast.add(Activation('relu'))
modelbreast.add(Dropout(0.05))

modelbreast.add(Dense(36, init='normal'))
modelbreast.add(Activation('relu'))
modelbreast.add(Dropout(0.05))

modelbreast.add(Dense(2, init='normal'))
modelbreast.add(Activation('softmax'))

rms = RMSprop(lr=0.001)

modelbreast.compile(loss='binary_crossentropy',
                    optimizer=rms, class_mode="binary")
```

and trained with a batch size of 1500 for 200 epochs.

Colon Cancer Neural Network Model Architecture

The architecture of the Keras multilayer perceptron neural network model trained on the colon cancer data is given explicitly below:

```
modelcolon = Sequential()
modelcolon.add(Dense(114, input_shape=(102,) ,init='normal'))
modelcolon.add(Activation('relu'))
modelcolon.add(Dropout(0.05))
modelcolon.add(Dense(50, init='normal'))
modelcolon.add(Activation('relu'))
modelcolon.add(Dropout(0.05))

modelcolon.add(Dense(35, init='normal'))
modelcolon.add(Activation('relu'))
modelcolon.add(Dropout(0.05))

modelcolon.add(Dense(2, init='normal'))
modelcolon.add(Activation('softmax'))

rms = RMSprop(lr=0.001)

modelcolon.compile(loss='binary_crossentropy',
                    optimizer=rms, class_mode="binary")
```

and trained with a batch size of 1500 for 200 epochs.

Lung Cancer Neural Network Model Architecture

The architecture of the Keras multilayer perceptron neural network model trained on the lung cancer data is given explicitly below:

```
modellung = Sequential()
modellung.add(Dense(114, input_shape=(114,) ,init='normal'))
modellung.add(Activation('relu'))
modellung.add(Dropout(0.1))
modellung.add(Dense(80, init='normal'))
modellung.add(Activation('relu'))
modellung.add(Dropout(0.1))
modellung.add(Dense(40, init='normal'))
modellung.add(Activation('relu'))
modellung.add(Dropout(0.1))

modellung.add(Dense(2, init='normal'))
modellung.add(Activation('softmax'))

rms = RMSprop(lr=0.001)
```

```
modellung.compile(loss='binary_crossentropy',
                  optimizer=rms, class_mode="binary")
```

and trained with a batch size of 2000 for 50 epochs.

Acknowledgments

Cras egestas velit mauris, eu mollis turpis pellentesque sit amet. Interdum et malesuada fames ac ante ipsum primis in faucibus. Nam id pretium nisi. Sed ac quam id nisi malesuada congue. Sed interdum aliquet augue, at pellentesque quam rhoncus vitae.

References

1. Pandora Media, Inc . Pandora Internet Radio - Listen to Free Music You'll Love; 2016 (accessed 12 Feb 2016). <http://www.pandora.com/>.
2. Goodreads Inc. Share Book Recommendations With Your Friends, Join Book Clubs, Answer Trivia; 2016 (accessed 12 Feb 2016). <http://www.goodreads.com/>.
3. Google Maps. 4901 Lang Ave NE to Albuquerque Sunport, Albuquerque, NM - Google Maps; 2016 (accessed 17 Feb 2016). <https://goo.gl/1D7Jwf>.
4. Sebastian Raschka. Python Machine Learning Essentials. Packt Publishing; 2015.
5. Cam Davidson-Pilon. Quickstart – lifelines 0.8.0.1 documentation; 2016 (accessed 14 Jan 2016). <http://lifelines.readthedocs.org/en/latest/Quickstart.html>.
6. Kumbasar U, Raubenheimer H, Sahaf MA, Asadi N, Cufari ME, Proli C, et al. Selection for adjuvant chemotherapy in completely resected stage I non-small cell lung cancer: External validation of a Chinese prognostic risk model. Journal of Thoracic Disease. 2016;8(1):140–144. Cited By 0. Available from: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84957096795&partnerID=40&md5=be331b5dc81ff3f73fd9e30aed8c00e0>.
7. Shin, Hyunjung and Nam, Yonghyun; ISCB Asia. A coupling approach of a predictor and a descriptor for breast cancer prognosis [Article; Proceedings Paper]. BMC MEDICAL GENOMICS. 2014 MAY 8;7(1). 3rd Annual Translational Bioinformatics Conference (TBC) / ISCB-Asia, Seoul, SOUTH KOREA, OCT 02-04, 2013.
8. Zolbanin, Hamed Majidi and Delen, Dursun and Zadeh, Amir Hassan. Predicting overall survivability in comorbidity of cancers: A data mining approach [Article]. DECISION SUPPORT SYSTEMS. 2015 JUN;74:150–161.
9. Gordon L, Olshen RA. Tree-structured survival analysis. Cancer Treatment Reports. 1985;69(10):1065–1068. Cited By 97. Available from: <http://www.scopus.com/inward/record.url?eid=2-s2.0-0021875130&partnerID=40&md5=9e112ed840960f801b6260b23bf6811d>.

10. Bou-Hamad I, Larocque D, Ben-Ameur H. A review of survival trees. *Statistics Surveys*. 2011;5:44–71. Cited By 15. Available from: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84857308440&partnerID=40&md5=f8af82017ade68e335fd258c6857bf49>.
11. Ishwaran H, Kogalur UB. Consistency of random survival forests. *Statistics and Probability Letters*. 2010;80(13-14):1056–1064. Cited By 26. Available from: <http://www.scopus.com/inward/record.url?eid=2-s2.0-77953020220&partnerID=40&md5=1e4478c51150f0159fdc6c1cb631968b>.
12. Ben Kuhn. Decision trees for survival analysis; 2016 (accessed 14 Jan 2016). <http://www.benkuhn.net/survival-trees>.
13. Allen Downey. *Think Stats*. O'Reilly Media; 2014.
14. IOBS. Supplemental Material — PAPERDATA; 2016 (accessed 18 Jan 2016). <https://github.com/doolingdavid/PAPERDATA.git>.
15. scikit-learn developers. 3.2.4.3.1. `sklearn.ensemble.RandomForestClassifier`; 2014 (accessed 25 Jan 2016). <http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>.
16. Kaggle Inc . Random Forests — Kaggle; 2015 (accessed 25 Jan 2016). <https://www.kaggle.com/wiki/RandomForests>.
17. Michael Nielsen. *Neural Networks and Deep Learning*; Jan 2016 (accessed 25 Jan 2016). <http://neuralnetworksanddeeplearning.com/>.
18. T Unterthiner, A Mayr, G Klambauer, and S Hochreiter. Toxicity Prediction using deep learning; 4 Mar 2015 (accessed 25 Jan 2016). <http://arxiv.org/abs/1503.01445>.
19. F Chollet. *Keras Documentation*; 2015 (accessed 25 Jan 2016). <http://keras.io/>.
20. Armin Roncaher. *Welcome — Flask (A Python Microframework)*; 2014 (accessed 29 Jan 2016). <http://flask.pocoo.org>.
21. Kai Porter, KOB Eyewitness News 4. Study links higher elevation with lower lung cancer risk; 26 Jan 2016 (accessed 27 Jan 2016). <http://www.kob.com/article/stories/s4029233.shtml#.VqlUafkrJhF>.
22. Statistical Software Information, University of Massachusetts Amherst. *Software - Statistical Consulting Center - UMass Amherst*; 2004 (accessed 29 Jan 2016). <https://www.umass.edu/statdata/statdata/stat-survival.html>.
23. National Cancer Institute, Division of Cancer Control and Population Sciences. *SEER-Medicare Linked Database*; 2015 (accessed 10 Feb 2016). <http://healthcaredelivery.cancer.gov/seermedicare/>.
24. Michael Bowles. *Machine Learning in Python: Essential Techniques for Predictive Analysis*. Wiley; 2015.
25. United States Census Bureau. *2010 FIPS Code Files for Counties - Geography - U.S. Census Bureau*; 2016 (accessed 18 Jan 2016). <https://www.census.gov/geo/reference/codes/cou.html>.

26. Google Developers. The Google Maps Geocoding API — Google Maps Geocoding API — Google Developers; 2016 (accessed 18 Jan 2016). <https://developers.google.com/maps/documentation/geocoding/intro>.
27. Google Developers. The Google Maps Elevation API — Google Maps Elevation API — Google Developers; 2016 (accessed 18 Jan 2016). <https://developers.google.com/maps/documentation/elevation/intro?hl=en>.