Personalized Prognostic Models for Oncology: A Machine Learning Approach

David Dooling[1], Angela Kim[1], Barbara McAneny[1], Jennifer Webster[1]

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 afficianados 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 propery 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, Epidemiolgy, 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. [8] use semi-supervised learning techniques to predict 5 year survival, essentially imputing values for SEER records where the survival infomation is censored at a value less than 5 years. Zolbanin et al. [9] 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. [10]. A summary of more recent developments concerning survival trees is provided by Bou-Hamad et al. [11]. These methods focus on altering the splitting critieria 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 [12].

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 liklihood 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 Infomation 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 multple 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 [19].

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

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

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

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

Knowing $P(Y = t_j | \mathbf{X}_i)$ for all t_j gives the full probablity distribution of dying at time Y [19]. The survival function is then readily derived from this distribution as

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

where $CDF(\mathbf{X}_i, t_k) = \sum_{j=1}^{k} P(Y = t_j | \mathbf{X}_i)$ is the cumulative density function correponding to the probability mass function in equation [20].

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_Dead
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 explodes 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 $(4165251, 103)$. Similary, the original, untransformed lung cancer DataFrame has shape $(177089, 115)$, and the total transformed lung cancer DataFrame has shape $(3079931, 115)$. The biggest explosion in dataset size occured 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 an extra 48 rows to the transformed dataset. The original, untransformed breast cancer DataFrame has shape $(329949, 67)$, and the total transformed breast cancer DataFrame has shape $(15085711, 67)$. Traning machine learning algorithms on such large datasets, even after splitting into training and testing sets described below, require 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 testing set. All records corresponding to a given patient were assigned exclusively to either the training or test set. 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 $(91163, 115)$. Finallly, 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.

Finally, we emphasize the crucial point that the features `survival_months` and `vital_status_recode_Dead` are dropped from both the training and and testing data, and are replaced with the features `months` and `newtarget`, as illustrated in Tables (1, **??**, 2, **??**). The information of which subjects represent censored data (`vital_status_recode_Dead` $== 0$) and which died is retained and recoverable through the `newindex` variable and is needed for proper evaluation of the performance metrics; when evaluating AUC curves for the 6, 12, and 60 month binary classifiers, we need to limit the test data to those subjects whose vital status is definitively known at those time points. We introduce the two machine learning algorithms used in this study below, chosen because of their high performance in machine learning competitions and their complementary methods, so that their mutual agreement shown below on the test datasets can be taken as indication that they are actually learning useful information.

Random Forests are made up of an ensemble of independent Decision trees that are purposefully exposed to only subsets of the data. The general philosophy is presented in the popular science book "The Wisdom of Crowds" [21]. The idea is that a large number of independent non-expert opinions converge on the correct answer when averaged. The success of this philosophy of prediction was startingly shown by the success of the political and world event predictions made by the prediction market site Intrade, before its forced closure by the Commodity Futures Trading Commission [22]. The other class of methods used in this study are called neural networks, and are modelled on how the human brain learns high level concepts from lower level ones. As opposed to the crowd-based wisdom of a random forest, a neural network is analgous to a seasoned expert. A Neural network learns from repeated exposure to the training data and improves its predictions with each pass over the data. The general philosophy is simlar to that represented by the well-known maxim that it takes 10,000 hours to become an expert in any given field [23].

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 [18]. For each subject i, all input data minus `months` and `newtarget` is represented by $\mathbf{X}_i$. After the classfier 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 intrepeted 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 successfuly 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 implemention of the Random Forest machine learning classifier [24]. Random Forests are frequent winners of the Kaggle machine learning competitions [25]. 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 [26]. 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 [27].

   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) [28]. 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 corresonding to 6, 12, and 60
months. If the survival function eveluted 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 eveluted 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.

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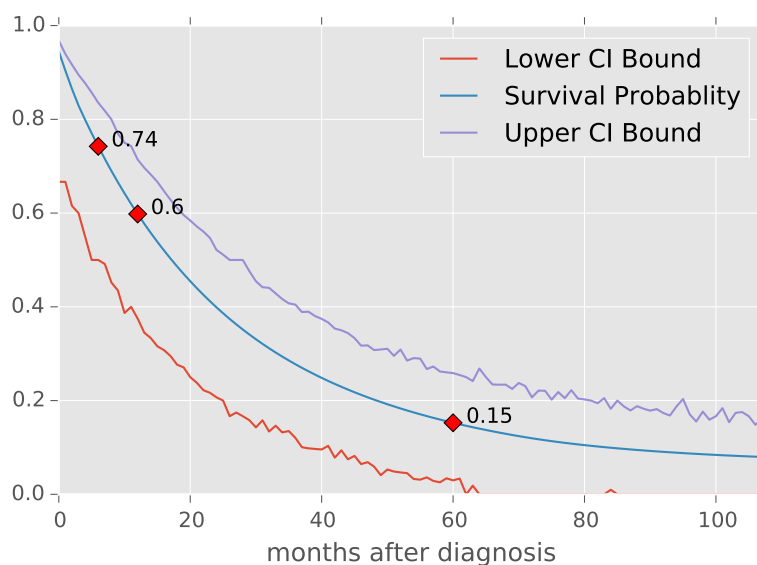
Figure 1. Example of the construction of the binary classifiers for 6, 12, and 60 months survival. A patient's hazard curve $\lambda(\mathbf{X}_i, t_j)$ is predicted by the model for times out to 107 months. The survival curve is then readily computed as in Equation (3). For this example, the 6-month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

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 somewhat anologous 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 ₃₀₉
survival. ₃₁₀

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 ₃₁₁
models is by comparing their predictions to each other for the same set of input data. ₃₁₂
Table 4 shows the strong agreement between the random forest and neural network ₃₁₃
classifiers for each cancer type. Python code showing how the values in Table 4 are ₃₁₄
computed is available in the files `NewPatientBreastCF.html`, ₃₁₅
`NewPatientColonCF.html`, and `NewPatientLung.html` in the GitHub repository ₃₁₆
containing supplemental matierial for this study [18]. Table 4 is computed as follows. ₃₁₇
For each cancer type (breast,colon, and lung), do the following: ₃₁₈

- use the corresponding Random Forest and Neural Network models to compute the ₃₁₉
  survival curves for all of the test subjects ₃₂₀
- extract the values of the survival curve evaluted for 6, 12, and 60 months for both ₃₂₁
  models ₃₂₂
- if both models predict less than .5 or both models predict greater than or equal to ₃₂₃
  .5, that counts as agreement ₃₂₄
- otherwise, the models disagree ₃₂₅

The high level of agreement between two models lends confidence to the notion that ₃₂₆
they have both learned from the training data and are generalizing well. ₃₂₇
Figures (3, 2, 4) show box plots of the value of the random forest prediction subtracted ₃₂₈
from the neural network prediction. We emphasize that when evaluating the model ₃₂₉
agreement, we put no restrictions on the distinct subjects in the respective test datasets; ₃₃₀
we are confronting the models against each other, not some known ground truth as in ₃₃₁
the AUC performance metric calculations. The number of distinct subjects in all three ₃₃₂
of the colon cancer survival binary classifiers (6, 12, and 60 month survival) was 5654; ₃₃₃
for lung cancer the number of subjects entering into the calculation of Table (4) was ₃₃₄
5313; and for breast cancer it was 3300. Exploring the outlier cases, for example where ₃₃₅
the difference between the neural network and random forest predictions for 5 year ₃₃₆
survival in breast cancer exceed .4, can reveal patterns indicating where the models ₃₃₇
need careful intepretation. The distribution of year of birth for patients where the ₃₃₈
neural network model prediction for 5 year survival exceeded the random forest ₃₃₉
prediction by more than .4 has a mode around 1930, a good 20 years earlier than the ₃₄₀
distrubtion for the remaining patients. The outliers in Figs. (3, 2, 4) corresponding to ₃₄₁
patients whose distributions for certain features are also outliers compared to the rest of ₃₄₂
the population used in training. ₃₄₃
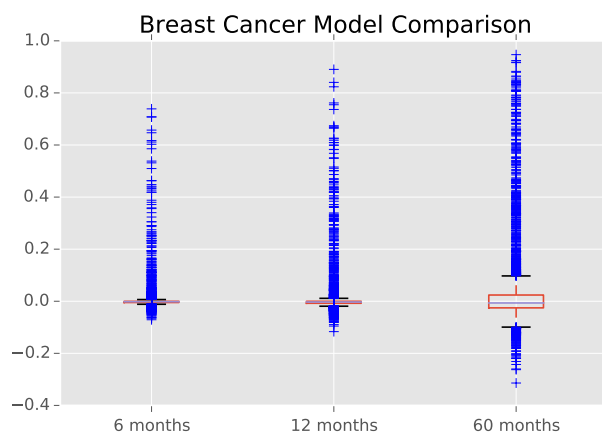
Figure 2. Box plots showing the distributions of the signed difference between the MLP model's prediction for the probability of surviving 6 months and the Random Forest model's prediction of the same quantity for breast cancer. The plot shows the same quantity for the 12 and 60 months classifiers. It is apparent from the figures that the outliers are due to the neural network models predicting higher survival probablitlies than the random forest for some few cases. These differences were evaluated for the 3300 test patients in the breast cancer data.
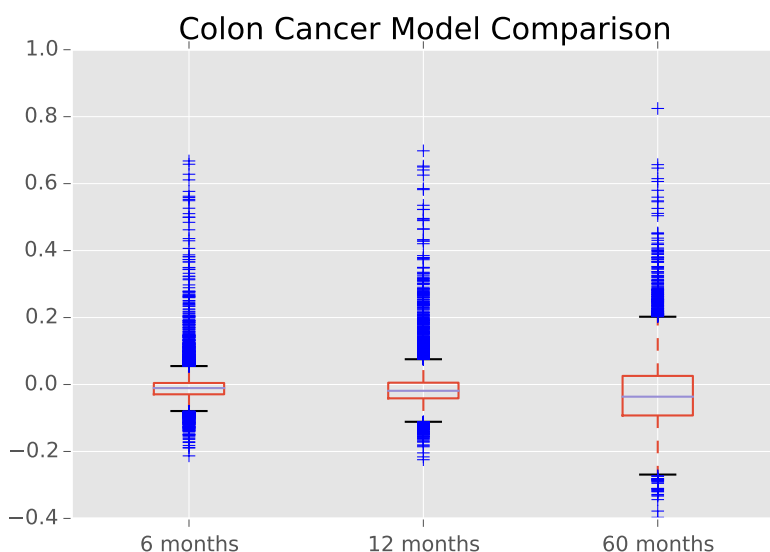
Figure 3. Box plots showing the distributions of the signed difference between the MLP model's prediction for the probability of surviving 6 months and the Random Forest model's prediction of the same quantity for colon cancer. The plot shows the same quantity for the 12 and 60 months classifiers. It is apparent from the figures that the outliers are due to the neural network models predicting higher survival probablitlies than the random forest for some few cases. These differences were evaluated for the 5654 test patients in the colon cancer data.

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
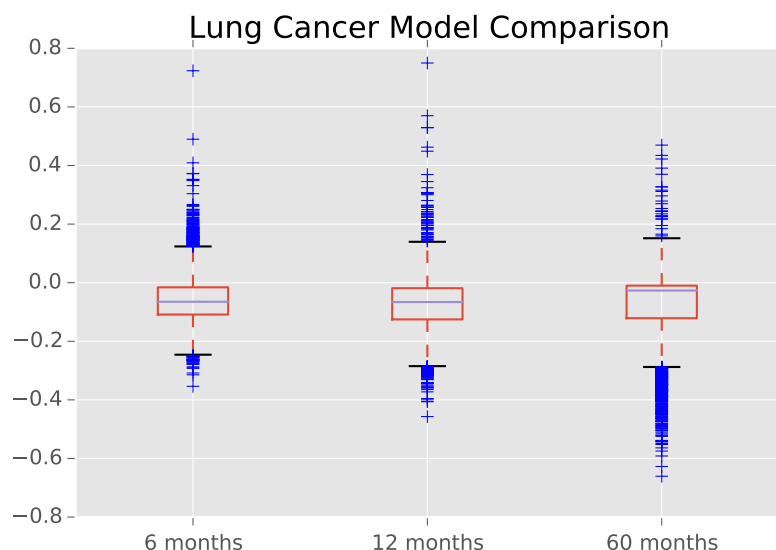
Figure 4. Box plots showing the distributions of the signed difference between the MLP model's prediction for the probability of surviving 6 months and the Random Forest model's prediction of the same quantity for lung cancer. The plot shows the same quantity for the 12 and 60 months classifiers. These differences were evaluated for the 5313 test patients in the lung cancer data. The Interquartile Ranges for lung cancer are visibly larger than those for breast cancer and colon cancer shown in fig 2 and fig 3.

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 githib respository containing supplemental material for this study [18].

Using the popular Flask microframework for web applications [29], 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.

1. breast cancer
   (a) random forest:
       https://github.com/doolingdavid/breast-cancer-rf-errors.git
   (b) neural network:
       https://github.com/doolingdavid/breast-cancer-nn-errors.git
2. lung cancer

(a) random forest:
   `https://github.com/doolingdavid/lung-cancer-rf-errors.git`

(b) neural network:
   `https://github.com/doolingdavid/lung-cancer-nn-errors.git`

3. colon cancer

   (a) random forest:
      `https://github.com/doolingdavid/colon-cancer-rf-errors.git`

   (b) neural network:
      `https://github.com/doolingdavid/colon-cancer-nn-errors.git`

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

```
>python hello.py
```

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

`http://localhost:5000`.

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

For example, using the Colon Cancer neural network app, and inputing the values listed in Table (5) results in the survival curve depicted in Figure (5); the predicted probablities of living at least 6, 12, and 60 months are .89, .83, and .50, respectively.

Table 5. 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
Martial 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 5 so that the address field is changed from Boston, Massachusetts to Denver, Colorado but keeping all other variables are unchanged results 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 probablities are noticeably higher and reflect the documented effects of both longitude and elevation on cancer treatment and prognosis in the United States [30].

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

Colon Cancer Survival Curve Prediction

Prediction:

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

Predicted Survival Curve from Model
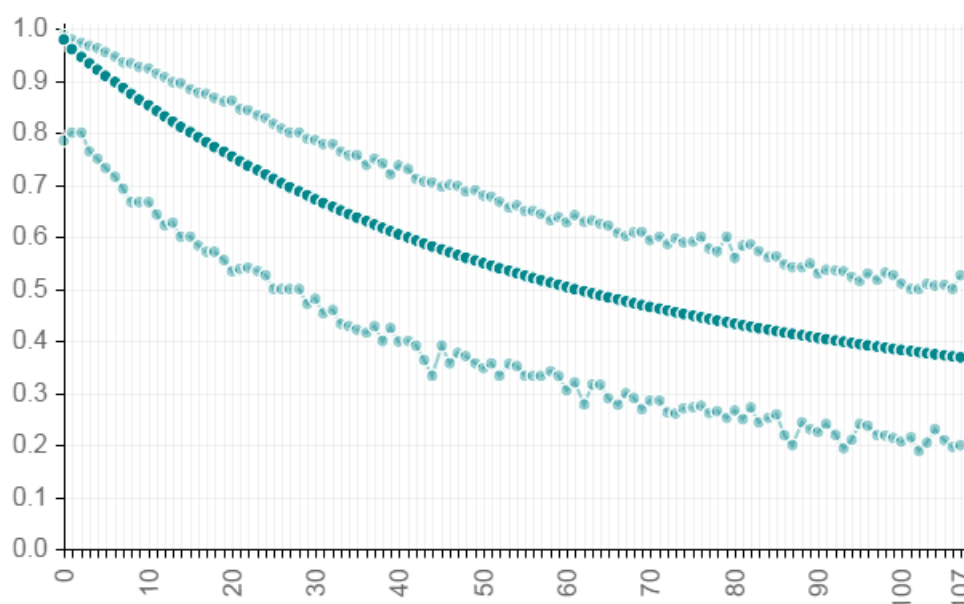
Figure 5. Colon Cancer Survival Curve predicted from the data in Table (5) using the neural network web app
`https://github.com/doolingdavid/colon-cancer-nn-errors.git`.

cancer. Changing the data in Table 6 by toggling between the male/female, and married/single four possible permutations results in the following prediction probabilites 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 `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

Table 6. 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
Martial 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

It it 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 help further the cause of PPPM medicine by developing models of personalized suvival curve prognosis. To help further refine the methodology, we would like to apply it to different survival datasets [31], 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 sophisitcated 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 [32]. 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 pioneeered by Judea Pearl of the University of California, Los Angeles [1] and seems particulary relevant and applicable to censored survival data. Combining the methodlogy 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 the explanatory variable is encoded using one binary feature for each of the variable's possible values [14].

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

---

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

approximately 3,000 counties. Clearly, transforming the `STATE-COUNTY RECODE` data 475
representation into distinct (state_county) columns will explode the dataset to become 476
wider than is optimal for machine learning. Adding extra columns to your dataset, 477
making it wider, requires more data rows (making it taller) in order for machine 478
learning algorithms to effectively learn [14]. Because one-hot coding 479
`STATE-COUNTY RECODE` would cause such drastic shape changes in our data, we wish 480
to avoid doing so. Fortunately, this variable, though given as a categorical variable, is 481
actually a recode for three ordinal variables. There is an ordering among the 482
(state_county) columns, namely longitude, latitude, and elevation. We can transform the 483
data in `STATE-COUNTY RECODE` into three new numerical columns: `lat`, `lng`, and 484
`elevation`. 485

For example, Table (7) shows how five entries of `STATE-COUNTY RECODE` 486
corresponding to counties within New Mexico can be represented by the `elevation`, 487
`lat`, and `lng` features. 488

Table 7. 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 489
`STATE-COUNTY RECODE` variable to the corresponding three values `elevation`, 490
`lat`, and `lng`. We use the publically available dafafile from the United States Census 491
Bureau [15] to map the state FIPS and county FIPS codes to query strings like those in 492
the `address` field in Table (7). It is then possible to programmatically query the 493
Google Maps Geocoding API for the latitude and longitude [16], and the Google Maps 494
Elevation API for the corresponding elevation [17]. An added benefit of this shift from 495
the single categorical variable `STATE-COUNTY RECODE` to the three continuous 496
numerical variables `lat`, `lng`, and `elevation` is that input into the web 497
applications described later are not restricted to the states and counties coverered in the 498
SEER registries; in fact, the input to the models can be any address you would enter 499
into Google Maps and calls to the Google Maps Geocoding API and the Google Maps 500
Elevation API provide the conversion from the address string to the input variables 501
`lat`, `lng`, and `elevation`. The full lookup table analogous to Table (7) is available 502
from a GitHub repository containing supplemental information for this study [18]. 503

Data Subsets 504

The four COLRECT.txt files were imported into a pandas DataFrame object. This data 505
was then filtered according to the conditions in Table (8). The RESPIR.txt and 506
BREAST.txt files were imported into separate dataframes in similar fashion and filtered 507
according to the conditions in Table (9) and Table (10), respectively. The SEER 508
variable `CS TUMOR SIZE` records the tumor size in millimeters if known. But if not 509
known, `CS TUMOR SIZE` is given as '999', to indicate that the tumor size is "Unknown; 510
size not stated; not stated in pateint record." In this study, we discard those records, as 511
indicated in Tables (10, 8, 9). 512

Table 8. Filters applied to the Colon Cancer data.

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

Table 9. Filters applied to the Lung Cancer data.

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

Table 10. Filters applied to the Breast Cancer data.

Column	Filter
SEQUENCE NUMBER-CENTRAL	$\neq$ "Unspecified"
AGE AT DIAGNOSIS	$\neq$ "Unknown age"
BIRTHDATE-YEAR	$\neq$ "Unknown year of birth"
YEAR OF DIAGNOSIS	≥ 2004
SURVIVAL MONTHS FLAG	= "1"
CS TUMOR SIZE EXT/EVAL	$\neq$ " "
CS TUMOR SIZE	$\neq 999$
SEER RECORD NUMBER	$= 1$
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` , 517
- `SPANISH/HISPANIC ORIGIN` , 518
- `GRADE` , 519
- `PRIMARY SITE` , 520
- `LATERALITY` , 521
- `SEER HISTORIC STAGE A` , 522
- `HISTOLOGY RECODE--BROAD GROUPINGS` , 523
- `MONTH OF DIAGNOSIS` , 524
- `VITAL STATUS RECODE` , 525

and the `STATE-COUNTY RECODE` variable was dropped and replaced with the 526
`elevation` , `lat` , and `lng` variables for all three datasets as illustrated in Table (7). 527

Colon Cancer Feature Selection 528

The feature set used as input into both the Random Forest and Neural Network models, 529
after the transformation described in section Transformation of Censored Data for 530
Machine Learning is given below and also available in full detail in the file 531
`NewPatientColonML.html` . 532

- cs_tumor_size 533
- elevation 534
- grade_cell type not determined 535
- grade_moderately differentiated 536
- grade_poorly differentiated 537
- grade_undifferentiated; anaplastic 538
- grade_well differentiated 539
- histology_recode_broad_groupings_acinar cell neoplasms 540
- histology_recode_broad_groupings_adenomas and adenocarcinomas 541
- histology_recode_broad_groupings_blood vessel tumors 542
- histology_recode_broad_groupings_complex epithelial neoplasms 543
- histology_recode_broad_groupings_complex mixed and stromal neoplasms 544
- histology_recode_broad_groupings_cystic, mucinous and serous neoplasms 545
- histology_recode_broad_groupings_ductal and lobular neoplasms 546
- histology_recode_broad_groupings_epithelial neoplasms, NOS 547
- histology_recode_broad_groupings_fibromatuos neoplasms 548
- histology_recode_broad_groupings_germ cell neoplasms 549
- histology_recode_broad_groupings_lipomatous neplasms 550
- histology_recode_broad_groupings_miscellaneous bone tumors 551
- histology_recode_broad_groupings_myomatous neoplasms 552
- histology_recode_broad_groupings_neuroepitheliomatous neoplasms 553
- histology_recode_broad_groupings_nevi and melanomas 554
- histology_recode_broad_groupings_paragangliomas and glumus tumors 555
- histology_recode_broad_groupings_soft tissue tumors and sarcomas, NOS 556
- histology_recode_broad_groupings_squamous cell neoplasms 557
- histology_recode_broad_groupings_synovial-like neoplasms 558
- histology_recode_broad_groupings_transistional cell papillomas and carcinomas 559
- histology_recode_broad_groupings_unspecified neoplasms 560
- lat 561
- laterality_Left: origin of primary 562
- laterality_Not a paired site 563
- laterality_Only one side involved, right or left origin unspecified 564

- laterality_Paired site, but no information concerning laterality; midline tumor $\quad$ 565
- laterality_Right: origin of primary $\quad$ 566
- lng $\quad$ 567
- marital_status_at_dx_Divorced $\quad$ 568
- marital_status_at_dx_Married (including common law) $\quad$ 569
- marital_status_at_dx_Separated $\quad$ 570
- marital_status_at_dx_Single (never married) $\quad$ 571
- marital_status_at_dx_Unknown $\quad$ 572
- marital_status_at_dx_Unmarried or domestic partner $\quad$ 573
- marital_status_at_dx_Widowed $\quad$ 574
- month_of_diagnosis_Apr $\quad$ 575
- month_of_diagnosis_Aug $\quad$ 576
- month_of_diagnosis_Dec $\quad$ 577
- month_of_diagnosis_Feb $\quad$ 578
- month_of_diagnosis_Jan $\quad$ 579
- month_of_diagnosis_Jul $\quad$ 580
- month_of_diagnosis_Jun $\quad$ 581
- month_of_diagnosis_Mar $\quad$ 582
- month_of_diagnosis_May $\quad$ 583
- month_of_diagnosis_Nov $\quad$ 584
- month_of_diagnosis_Oct $\quad$ 585
- month_of_diagnosis_Sep $\quad$ 586
- number_of_primaries $\quad$ 587
- race_ethnicity_Amerian Indian, Aleutian, Alaskan Native or Eskimo $\quad$ 588
- race_ethnicity_Asian Indian $\quad$ 589
- race_ethnicity_Asian Indian or Pakistani $\quad$ 590
- race_ethnicity_Black $\quad$ 591
- race_ethnicity_Chinese $\quad$ 592
- race_ethnicity_Fiji Islander $\quad$ 593
- race_ethnicity_Filipino $\quad$ 594
- race_ethnicity_Guamanian $\quad$ 595
- race_ethnicity_Hawaiian $\quad$ 596
- race_ethnicity_Hmong $\quad$ 597
- race_ethnicity_Japanese $\quad$ 598
- race_ethnicity_Kampuchean $\quad$ 599
- race_ethnicity_Korean $\quad$ 600
- race_ethnicity_Laotian $\quad$ 601
- race_ethnicity_Melanesian $\quad$ 602
- race_ethnicity_Micronesian $\quad$ 603
- race_ethnicity_New Guinean $\quad$ 604
- race_ethnicity_Other $\quad$ 605
- race_ethnicity_Other Asian $\quad$ 606
- race_ethnicity_Pacific Islander $\quad$ 607
- race_ethnicity_Pakistani $\quad$ 608
- race_ethnicity_Polynesian $\quad$ 609
- race_ethnicity_Samoan $\quad$ 610
- race_ethnicity_Thai $\quad$ 611
- race_ethnicity_Tongan $\quad$ 612
- race_ethnicity_Unknown $\quad$ 613
- race_ethnicity_Vietnamese $\quad$ 614
- race_ethnicity_White $\quad$ 615
- seer_historic_stage_a_Distant $\quad$ 616

- seer_historic_stage_a_In situ   617
- seer_historic_stage_a_Localized   618
- seer_historic_stage_a_Regional   619
- seer_historic_stage_a_Unstaged   620
- sex_Female   621
- spanish_hispanic_origin_Cuban   622
- spanish_hispanic_origin_Dominican Republic   623
- spanish_hispanic_origin_Mexican   624
- spanish_hispanic_origin_Non-Spanish/Non-hispanic   625
- spanish_hispanic_origin_Other specified Spanish/Hispanic origin (excludes   626
  Dominican Repuclic)   627
- spanish_hispanic_origin_Puerto Rican   628
- spanish_hispanic_origin_South or Central American (except Brazil)   629
- spanish_hispanic_origin_Spanish surname only   630
- spanish_hispanic_origin_Spanish, NOS; Hispanic, NOS; Latino, NOS   631
- spanish_hispanic_origin_Uknown whether Spanish/Hispanic or not   632
- year_of_birth   633
- year_of_diagnosis   634
- month   635

and `newtarget` is the target variable, indicating whether or not the subject died in   636
month given by the value of the `month` variable.   637

Lung Cancer Feature Selection   638

The feature set used as input into both the Random Forest and Neural Network models,   639
after the transformation described in section Transformation of Censored Data for   640
Machine Learning is given below and also available in full detail in the file   641
`NewPatientLungML.html`.   642

- cs_tumor_size   643
- elevation   644
- grade_cell type not determined   645
- grade_moderately differentiated   646
- grade_poorly differentiated   647
- grade_undifferentiated; anaplastic   648
- grade_well differentiated   649
- histology_recode_broad_groupings_acinar cell neoplasms   650
- histology_recode_broad_groupings_adenomas and adenocarcinomas   651
- histology_recode_broad_groupings_blood vessel tumors   652
- histology_recode_broad_groupings_complex epithelial neoplasms   653
- histology_recode_broad_groupings_complex mixed and stromal neoplasms   654
- histology_recode_broad_groupings_cystic, mucinous and serous neoplasms   655
- histology_recode_broad_groupings_ductal and lobular neoplasms   656
- histology_recode_broad_groupings_epithelial neoplasms, NOS   657
- histology_recode_broad_groupings_fibroepithelial neoplasms   658
- histology_recode_broad_groupings_fibromatuos neoplasms   659
- histology_recode_broad_groupings_germ cell neoplasms   660
- histology_recode_broad_groupings_gliomas   661
- histology_recode_broad_groupings_granular cell tumors & alveolar soft part   662
  sarcomas   663
- histology_recode_broad_groupings_lipomatous neplasms   664
- histology_recode_broad_groupings_miscellaneous bone tumors   665

- histology_recode_broad_groupings_miscellaneous tumors
- histology_recode_broad_groupings_mucoepidermoid neoplasms
- histology_recode_broad_groupings_myomatous neoplasms
- histology_recode_broad_groupings_myxomatous neoplasms
- histology_recode_broad_groupings_nerve sheath tumors
- histology_recode_broad_groupings_neuroepitheliomatous neoplasms
- histology_recode_broad_groupings_nevi and melanomas
- histology_recode_broad_groupings_osseous and chondromatous neoplasms
- histology_recode_broad_groupings_paragangliomas and glumus tumors
- histology_recode_broad_groupings_soft tissue tumors and sarcomas, NOS
- histology_recode_broad_groupings_squamous cell neoplasms
- histology_recode_broad_groupings_synovial-like neoplasms
- histology_recode_broad_groupings_thymic epithelial neoplasms
- histology_recode_broad_groupings_transistional cell papillomas and carcinomas
- histology_recode_broad_groupings_trophoblastic neoplasms
- histology_recode_broad_groupings_unspecified neoplasms
- lat
- laterality_Bilateral involvement, lateral origin unknown; stated to be single primary
- laterality_Left: origin of primary
- laterality_Not a paired site
- laterality_Only one side involved, right or left origin unspecified
- laterality_Paired site, but no information concerning laterality; midline tumor
- laterality_Right: origin of primary
- lng
- marital_status_at_dx_Divorced
- marital_status_at_dx_Married (including common law)
- marital_status_at_dx_Separated
- marital_status_at_dx_Single (never married)
- marital_status_at_dx_Unknown
- marital_status_at_dx_Unmarried or domestic partner
- marital_status_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
- number_of_primaries
- race_ethnicity_Amerian Indian, Aleutian, Alaskan Native or Eskimo
- race_ethnicity_Asian Indian
- race_ethnicity_Asian Indian or Pakistani
- race_ethnicity_Black
- race_ethnicity_Chamorran
- race_ethnicity_Chinese
- race_ethnicity_Fiji Islander

666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717

- race_ethnicity_Filipino 718
- race_ethnicity_Guamanian 719
- race_ethnicity_Hawaiian 720
- race_ethnicity_Hmong 721
- race_ethnicity_Japanese 722
- race_ethnicity_Kampuchean 723
- race_ethnicity_Korean 724
- race_ethnicity_Laotian 725
- race_ethnicity_Melanesian 726
- race_ethnicity_Micronesian 727
- race_ethnicity_New Guinean 728
- race_ethnicity_Other 729
- race_ethnicity_Other Asian 730
- race_ethnicity_Pacific Islander 731
- race_ethnicity_Pakistani 732
- race_ethnicity_Polynesian 733
- race_ethnicity_Samoan 734
- race_ethnicity_Thai 735
- race_ethnicity_Tongan 736
- race_ethnicity_Unknown 737
- race_ethnicity_Vietnamese 738
- race_ethnicity_White 739
- seer_historic_stage_a_Distant 740
- seer_historic_stage_a_In situ 741
- seer_historic_stage_a_Localized 742
- seer_historic_stage_a_Regional 743
- seer_historic_stage_a_Unstaged 744
- sex_Female 745
- spanish_hispanic_origin_Cuban 746
- spanish_hispanic_origin_Dominican Republic 747
- spanish_hispanic_origin_Mexican 748
- spanish_hispanic_origin_Non-Spanish/Non-hispanic 749
- spanish_hispanic_origin_Other specified Spanish/Hispanic origin (excludes 750
  Dominican Repuclic) 751
- spanish_hispanic_origin_Puerto Rican 752
- spanish_hispanic_origin_South or Central American (except Brazil) 753
- spanish_hispanic_origin_Spanish surname only 754
- spanish_hispanic_origin_Spanish, NOS; Hispanic, NOS; Latino, NOS 755
- spanish_hispanic_origin_Uknown whether Spanish/Hispanic or not 756
- year_of_birth 757
- year_of_diagnosis 758
- month 759

and `newtarget` is the target variable, indicating whether or not the subject died in 760
month given by the value of the `month` variable. 761

Breast Cancer Feature Selection 762

The feature set used as input into both the Random Forest and Neural Network models, 763
after the transformation described in section Transformation of Censored Data for 764
Machine Learning is given below and also available in full detail in the file 765
`NewPatientBreastML.html`. 766

- 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 (inclding 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, Aletian, 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 {#819}
- seer_historic_stage_a_Distant {#820}
- seer_historic_stage_a_In sit {#821}
- seer_historic_stage_a_Localized {#822}
- seer_historic_stage_a_Unstaged {#823}
- sex_Female {#824}
- spanish_hispanic_origin_Cuban {#825}
- spanish_hispanic_origin_Mexican {#826}
- spanish_hispanic_origin_Non-Spanish/Non-hispanic {#827}
- spanish_hispanic_origin_Other specified Spanish/Hispanic origin (excldes Dominican Republic) {#828 #829}
- spanish_hispanic_origin_Spanish surname only {#830}
- spanish_hispanic_origin_Spanish, NOS; Hispanic, NOS; Latino, NOS {#831}
- year_of_birth {#832}
- year_of_diagnosis {#833}
- month {#834}

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,                              867
                                          n_jobs=5,verbose=2,random_state=33)           868
```

Colon Random Forest Model Hyperparameters

```
rf = RandomForestClassifier(n_estimators=25,min_samples_split=3,                        870
                              max_depth = 10,                                           871
                              max_features = .5,                                        872
                                n_jobs=5,verbose=2,random_state=3)                      873
```

Lung Random Forest Model Hyperparameters

```
rf = RandomForestClassifier(n_estimators=25,min_samples_split=3,                        875
                              max_depth = 11,                                           876
                              max_features = .8,                                        877
                                n_jobs=5,verbose=2,random_state=3)                      878
```

Breast Neural Network Model Architecture

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

```
modelbreast = Sequential()                                                             882
modelbreast.add(Dense(114, input_shape=(66,) ,init='normal'))                          883
modelbreast.add(Activation('relu'))                                                    884
modelbreast.add(Dropout(0.05))                                                         885
modelbreast.add(Dense(50, init='normal'))                                              886
modelbreast.add(Activation('relu'))                                                    887
modelbreast.add(Dropout(0.05))                                                         888
                                                                                       889
modelbreast.add(Dense(36, init='normal'))                                              890
modelbreast.add(Activation('relu'))                                                    891
modelbreast.add(Dropout(0.05))                                                         892
                                                                                       893
modelbreast.add(Dense(2, init='normal'))                                               894
modelbreast.add(Activation('softmax'))                                                 895
                                                                                       896
rms = RMSprop(lr=0.001)                                                                897
                                                                                       898
modelbreast.compile(loss='binary_crossentropy',                                        899
            optimizer=rms, class_mode="binary")                                        900
                                                                                       901
```

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

Colon Cancer Neural Network Model Architecture

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

```
modelcolon = Sequential()                                                              908
modelcolon.add(Dense(114, input_shape=(102,) ,init='normal'))                          909
```

```
modelcolon.add(Activation('relu'))                                   910
modelcolon.add(Dropout(0.05))                                        911
modelcolon.add(Dense(50, init='normal'))                             912
modelcolon.add(Activation('relu'))                                   913
modelcolon.add(Dropout(0.05))                                        914
                                                                     915
                                                                     916
modelcolon.add(Dense(35, init='normal'))                             917
modelcolon.add(Activation('relu'))                                   918
modelcolon.add(Dropout(0.05))                                        919
                                                                     920
modelcolon.add(Dense(2, init='normal'))                              921
modelcolon.add(Activation('softmax'))                                922
                                                                     923
rms = RMSprop(lr=0.001)                                              924
                                                                     925
modelcolon.compile(loss='binary_crossentropy',                       926
          optimizer=rms, class_mode="binary")                       927
                                                                     928
```

and trained with a batch size of 1500 for 200 epochs.   929

Lung Cancer Neural Network Model Architecture   930

The archictecture of the Keras multilayer perceptron neural network model trained on   931
the lung cancer data is given explicitly below: 932

```
                                                                     933
modellung = Sequential()                                             934
modellung.add(Dense(114, input_shape=(114,) ,init='normal'))         935
modellung.add(Activation('relu'))                                    936
modellung.add(Dropout(0.1))                                          937
modellung.add(Dense(80, init='normal'))                              938
modellung.add(Activation('relu'))                                    939
modellung.add(Dropout(0.1))                                          940
modellung.add(Dense(40, init='normal'))                              941
modellung.add(Activation('relu'))                                    942
modellung.add(Dropout(0.1))                                          943
                                                                     944
                                                                     945
modellung.add(Dense(2, init='normal'))                               946
modellung.add(Activation('softmax'))                                 947
                                                                     948
                                                                     949
rms = RMSprop(lr=0.001)                                              950
                                                                     951
modellung.compile(loss='binary_crossentropy',                        952
            optimizer=rms, class_mode="binary")                     953
                                                                     954
```

and trained with a batch size of 2000 for 50 epochs.   955

Acknowledgments

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/lD7Jwf`.

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. Van Poucke S, Zhang Z, Schmitz M, Vukicevic M, Laenen MV, Celi LA, et al. Scalable predictive analysis in critically ill patients using a visual open data analysis platform. PLoS ONE. 2016;11(1). Cited By 0. Available from: `http://www.scopus.com/inward/record.url?eid=2-s2.0-84953931466&partnerID=40&md5=7a0cad7137c03146e4b75f3295f84cc6`.

8. 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.

9. 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.

10. 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`.

11. 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`.

12. 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.

13. National Cancer Institute, the Surveillance, Epidemiology, and End Results Program. Documentation for ASCII Text Data Files - SEER Datasets; 2016 (accessed 15 Jan 2016). http://seer.cancer.gov/data/documentation.html.

14. Michael Bowles. Machine Learning in Python: Essential Techniques for Predictive Analysis. Wiley; 2015.

15. 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.

16. 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.

17. 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.

18. IOBS. Supplemental Material — PAPERDATA; 2016 (accessed 18 Jan 2016. https://github.com/doolingdavid/PAPERDATA.git.

19. Ben Kuhn. Decision trees for survival analysis; 2016 (accessed 14 Jan 2016). http://www.benkuhn.net/survival-trees.

20. Allen Downey. Think Stats. O'Reilly Media; 2014.

21. James Surowiecki. The Wisdom of Crowds. Doubleday; 2004.

22. John Cassidy. What killed Intrade?; 13 Mar 2013 (accessed 25 Jan 2016). http://www.newyorker.com/news/john-cassidy/what-killed-intrade.

23. Malcolm Gladwell. Outliers. Back Bay Books; 2011.

24. 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.

25. Kaggle Inc . Random Forests — Kaggle; 2015 (accessed 25 Jan 2016. https://www.kaggle.com/wiki/RandomForests.

26. Michael Nielsen. Neural Networks and Deep Learning; Jan 2016 (accessed 25 Jan 2016). http://neuralnetworksanddeeplearning.com/.

27. 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.

28. F Chollet. Keras Documentation; 2015 (accessed 25 Jan 2016). http://keras.io/.

29. Armin Roncaher. Welcome — Flask (A Python Microframework; 2014 (accessed 29 Jan 2016). http://flask.pocoo.org.

30. 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.

31. 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.

32. National Cancer Institute, Division of Cancer Control and Population Sciences. SEER-Medicare Linked Database; 2015 (accessed 10 Feb 2016). http://healthcaredelivery.cancer.gov/seermedicare/.