Machine Learning for Survival Analysis: A New Approach

David Dooling[1,◑], Angela Kim[1,‡], Jennifer Webster[1,◑]

1 Innovative Oncology Business Solutions, Albuquerque, NM, USA

◑**These authors contributed equally to this work.**
‡**These authors also contributed equally to this work.**
* 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.

Author Summary

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur eget porta erat. Morbi consectetur est vel gravida pretium. Suspendisse ut dui eu ante cursus gravida non sed sem. Nullam sapien tellus, commodo id velit id, eleifend volutpat quam. Phasellus mauris velit, dapibus finibus elementum vel, pulvinar non tellus. Nunc pellentesque pretium diam, quis maximus dolor faucibus id. Nunc convallis sodales ante, ut ullamcorper est egestas vitae. Nam sit amet enim ultrices, ultrices elit pulvinar, volutpat risus.

Introduction

Opportunities are emerging in many indutries today to develop and deploy services that cater to individual needs and preferences. Music afficianados can create their own radio stations tailored to their individual tastes from Pandora[1], bibliophiles can receive highly trustworthy book recommendations from goodreads.com[2], and Google will provide directions between any two points, giving options such as mode of transportation and as well as warnings of delays in realtime.[3] These individualized services share many

---

[1]Pandora Internet Radio - Listen to Free Music You'll Love, `http://www.pandora.com/` (accessed 27 Jan 2016)

[2]Share Book Recommendations With Your Friends, Join Book Clubs, Answer Trivia, `https://www.goodreads.com/` (accessed 27 Jan 2016)

[3]Google Maps, `https://goo.gl/1D7Jwf` (accessed 27 Jan 2016)

common features. In particular, they leverage large databases of aggregated information 8
to learn and extract information relevant to individuals. Extracting actionable 9
information from data is changing the fabric of modern business. A class of techniques 10
that transforms data into actionable information goes by the name of Machine 11
Learning [1]. Machine Learning has recently become a popular method to answer 12
questions and solve problems that are too complex to solve via traditional methods. 13

The primary objective of this study is to show how machine learning methods can be 14
trained with data in cancer registries to produce personalized survival prognosis curves, 15
but the methods presented below can be applied to any type of survival data. 16
Traditionally, cancer survival curves have been estimated using Kaplan-Meier 17
methods [2]. Kaplan-Meier methodology also uses large datasets to make predictions, 18
but the resulting information is not personal; the resulting curves are summaries for a 19
population and not necessarily relevant or particularly accurate for any given individual. 20
This propery of Kaplan-Meier methods is exacerbated when dealing with heterogeneous 21
populations. The methods described below also take full advantage of all relevant 22
aggregate information, but are able to provide personalized survival curves relevant to 23
individual subjects. This objective is in keeping with the recent movement in medicine 24
known as Predictive, Preventive and Personalized Medicine (PPPM), which aims to 25
leverage increasing amounts of health related data to maximize quality of care and to 26
intelligenctly eliminate inefficient and unecessary use of resources [3]. This capability of 27
providing individualized survival curve prognosis is a direct result of the recent advances 28
in computing power and machine learning algorithms, and similar methodology is 29
becoming commonplace in many industries. These techniques are now infiltrating the 30
healthcare industry, in spite of some of the data aggregation challenges posed by the 31
Health Insurance Portability and Accountability Act (HIPPA) of 1996. This study 32
makes use of a freely available data source that circumvents the restrictions imposed by 33
HIPPA. 34

The Surveillance, Epidemiolgy, and End Results (SEER) Program of the National 35
Cancer Institute (NCI) has been collecting data because intuitively researchers feel 36
confident that this data will eventually allow researches to detect information crucial to 37
patients and providers including the relationships between the types of data collected 38
(demographic as well as staging information, treatment and disease characteristics) and 39
the survival outcomes. Though these relationships evade capture by traditional 40
methods, it is possible to surface them with two machine learning techniques known as 41
Random Forests and *Neural Networks*. As will be demonstrated in section , these two 42
methods produce very similar results when applied to the SEER dataset, and are based 43
on almost diametrically opposed learning philosophies, which lends confidence in the 44
validity of the results. 45

The Surveillance, Epidemiolgy, and End Results (SEER) Program of the National 46
Cancer Institute (NCI) is the most recognized authoritative source of information on 47
cancer incidence and survival in the United States. SEER currently collects and 48
publishes cancer incidence and survival data from population-based cancer registries 49
covering approximately 28 percent of the US population. 50

Quoting directly from the SEER website [4]: 51

> The SEER program registries routinely collect data on patient 52
> demographics, primary tumor site, tumor morphology and stage at diagnosis, 53
> first course of treatment, and follow-up for vital status. This program is the 54
> only comprehensive source of population-based information in the United 55
> States that includes stage of cancer at the time of diagnosis and patient 56
> survival data. The mortality data reported by SEER are provided by the 57
> National Center for Health Statistics. The population data used in 58
> calculating cancer rates is obtained periodically from the Census Bureau. 59

Updated annually and provided as a public service in print and electronic formats, SEER data are used by thousands of researchers, clinicians, public health officials, legislators, policymakers, community groups, and the public.

One characterstic of the SEER data that is shared by many datasets in the medical field goes by the name of "censored data." Observations are labeled censored when the survival time information is incomplete. The SEER data contains the number of months each patient survived, as well as an indicator variable showing whether or not the patient is still alive at the end of the data collection period. Methods to deal effectively with this kind of "right-censored data" include Kaplan-Meier curves and Cox Proportional Hazard models [2]. The Kaplan-Meier techniques only give estimates for cohorts of patients and are not applicable for predicting the surival curve for a single patient, and the Cox Proportional Hazard models require a fairly restrictive set ot assumptions to be satisifed in order to yield reliable results.

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. [5] use semi-supervised learning techniques to predict 5 year survival, essentially imputing values for SEER records where the survival months infomation is censored at a value less than 5 years. Zolbanin et al. [6] investigate the effects of comordbidities; i.e., patients with two different cancer diagnosises, but their treatment of the censored data underestimates the survival probabilities. All records representing patients who survived at least 60 months as well as all those who died earlier than 60 months were considered, but patients alive prior to 60 months but censored out of the study before 60 months were not included. This treatment biases the data and the predictions, leading to overly pessimistic survival probablilites predicted by the models.

Previous work applying machine learning methods based on decision trees to survival data in general have a long history, starting with Gordon et al. [7]. A summary of more recent developments concerning *survival trees* is provided by Bou-Hamad et al. [8]. These methods focus on altering the splitting critieria used in decision tree growth to account for the censoring, and use 1958 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* [9].

IOBS has applied a little-known technique to transform the SEER data to make it amenable to more powerful machine learning methods. Instead of modifying existing learning algorithms in drastic ways, we focus attention on the input data. This approach allows for different machine learning algorithms to use the same data with no modification. The essential idea is to recast the problem to an appropriate discrete classification problem instead of a regression problem (predicting survival months). Treating months after diagnosis as just another discrete feature, the SEER data (or any other right-censored data) can be transformed to make predictions for the hazard function (probability of dying in the next month, given that the patient has not yet died). The full survival function can then be derived from the hazard function.

This paper is organized as follows. We introduce the subsets of the SEER data used for this study, and present survival curves computed from traditional methods based on this data for the three cancer types *lung*, *breast*, and *colon*. We then present the essential methodology of this work, the data trasnformation that allows censored survival data to be used as input to exisiting machine learning classifiers. Then we present the details of the trained models, including some some subtleties arising from the data transformation pertaining to the partition into training and test datasets. The method of deriving binary classifiers from the models' predictions for the survival curves is presented. In this paper, we have constructed binary classifiers corresponding to 6, 12, and 60 months, as these are standard metrics in cancer survival prognosis. Then follows

a dicussion of the evaluation of the trained models. The performance metrics are the 18   112
AUC curves associated with the 6, 12, and 60 month survival binary classifiers for the   113
two models associated with each cancer type. We also present additional evidence   114
supporting validity of the predictions by computing the levels of agreement between the   115
random forest and neural network models for each of the 18 binary classifiers and find   116
striking agreement. Next we provide urls for 6 web applications that use the trained   117
models to predict individual cancer survival prognosis curves. These apps are hosted on   118
the popular Heroku website, and allow for exploration of the nonlinear relationships   119
between the input features and resulting survival prognosis. It is exactly these kinds of   120
tools that are the goal of Predictive, Preventitive and Personalized Medicine. Finally,   121
we present avenues for future research.   122

Materials and Methods   123

For this study we use the publically available 1973-2012 SEER incidence data files   124
corresponding to colon, breast and lung cancer contained in the list below. SEER   125
requires that researchers submit a request for the data, which includes an agreement   126
form. Detailed documentation explaining the contents of both the incidence data files   127
used in this study as well as a data dictionary for the 1973-2012 SEER incidence data   128
files are available without the need to register or submit a data request [10].   129

- incidence\yr1973_2012.seer9\COLRECT.txt   130
- incidence\yr1973_2012.seer9\BREAST.txt   131
- incidence\yr1973_2012.seer9\RESPIR.txt   132
- incidence\yr1992_2012.sj_la_rg_ak\COLRECT.txt   133
- incidence\yr1992_2012.sj_la_rg_ak\BREAST.txt   134
- incidence\yr1992_2012.sj_la_rg_ak\RESPIR.txt   135
- incidence\yr2000_2012.ca_ky_lo_nj_ga\COLRECT.txt   136
- incidence\yr2000_2012.ca_ky_lo_nj_ga\BREAST.txt   137
- incidence\yr2000_2012.ca_ky_lo_nj_ga\RESPIR.txt   138
- incidence\yr2005.lo_2nd_half\COLRECT.txt   139
- incidence\yr2005.lo_2nd_half\BREAST.txt   140
- incidence\yr2005.lo_2nd_half\RESPIR.txt   141

Data preparation and preprocessing   142

A great deal of data munging is necessary before using these SEER incidence files as   143
input into machine learning algorithms. A preprocessing step common to each of the   144
three cancer types studied involves the SEER `STATE-COUNTY RECODE` variable. The   145
`STATE-COUNTY RECODE` field is a state-county combination where the first two   146
characters represent the state FIPS code and the last three digits represent the FIPS   147
county code. The FIPS code is a five-digit Federal Information Processing Standard   148
(FIPS) code which uniquely identifies counties and county equivalents in the United   149
States, certain U.S. possessions, and certain freely associated states. This particular   150
field illustrates an important characteristic of machine learning, that is, the difference   151
between *categorical features* and *numeric features*. All input into a machine learning   152
algorithm must be numeric, but real numbers carry with them the usually extremely   153
useful property known as the well-ordering property. Machine learning algorithms use   154
the well-ordering property of the real numbers to learn. But if one is tasked with   155
encoding a categorical feature into suitable numeric format for machine learning, it is   156
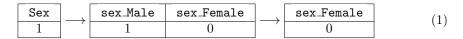necessary to do so in a way that removes the well-ordering property [11].   157

As a simple example of how to correctly treat categorical variables in a machine   158
learning context, consider the SEER variable `SEX`. This variable is encoded in the   159

Code	Description
1	Male
2	Female

Table 1. Encoding of gender in the SEER incidence files. These types of categorical variables need to be transformed via one-hot-encoding.

SEER raw data files with a numeric 1 for males and a numeric 2 for females as shown in Table (1). Values such as "Male" and "Female" encoded as numbers are dangerous because if not handled properly, they can generate bogus results [12]. Leaving the infomation for `SEX` as in Table (1) implies that Female is somehow greater than Male. This implied ordering affects the machine learning algorithms' convergence on a model. Simply encoding Male by 2 and Female by 1 would result in a comletely different model, because of the now completely reversed ordering implied in the `SEX` variable. The proper way to transform the SEER `SEX` variable is to create two additional variables: `sex_Male` and `sex_Female`, and then to eliminate the variables `SEX` and `sex_Male` (keeping both of the variables `sex_Male` and `sex_Female` is a redundant represetation). For example,

$$
\boxed{\begin{array}{c} \text{Sex} \\ \hline 1 \end{array}} \longrightarrow \boxed{\begin{array}{c|c} \texttt{sex\_Male} & \texttt{sex\_Female} \\ \hline 1 & 0 \end{array}} \longrightarrow \boxed{\begin{array}{c} \texttt{sex\_Female} \\ \hline 0 \end{array}} \tag{1}
$$

and

$$
\boxed{\begin{array}{c} \text{Sex} \\ \hline 2 \end{array}} \longrightarrow \boxed{\begin{array}{c|c} \texttt{sex\_Male} & \texttt{sex\_Female} \\ \hline 0 & 1 \end{array}} \longrightarrow \boxed{\begin{array}{c} \texttt{sex\_Female} \\ \hline 1 \end{array}} \tag{2}
$$

The procedure outlined in Equations (1, 2) is known as one-hot encoding and 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 [11]. 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 (2) shows how five entries of `STATE-COUNTY RECODE` corresponding to counties within New Mexico can be represented by the `elevation`, `lat`, and `lng` features.

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 dafafile from the United States Census Bureau [13] to map the state FIPS and county FIPS codes to query strings like those in the `address` field in Table (2). It is then possible to programmatically query the

Table 2. 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

Google Maps Geocoding API for the latitude and longitude [14], and the Google Maps Elevation API for the corresponding elevation [15]. 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 coverered 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 (2) is available from a GitHub repository containing supplemental information for this study [16].

This study focused on three different cancer types, namely colorectal cancer, lung cancer, and breast cancer. In the SEER data, there are instances of subjects with multiple rows; whenever a subject, or patient, is diagnosed with a new tumor, an additional record is added. In this study, we restrict attention to the data corresponding to the first record of each subject; i.e., we wish to make models that predict survival prognosis based on the data available right after diagnosis. The full set of conditions defining the subsets of the SEER data used in this study follows below.

The four COLRECT.txt files were imported into a pandas DataFrame object. This data was then filtered according to the conditions in Table (3). The RESPIR.txt and BREAST.txt files were imported into separate dataframes in similar fashion and filtered according to the conditions in Table (4) and Table (5), 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 (5, 3, 4).

Table 3. 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 4. 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 5. 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 ,
- SPANISH/HISPANIC ORIGIN ,
- GRADE ,
- PRIMARY SITE ,
- LATERALITY ,
- SEER HISTORIC STAGE A ,
- HISTOLOGY RECODE--BROAD GROUPINGS ,
- MONTH OF DIAGNOSIS ,
- VITAL STATUS RECODE ,

and the STATE-COUNTY RECODE variable was dropped and replaced with the elevation , lat , and lng variables for all three datasets as illustrated in Table (2).

Before applying machine learning models trained with these datasets, we review below the sailent features of survival analysis and censored data. We then 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   239
set of features and target variable listed in the back of this report.   240

Traditional Survival Analysis   241

Survival analysis pertains to data containing survival times, which are *intervals* between   242
certain kinds of events, e.g.; cancer diagnosis date and expiry date. These intervals are   243
often affected by a kind of "partial missingness" called *censoring*. Censored data must   244
be analyzed in a special way to avoid biased estimates and bogus conclusions. Special   245
methods have been developed long ago to analyze censored data properly.   246

   With survival data, including the SEER data considered in this study, you may not   247
know the exact time of death for some subjects. Some of the SEER subjects are still   248
alive at the the time of the latest SEER data release. When the   249
`VITAL STATUS RECODE` variable indicates that the subject is still alive, the   250
`SURVIVAL MONTHS` variable is only a lower bound on the true number of survival   251
months; this is called the *date of last contact* mode of censoring. You know that each   252
subject either died on a certain date or was definitely alive up to some last-seen date   253
(and you don't know how far beyond that date he or she may ultimately have lived).   254
The latter situation is called a *censored* observation.   255

   Statisticians have developed some traditional techniques to utilize the partial   256
information contained in censored observations: the life-table method and the   257
Kaplan-Meier method. Both of these methods make use of the partial information to   258
provide unbiased estimates of the two fundamental concepts: - *hazard* and *survival*,   259
both of which are functions of time: 260

- **The hazard rate** $\lambda(t)$ is the probability of dying in the next small interval of   261
  time, assuming that the subject is alive right now.   262
- **The survival rate** $S(t)$ is the probability of living for a certain amount of time   263
  after some starting point.   264

   Incorrect treatment of survival data still seen in practice, and leading to biased   265
results, includes simply excluding all subjects with a censored survival time from any   266
survival analysis, and *imputing* (replacing) the censored (last-seen) date with some   267
reasonable value. Both of these techniques destroy the partial information contained in   268
the censored observations and nullify the validity of the resulting estimates for the   269
hazard rate and survival rate [2].   270
   In 1958, Edward L. Kaplan and Paul Meier collaborated to publish the seminal   271
paper on how to estimate the hazard and survival rates for data containing censored   272
observations [17]. The method is straightforward and for small datasets can be   273
performed by hand. As an example, consider the survival data shown in Table (6). In   274
the Kaplan-Meier calculation of the survival curve, the first step is to sort the subjects   275
in Table (6) labeled 0 through 9 by *Survival Time* in ascending order. This process   276
results in the first two columns (*Censored Status*, and *Survival Times*) in Table (7).   277
The *At Risk* column decreases by one for each row; in every row a subject has either   278
been censored out of the study or has died. The hazard rate is then computed for each   279
value of *Survival Time* (necessarily a discrete function because the number of subjects is   280
countable), by dividing the value in *Censored Status* by the value in *At Risk*. The   281
hazard function is shown in the *Hazard Function* column in Table (7). It is then   282
straightforward to calculate the survival function; 1 - hazard function represents the   283
probability of not dying in the next interval of time, assuming that the subject has   284
survived up until now and is represented by column *Prob of Surv*. The cumulative   285
survival probability can then be obtained by sucessively multiplying all these individual   286
time-slice probabilities together. In order to survive 2.4 years, first the subject has to   287

survive .5 years, then survive .75 years, 2.3 years and 2.4 years. The probability of 288
surviving 2.4 years is then the product of these 3 probabilities and is given as .666 in 289
Table(7) in the *Survival Function* column. The Kaplan-Meier survival estimate 290
corresponding to the data given in Table (6) is shown in Table (7). 291

Table 6. Example data to illustate traditional Survival Analsyis.

	Survival Time (Years)	Censored Status
0	0.75	1
1	6.10	1
2	7.00	0
3	2.40	1
4	0.50	0
5	4.50	1
6	3.50	0
7	5.80	0
8	2.30	1
9	5.20	1

Table 7. Kaplan-Meier table corresponding to the example data in Table (6).

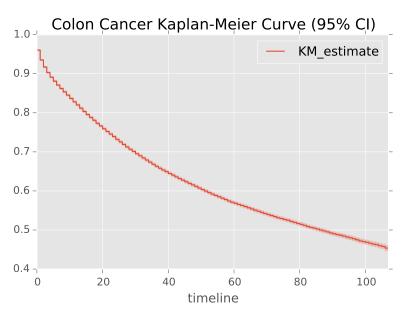|	Censored Status	Survival Time	At Risk	Hazard Function	Prob of Surv	Survival Function
4	0	0.50	10	0.000000	1.000000	1.000000
0	1	0.75	9	0.111111	0.888889	0.888889
8	1	2.30	8	0.125000	0.875000	0.777778
3	1	2.40	7	0.142857	0.857143	0.666667
6	0	3.50	6	0.000000	1.000000	0.666667
5	1	4.50	5	0.200000	0.800000	0.533333
9	1	5.20	4	0.250000	0.750000	0.400000
7	0	5.80	3	0.000000	1.000000	0.400000
1	1	6.10	2	0.500000	0.500000	0.200000
2	0	7.00	1	0.000000	1.000000	0.200000

After the above one-hot encoding procedure, the new variable 292
`vital_status_recode_Dead` indicates that the patient is deceased if this variable $= 1$, 293
or else that the patient's record is right-censored if this variable $= 0$. 294
`SURVIVAL MONTHS` and `vital_status_recode_Dead` are all that is needed to 295
construct the Kaplan-Meier estimates for the SEER datasets. The Kaplan-Meier 296
estimates of the survival curves for colon (Figure (1)), lung (Figure (3)), and breast 297
cancer (Figure (2)) are constructed from the full population of cancer patients in the 298
respective datasets. An unsatisfactory feature of these curves is that these estimates are 299
based on populations and data with enough heterogeneity to make them not very 300
meaningful to an indivual. Patients with very disparate characteristics are given the 301
same prognosis by these Kaplan-Meier survival curve estimates. Therefore it is desirable 302
to find robust predictors for survival curves of individual subjects where the input is an 303
individual record as opposed to a population. We present below the data transformation 304
that allows for machine learning to be applied to censored data. 305

Transformation of Censored Data for Machine Learning 306

In this section we describe an inuitive way to transform right-censored data 307
appropriately so that it may be used as input to machine learning algorithms that learn 308

Figure 1. Traditional Kaplan-Meier estimate of the survival curve for all colon cancer patients. Fitted with 113072 observations, 71804 censored.

Figure 2. Traditional Kaplan-Meier estimate of the survival curve for all breast cancer patients. Fitted with 329949 observatins, 292279 censored.

Figure 3. Traditional Kaplan-Meier estimate of the survival curve for all lung cancer patients. Fitted with 177089 observatins, 47409 censored.

the hazard fuction. The full details of this transformation, and a large inspiration for this study, can be flound in this blog post [18].

The overall philosophy of the Kaplan-Meier estimate of the survival curve for a population differs fundamentally from the methods described below and used in this study. The Kaplan-Meier estimate of the survival curve is given by
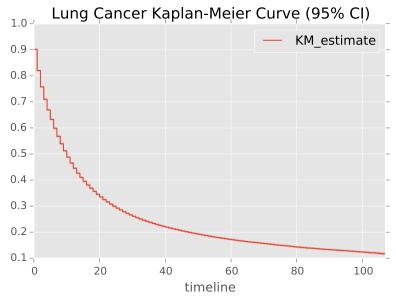
$$\hat{S}(t) = \prod_{t_i < t} \frac{n_i - d_i}{n_i} \tag{3}$$

where d_i are the number of death events at time t and n_t is the number of subjects at risk of death just prior to time t. Equation (3) uses the entire data set to arrive at an estimate of the entire population survival curve. In contrast, the method described below uses the entire data set to learn a model so as to predict hazard and survival curves from the data for as yet unseen individuals.

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

$$\lambda(\mathbf{X}, t) = P(Y = t | Y \geq t, \mathbf{X}), \tag{4}$$

the probability that, if someone has survived up until month t, they will die in that month. where $\mathbf{X}$ represents all of the data for that particular record, and in our case 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 4 that

$$P(Y = t | \mathbf{X}) = \lambda(\mathbf{X}, t) \prod_{i=1}^{t-1}(1 - \lambda(\mathbf{X}, i)) \tag{5}$$

, which is the full probablity distribution of dying at time Y [18]. The survival function is then readily derived from this distribution as

$$S(\mathbf{X}, t_k) = 1 - CDF(\mathbf{X}, t_k) \qquad (6)$$

where $CDF(\mathbf{X}, t_k) = \sum_{k=1}^{n} P(Y = t_k | \mathbf{X})$ is the cumulative density function correponding to the probability mass function in equation 5 [12].

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, and the classification problem is "did point $\mathbf{X}_i$ die in month t." We will call this new variable D_{it} (`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_{it} as the target, and derive a survival model from the orginal dataset. Psuedocode for this transformation is found in section Pseudocode for the Data Transformation.

Explicit examples will help make this transformation clear. The untransformed datapoint represented Table (8) is transformed to the multiple records shown in Table (10). All uncensored data is transformed in this way. All censored data is similarly transformed. The untransformed datapoint represented Table (9) is transformed to the multiple records shown in Table (11).

Table 8. Example of four columns in an uncensored record in the untransformed dataset.

	cs_tumor_size	year_of_birth	survival_months	vital_status_recode_Dead
newindex				
205	60	1951	3	1

Table 9. Example of four columns in a censored record in the untransformed dataset.

	cs_tumor_size	year_of_birth	survival_months	vital_status_recode_Dead
newindex				
205	40	1950	3	0

Table 10. Example of four columns in an uncensored record 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

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

Table 11. Example of four columns in a censored record in the transformed dataset.

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

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.

Training and Test Partitions

After performing the data transformation adumbrated above, it is necessary to be mindful of how we partition the data into training and testing data. Each subject that was represented by a single row in the original untransformed dataset now potentially is represented by multiple rows in the transformed dataset, and care must be taken to ensure that all of the rows corresponding to a particular subject are either assigned exclusively to the training set or exclusive to the testing 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`. Fortunately, 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 thus learning 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 (8, 9, 10, 11). The information of which subjects represent censored data

(`vital_status_recode_Dead` $== 0$) and which died is retained and recoverable trough 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 that we know definitively whether or not they survived 6, 12 or 60 months respectively. This requirement will necessitate the elmination of some of the censored data when computing some of the performance metrics. 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" [19]. 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 [20]. The other class of methods used by IOBS to develop predictive models 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 [21].

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 Supporting Information section for each cancer type and 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. Fortunately, both random forests and neural networks are capable of not only performing strict class prediction, i.e. prediciting whether `newtarget` is 0 or 1, but are also able to predict the *probability* of `newtarget` being 0 or 1, and thus learning the hazard function. The models learn $\lambda(\mathbf{X}, $ `months` $)$. This prediction task should not be confused with the regression problem of trying to predict precisely in what month a patient will die.

The hazard functions thus learned and predicted are intermediary products; what we are really pursuing are the survival functions for each patient that are derived from the predicted hazard functions. From the resulting hazard functions for each unique patient, we can construct the resulting survival functions as presented in section () and Equation (**??**) and explicitly given in python code in the notebooks at the github repository containing supplemental material for this study [16]. 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
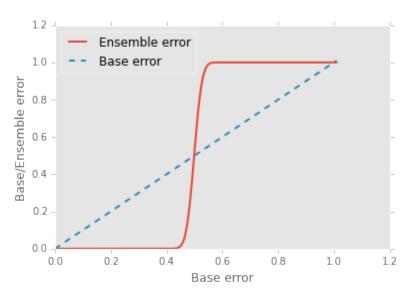
chosen for this study. ₄₄₂

Decision Trees and Random Forests *Decision tree* classifiers are attractive ₄₄₃ models because they can be intrepeted easily. Like the name decision tree suggests, we ₄₄₄ can think of this model as breaking down our data by making decisions based on asking ₄₄₅ a series of questions. Based on the features in our training set, the decision tree model ₄₄₆ learns a series of questions to infer the class labels of the samples. ₄₄₇

Random forests have gained huge popularity in applications of machine learning ₄₄₈ during the last decade due to their good classification performance, scalability, and ease ₄₄₉ of use. Intuitively, a random forest can be considered as an *ensemble of decision trees.* ₄₅₀ The idea behind ensemble learning is to combine *weak learners* to build a more robust ₄₅₁ model, a *strong learner*, that has a better generalization error and is less susceptible to ₄₅₂ overfitting. ₄₅₃

The goal behind *ensemble methods* is to combine different classifiers into a ₄₅₄ meta-classifier that has a better generalization performance than each individual ₄₅₅ classifier alone. For example, assuming that we collected predictions from 10 experts, ₄₅₆ ensemble methods would allow us to strategically combine these predictions by the 10 ₄₅₇ experts to come up with a prediction that is more accurate and robust than the ₄₅₈ predictions by each individual expert. The individual decision trees that make an ₄₅₉ ensemble are called base learners, and as long as the error rate of each base learner is ₄₆₀ less than .50, the combined random forest will benefit from the affects of combining ₄₆₁ predictions to achieve a far greater accuracy. ₄₆₂

Figure (4) illustrates the power of ensemble methods; the Figure illustrates how the ₄₆₃ ensemble error rate is much lower than the Base learner error rate, as long as the Base ₄₆₄ learner error rate is less than 0.5. The Figure illustrates this effect for an ensemble of ₄₆₅ 500 base learners. ₄₆₆

Figure 4. Illustration of ensemble methods showing how a collection of base learners with poor accuracy can combine to produce an accurate ensemble learner.

A big advantage of random forests is that honing in on suitable hyperparameter ₄₆₇ values (the number of trees in the forest, the depth of each decision tree, the specific ₄₆₈ measure of information gain used to choose the node splitting, etc) is not very difficult. ₄₆₉ The ensemble method is robust to noise from the individual decision trees, which helps ₄₇₀ to prevent overfitting (memorizing the training dataset targets instead of generalizing ₄₇₁

from learned rules to perform successfuly on unseen data). The only parameter that has   472
a clearly noticeable effect on performance is the number of trees to include in the forest;   473
in general, the more trees the better the performance, but there is a price to pay in   474
terms of computational cost. The number of trees for the forests trained in this study   475
was relatively small, 20 trees for breast cancer and 25 for both the lung and colon   476
cancer models.   477

    IOBS has chosen to use the Python scikit-learn implemenation of the Random Forest   478
machine learning classifier [22]. Random Forests are frequent winners of the Kaggle   479
machine learning competitions [23]. The model parameters for each cancer type are   480
given in sections (Lung Random Forest Model Hyperparameters, Colon Random Forest   481
Model Hyperparameters, Breast Random Forest Model Hyperparameters).   482

Multi-Layer Perceptron Neural Networks   Neural networks are a   483
biologically-inspired programming paradigm that enable computers to learn from   484
observational data [24]. Deep learning can be understood as a set of algorithms that   485
were developed to train artificial neural networks with many layers most efficiently.   486
Neural networks are a hot topic not only in academic research, but also in big   487
technology companies such as Facebook, Microsoft, and Google who invest heavily in   488
artificial neural networks and deep learning research. As of today, complex neural   489
networks powered by deep learning algorithms are considered as state-of-the-art when it   490
comes to complex problem solving such as image and voice recognition. In addition, the   491
pharmaceutical industry recently started to use deep learning techniques for drug   492
discovery and toxicity prediction, and research has shown that these novel techniques   493
substantially exceed the performance of traditional methods for virtual screening [25].   494

    IOBS has chosen to use the Multi-Layer Perceptron Neural Network (MLP neural   495
network) implementation Keras developed at MIT. Keras was initially developed as part   496
of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent   497
Robot Operating System) [26]. Keras is a minimalist, highly modular neural networks   498
library, written in Python and capable of running on top of either TensorFlow or   499
Theano. The model architecture for each cancer type are given in sections (Breast   500
Neural Network Model Architecture, Colon Cancer Neural Network Model   501
Architecture, Lung Cancer Neural Network Model Architecture). Training a neural   502
network and choosing an appropriate architecture is as much art as science [24], and the   503
search for a good neural network architecture for the lung cancer case was more   504
demanding than for the breast and colon cases. The presence of both non-small cell   505
lung cancer (NSCLC) and small cell lung cancer (SCLC) in the SEER data may be the   506
source of this need for more iterations and trials of different architectures when training   507
the lung cancer neural network models.   508

Results   509

In order to evaluate the performance of the models, we first construct three binary   510
classifiers corresponding to whether or not a subject survived 6, 12, or 60 months after   511
diagnosis. This is done by iterating over all distinct patient indices in the test set,   512
prediciting the full survival function, and capturing the values corresonding to 6, 12,   513
and 60 months. If the survival function evaluted at 6 months is greater than or equal to   514
.5 for a given subject, then the 6 months binary classifier predicts that that subject will   515
be alive 6 months after diagnosis. Similarly, if the survival function evaluted at 60   516
months is less than .5, then the 12 months binary classifier predicts that that subject   517
will be dead 12 months after diagnosis. Figure (5) illustrates the method; in this case   518
the 6-month and 12-month classifiers predict survival, while the 60-month classifier   519
predicts expiry.   520
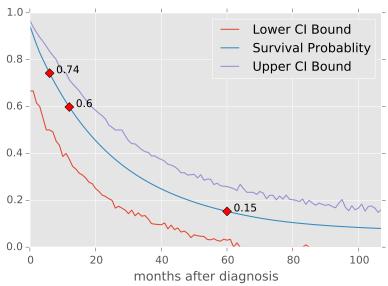
Figure 5. Example of the construction of the binary classifiers for 6, 12, and 60 months survival. A subject's hazard curve $\lambda(\mathbf{X}, t)$ is predicted by the model for times out to 107 months. The survival curve is then readily computed as in Equation (6). For this example, the 6-month and 12-month classifiers predict survival, while the 60-month classifier predicts expiry.

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 standard calculation of confidence intervals used in the Kaplan-Meier estimates of survival curves does not apply for these personal predictions. The following bootstrap method was used to calculate the upper and lower bounds corresponding to 95% confidence intervals. From equation 6, 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 represent one row of the training data). A subject 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 544
  points to draw from the survival curve CDF, and compute a new survival curve 545
- Repeat the previous step 10,000 times and collect the curves into a list. Changing 546
  the number of curves affects how smooth the upper and lower bounds are, but 547
  does not affect the interval size between for each month. 548
- extract for each month from the list of curves the .975 and .025 percentiles to 549
  record the values for the upper and lower curves 550

The process is somewhat anologous to the following hypothetical situation. Imagine 551
a patient going to an expert, and the expert after collecting data on the patient and 552
keeping records predicts the central, single survival curve. The patient then seeks 553
multiple "second opinions." These second opinions are generated not from independent 554
examinations of the patient, but by outside experts sampling from the data already 555
collected by the expert initially consulted. Then the predictions of 95% of these 10,000 556
experts all fall within the band determined by the upper and lower curves. 557

Performance Metrics 558

The AUC scores for each of the 18 different binary classifiers are listed in Table (12). 559
We emphasize the above-mentioned discussion concerning the correct treatment of the 560
censored test data when evaluating performance metrics. Namely, when computing the 561
AUC for the 12 month survival curve classifiers, we restrict the test data subjects to 562
those that in the untransformed data set that satisfy either of the following mutually 563
exclusive conditions: 564

- `survival_months` $>= 12$ AND `vital_status_recode` $== 0$ 565
- `vivtal_status_recode` $== 1$ 566

We limit evaluation data to subsets of the data where we know for certain whether 567
or not the subject survived at least 12 months. Similar considerations apply to the 12 568
and 60 months AUC calculations. The lowest AUC in Table 12 is .765, corresponding to 569
the lung neural network model predictions for 6 months survival, while the highest AUC 570
in Table 12 is .885, corresponding to the breast random forest model predictions for 12 571
months survival. 572

Table 12. 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 573

An additional means of validating the predictions of these models is by comparing their 574
predictions to each other for the same set of input data. Table 13 shows the strong 575
agreement between the random forest and neural network classifiers for each cancer 576

type. Python code showing how the values in Table 13 are computed is available in the 577
files `NewPatientBreastCF.html`, `NewPatientColonCF.html`, and 578
`NewPatientLung.html` in the GitHub repository containing supplemental matierial for 579
this study [16]. Table 13 is computed as follows. For each cancer type (breast,colon, and 580
lung), do the following: 581

- use the corresponding Random Forest and Neural Network models to compute the 582
  survival curves for all of the test subjects 583
- extract the values of the survival curve evaluted for 6, 12, and 60 months for both 584
  models 585
- if both models predict less than .5 or both models predict greater than or equal to 586
  .5, that counts as agreement 587
- otherwise, the models disagree 588

The high level of agreement between two models lends confidence to the notion that 589
they have both learned from the training data and are generalizing well. 590
Figures (7, 6, 8) show box plots of the value of the random forest prediction subtracted 591
from the neural network prediction. We emphasize that when evaluating the model 592
agreement, we put no restrictions on the distinct subjects in the respective test datasets; 593
we are confronting the models against each other, not some known ground truth as in 594
the AUC performance metric calculations. The number of distinct subjects in all three 595
of the colon cancer survival binary classifiers (6, 12, and 60 month survival) was 5654; 596
for lung cancer the number of subjects entering into the calculation of Table (13) was 597
5313; and for breast cancer it was 3300. 598

Table 13. 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

Survival Curve Prediction Apps 599

The six models described in section Prediction Models, namely the random forest and 600
MLP neural network models for each of the three cancer types considered in this study, 601
have their full hyperparameter and architecture presented in section Supporting 602
Information. Python code for all six model training and evaluation is available at the 603
githib respository containing supplemental material for this study [16]. 604

Using the popular Flask microframework for web applications [27], we have made 605
web applications corresponding to the six models. The list of web applications below 606
will allow readers to freely experiment with the models. 607

1. breast cancer 608
   (a) random forest: 609
       https://github.com/doolingdavid/breast-cancer-rf-errors.git 610
   (b) neural network: 611
       https://github.com/doolingdavid/breast-cancer-nn-errors.git 612
2. lung cancer 613
   (a) random forest: 614
       https://github.com/doolingdavid/lung-cancer-rf-errors.git 615

Figure 6. 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 7. 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.
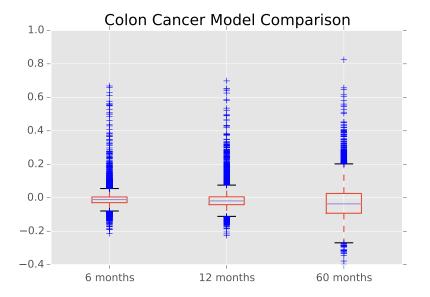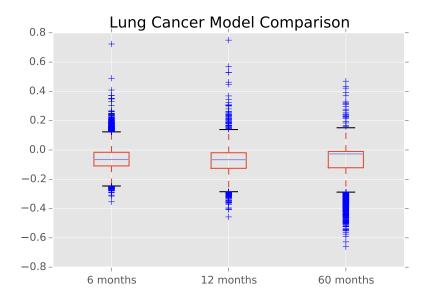
Figure 8. Box plots showing the distributions of the signed difference between the MLP model's prediction for the probability of surviving 6 months and the Random Forest model's prediction of the same quantity for lung cancer. The plot shows the same quantity for the 12 and 60 months classifiers. These 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 6 and fig 7.

    (b) neural network: 616
        `https://github.com/doolingdavid/lung-cancer-nn-errors.git`   617

3. colon cancer   618

    (a) random forest: 619
        `https://github.com/doolingdavid/colon-cancer-rf-errors.git`   620
    (b) neural network: 621
        `https://github.com/doolingdavid/colon-cancer-nn-errors.git`   622

After downloading the .zip file associate with one of the above web applications, and   623
assuming python is installed on your system, you can launch the application by running   624

`>python hello.py`   625

and pointing the browser to the local server: `http://127.0.0.1:5000` .   626

These machine learning models are used to predict survival curves for a given set of   627
input data. The resulting surival curves predict the probablitiy that a patient with the   628
given input data will survive at least up to month x.   629

For example, using the Colon Cancer neural network app, and inputing the values   630
listed in Table (14) results in the survival curve depicted in Figure (9); the predicted   631
probablities of living at least 6, 12, and 60 months are .89, .83, and .50, respectively.   632

Changing the data in Table 14 so that the address field is changed from Boston,   633
Massachusetts to Denver, Colorado but keeping all other variables are unchanged results   634
in the predicted probabilities of living at least 6, 12, and 60 months: .945, .902, .665.   635
Behind the scenes, the apps use the input to the address field to make a call to the   636
Google Maps API to convert the address into a latitude, longitude and elevation. These   637

Table 14. 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

probablities are noticeably higher and reflect the documented effects of both longitude and elevation on cancer treatment and prognosis in the United States [28].

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 15 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

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

Nulla mi mi, venenatis sed ipsum varius, Table ?? volutpat euismod diam. Proin rutrum vel massa non gravida. Quisque tempor sem et dignissim rutrum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi at justo vitae nulla elementum commodo eu id massa. In vitae diam ac augue semper tincidunt eu ut eros. Fusce

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 9. Colon Cancer Survival Curve predicted from the data in Table (14) using the neural network web app
`https://github.com/doolingdavid/colon-cancer-nn-errors.git`.

fringilla erat porttitor lectus cursus, vel sagittis arcu lobortis. Aliquam in enim semper, aliquam massa id, cursus neque. Praesent faucibus semper libero.

LOREM and IPSUM Nunc blandit a tortor.

CO_2 Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque. Quisque augue sem, tincidunt sit amet feugiat eget, ullamcorper sed velit.

Sed non aliquet felis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris commodo justo ac dui pretium imperdiet. Sed suscipit iaculis mi at feugiat. Ut neque ipsum, luctus id lacus ut, laoreet scelerisque urna. Phasellus venenatis, tortor nec vestibulum mattis, massa tortor interdum felis, nec pellentesque metus tortor nec nisl. Ut ornare mauris tellus, vel dapibus arcu suscipit sed. Nam condimentum sem eget mollis euismod. Nullam dui urna, gravida venenatis dui et, tincidunt sodales ex. Nunc est dui, sodales sed mauris nec, auctor sagittis leo. Aliquam tincidunt, ex in facilisis

Table 15. 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

elementum, libero lectus luctus est, non vulputate nisl augue at dolor. For more   679
information, see S1 Text.   680

Supporting Information   681

Colon Cancer Feature Selection   682

The feature set used as input into both the Random Forest and Neural Network models,   683
after the transformation described in section Transformation of Censored Data for   684
Machine Learning is given below and also available in full detail in the file   685
`NewPatientColonML.html`.   686

- cs_tumor_size   687
- elevation   688
- grade_cell type not determined   689
- grade_moderately differentiated   690
- grade_poorly differentiated   691
- grade_undifferentiated; anaplastic   692
- grade_well differentiated   693
- histology_recode_broad_groupings_acinar cell neoplasms   694
- histology_recode_broad_groupings_adenomas and adenocarcinomas   695
- histology_recode_broad_groupings_blood vessel tumors   696
- histology_recode_broad_groupings_complex epithelial neoplasms   697
- histology_recode_broad_groupings_complex mixed and stromal neoplasms   698
- histology_recode_broad_groupings_cystic, mucinous and serous neoplasms   699
- histology_recode_broad_groupings_ductal and lobular neoplasms   700
- histology_recode_broad_groupings_epithelial neoplasms, NOS   701
- histology_recode_broad_groupings_fibromatuos neoplasms   702
- histology_recode_broad_groupings_germ cell neoplasms   703
- histology_recode_broad_groupings_lipomatous neplasms   704
- histology_recode_broad_groupings_miscellaneous bone tumors   705
- histology_recode_broad_groupings_myomatous neoplasms   706

- histology_recode_broad_groupings_neuroepitheliomatous neoplasms   707
- histology_recode_broad_groupings_nevi and melanomas   708
- histology_recode_broad_groupings_paragangliomas and glumus tumors   709
- histology_recode_broad_groupings_soft tissue tumors and sarcomas, NOS   710
- histology_recode_broad_groupings_squamous cell neoplasms   711
- histology_recode_broad_groupings_synovial-like neoplasms   712
- histology_recode_broad_groupings_transistional cell papillomas and carcinomas   713
- histology_recode_broad_groupings_unspecified neoplasms   714
- lat   715
- laterality_Left: origin of primary   716
- laterality_Not a paired site   717
- laterality_Only one side involved, right or left origin unspecified   718
- laterality_Paired site, but no information concerning laterality; midline tumor   719
- laterality_Right: origin of primary   720
- lng   721
- marital_status_at_dx_Divorced   722
- marital_status_at_dx_Married (including common law)   723
- marital_status_at_dx_Separated   724
- marital_status_at_dx_Single (never married)   725
- marital_status_at_dx_Unknown   726
- marital_status_at_dx_Unmarried or domestic partner   727
- marital_status_at_dx_Widowed   728
- month_of_diagnosis_Apr   729
- month_of_diagnosis_Aug   730
- month_of_diagnosis_Dec   731
- month_of_diagnosis_Feb   732
- month_of_diagnosis_Jan   733
- month_of_diagnosis_Jul   734
- month_of_diagnosis_Jun   735
- month_of_diagnosis_Mar   736
- month_of_diagnosis_May   737
- month_of_diagnosis_Nov   738
- month_of_diagnosis_Oct   739
- month_of_diagnosis_Sep   740
- number_of_primaries   741
- race_ethnicity_Amerian Indian, Aleutian, Alaskan Native or Eskimo   742
- race_ethnicity_Asian Indian   743
- race_ethnicity_Asian Indian or Pakistani   744
- race_ethnicity_Black   745
- race_ethnicity_Chinese   746
- race_ethnicity_Fiji Islander   747
- race_ethnicity_Filipino   748
- race_ethnicity_Guamanian   749
- race_ethnicity_Hawaiian   750
- race_ethnicity_Hmong   751
- race_ethnicity_Japanese   752
- race_ethnicity_Kampuchean   753
- race_ethnicity_Korean   754
- race_ethnicity_Laotian   755
- race_ethnicity_Melanesian   756
- race_ethnicity_Micronesian   757
- race_ethnicity_New Guinean   758

- race_ethnicity_Other   759
- race_ethnicity_Other Asian   760
- race_ethnicity_Pacific Islander   761
- race_ethnicity_Pakistani   762
- race_ethnicity_Polynesian   763
- race_ethnicity_Samoan   764
- race_ethnicity_Thai   765
- race_ethnicity_Tongan   766
- race_ethnicity_Unknown   767
- race_ethnicity_Vietnamese   768
- race_ethnicity_White   769
- seer_historic_stage_a_Distant   770
- seer_historic_stage_a_In situ   771
- seer_historic_stage_a_Localized   772
- seer_historic_stage_a_Regional   773
- seer_historic_stage_a_Unstaged   774
- sex_Female   775
- spanish_hispanic_origin_Cuban   776
- spanish_hispanic_origin_Dominican Republic   777
- spanish_hispanic_origin_Mexican   778
- spanish_hispanic_origin_Non-Spanish/Non-hispanic   779
- spanish_hispanic_origin_Other specified Spanish/Hispanic origin (excludes   780
  Dominican Repuclic)   781
- spanish_hispanic_origin_Puerto Rican   782
- spanish_hispanic_origin_South or Central American (except Brazil)   783
- spanish_hispanic_origin_Spanish surname only   784
- spanish_hispanic_origin_Spanish, NOS; Hispanic, NOS; Latino, NOS   785
- spanish_hispanic_origin_Uknown whether Spanish/Hispanic or not   786
- year_of_birth   787
- year_of_diagnosis   788
- month   789

and `newtarget` is the target variable, indicating whether or not the subject died in   790
month given by the value of the `month` variable.   791

Lung Cancer Feature Selection   792

The feature set used as input into both the Random Forest and Neural Network models,   793
after the transformation described in section Transformation of Censored Data for   794
Machine Learning is given below and also available in full detail in the file   795
`NewPatientLungML.html`.   796

- cs_tumor_size   797
- elevation   798
- grade_cell type not determined   799
- grade_moderately differentiated   800
- grade_poorly differentiated   801
- grade_undifferentiated; anaplastic   802
- grade_well differentiated   803
- histology_recode_broad_groupings_acinar cell neoplasms   804
- histology_recode_broad_groupings_adenomas and adenocarcinomas 805
- histology_recode_broad_groupings_blood vessel tumors   806
- histology_recode_broad_groupings_complex epithelial neoplasms 807

- histology_recode_broad_groupings_complex mixed and stromal neoplasms   808
- histology_recode_broad_groupings_cystic, mucinous and serous neoplasms   809
- histology_recode_broad_groupings_ductal and lobular neoplasms   810
- histology_recode_broad_groupings_epithelial neoplasms, NOS   811
- histology_recode_broad_groupings_fibroepithelial neoplasms   812
- histology_recode_broad_groupings_fibromatuos neoplasms   813
- histology_recode_broad_groupings_germ cell neoplasms   814
- histology_recode_broad_groupings_gliomas   815
- histology_recode_broad_groupings_granular cell tumors & alveolar soft part   816
  sarcomas   817
- histology_recode_broad_groupings_lipomatous neplasms   818
- histology_recode_broad_groupings_miscellaneous bone tumors   819
- histology_recode_broad_groupings_miscellaneous tumors   820
- histology_recode_broad_groupings_mucoepidermoid neoplasms   821
- histology_recode_broad_groupings_myomatous neoplasms   822
- histology_recode_broad_groupings_myxomatous neoplasms   823
- histology_recode_broad_groupings_nerve sheath tumors   824
- histology_recode_broad_groupings_neuroepitheliomatous neoplasms   825
- histology_recode_broad_groupings_nevi and melanomas   826
- histology_recode_broad_groupings_osseous and chondromatous neoplasms   827
- histology_recode_broad_groupings_paragangliomas and glumus tumors   828
- histology_recode_broad_groupings_soft tissue tumors and sarcomas, NOS   829
- histology_recode_broad_groupings_squamous cell neoplasms   830
- histology_recode_broad_groupings_synovial-like neoplasms   831
- histology_recode_broad_groupings_thymic epithelial neoplasms   832
- histology_recode_broad_groupings_transistional cell papillomas and carcinomas 833
- histology_recode_broad_groupings_trophoblastic neoplasms   834
- histology_recode_broad_groupings_unspecified neoplasms   835
- lat   836
- laterality_Bilateral involvement, lateral origin unknown; stated to be single 837
  primary   838
- laterality_Left: origin of primary   839
- laterality_Not a paired site   840
- laterality_Only one side involved, right or left origin unspecified   841
- laterality_Paired site, but no information concerning laterality; midline tumor 842
- laterality_Right: origin of primary   843
- lng   844
- marital_status_at_dx_Divorced   845
- marital_status_at_dx_Married (including common law)   846
- marital_status_at_dx_Separated   847
- marital_status_at_dx_Single (never married)   848
- marital_status_at_dx_Unknown   849
- marital_status_at_dx_Unmarried or domestic partner   850
- marital_status_at_dx_Widowed   851
- month_of_diagnosis_Apr   852
- month_of_diagnosis_Aug   853
- month_of_diagnosis_Dec   854
- month_of_diagnosis_Feb   855
- month_of_diagnosis_Jan   856
- month_of_diagnosis_Jul   857
- month_of_diagnosis_Jun   858
- month_of_diagnosis_Mar   859

- month_of_diagnosis_May   860
- month_of_diagnosis_Nov   861
- month_of_diagnosis_Oct   862
- month_of_diagnosis_Sep   863
- number_of_primaries   864
- race_ethnicity_Amerian Indian, Aleutian, Alaskan Native or Eskimo   865
- race_ethnicity_Asian Indian   866
- race_ethnicity_Asian Indian or Pakistani   867
- race_ethnicity_Black   868
- race_ethnicity_Chamorran   869
- race_ethnicity_Chinese   870
- race_ethnicity_Fiji Islander   871
- race_ethnicity_Filipino   872
- race_ethnicity_Guamanian   873
- race_ethnicity_Hawaiian   874
- race_ethnicity_Hmong   875
- race_ethnicity_Japanese   876
- race_ethnicity_Kampuchean   877
- race_ethnicity_Korean   878
- race_ethnicity_Laotian   879
- race_ethnicity_Melanesian   880
- race_ethnicity_Micronesian   881
- race_ethnicity_New Guinean   882
- race_ethnicity_Other   883
- race_ethnicity_Other Asian   884
- race_ethnicity_Pacific Islander   885
- race_ethnicity_Pakistani   886
- race_ethnicity_Polynesian   887
- race_ethnicity_Samoan   888
- race_ethnicity_Thai   889
- race_ethnicity_Tongan   890
- race_ethnicity_Unknown   891
- race_ethnicity_Vietnamese   892
- race_ethnicity_White   893
- seer_historic_stage_a_Distant   894
- seer_historic_stage_a_In situ   895
- seer_historic_stage_a_Localized   896
- seer_historic_stage_a_Regional   897
- seer_historic_stage_a_Unstaged   898
- sex_Female   899
- spanish_hispanic_origin_Cuban   900
- spanish_hispanic_origin_Dominican Republic   901
- spanish_hispanic_origin_Mexican   902
- spanish_hispanic_origin_Non-Spanish/Non-hispanic   903
- spanish_hispanic_origin_Other specified Spanish/Hispanic origin (excludes   904
  Dominican Repuclic)   905
- spanish_hispanic_origin_Puerto Rican   906
- spanish_hispanic_origin_South or Central American (except Brazil)   907
- spanish_hispanic_origin_Spanish surname only   908
- spanish_hispanic_origin_Spanish, NOS; Hispanic, NOS; Latino, NOS   909
- spanish_hispanic_origin_Uknown whether Spanish/Hispanic or not   910
- year_of_birth   911

- year_of_diagnosis   912
- month   913

Breast Cancer Feature Selection   914

The feature set used as input into both the Random Forest and Neural Network models,   915
after the transformation described in section Transformation of Censored Data for   916
Machine Learning is given below and also available in full detail in the file   917
`NewPatientBreastML.html` .   918

- cs_tumor_size   919
- elevation   920
- grade_moderately differentiated   921
- grade_poorly differentiated   922
- grade_ndifferentiated; anaplastic   923
- grade_well differentiated   924
- histology_recode_broad_groupings_adenomas and adenocarcinomas   925
- histology_recode_broad_groupings_adnexal and skin appendage neoplasms   926
- histology_recode_broad_groupings_basal cell neoplasms   927
- histology_recode_broad_groupings_complex epithelial neoplasms   928
- histology_recode_broad_groupings_cystic, mucinous and serous neoplasms   929
- histology_recode_broad_groupings_ductal and lobular neoplasms   930
- histology_recode_broad_groupings_epithelial neoplasms, NOS   931
- histology_recode_broad_groupings_nerve sheath tumors   932
- histology_recode_broad_groupings_unspecified neoplasms   933
- lat   934
- laterality_Bilateral involvement, lateral origin unknown; stated to be single   935
  primary   936
- laterality_Paired site, but no information concerning laterality; midline tumor   937
- laterality_Right: origin of primary   938
- lng   939
- marital_stats_at_dx_Divorced   940
- marital_stats_at_dx_Married (inclding common law)   941
- marital_stats_at_dx_Separated   942
- marital_stats_at_dx_Single (never married)   943
- marital_stats_at_dx_Unknown   944
- marital_stats_at_dx_Unmarried or domestic partner   945
- marital_stats_at_dx_Widowed   946
- month_of_diagnosis_Apr   947
- month_of_diagnosis_Aug   948
- month_of_diagnosis_Dec   949
- month_of_diagnosis_Feb   950
- month_of_diagnosis_Jan   951
- month_of_diagnosis_Jul   952
- month_of_diagnosis_Jun   953
- month_of_diagnosis_Mar   954
- month_of_diagnosis_May   955
- month_of_diagnosis_Nov   956
- month_of_diagnosis_Oct   957
- month_of_diagnosis_Sep   958
- race_ethnicity_Amerian Indian, Aletian, Alaskan Native or Eskimo   959
- race_ethnicity_Asian Indian   960
- race_ethnicity_Black   961

- 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.

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)                              1011
for t = 1 to (the last month where h has any data):                1012
    // h knows p(T = t | T >= t, X), we call this p_cur            1013
    p_cur = h's prediction for (X, t)                              1014
    append (p_so_far * p_cur) to A                                 1015
    p_so_far *= (1 - p_cur)                                        1016
                                                                   1017
```

Breast Random Forest Model Hyperparameters   1018

```
f = RandomForestClassifier(n_estimators=20,min_samples_split=3,    1019
                           max_depth = 15,                         1020
                           max_features = .8,                      1021
                           n_jobs=5,verbose=2,random_state=33)      1022
```

Colon Random Forest Model Hyperparameters   1023

```
rf = RandomForestClassifier(n_estimators=25,min_samples_split=3,   1024
                            max_depth = 10,                        1025
                            max_features = .5,                     1026
                            n_jobs=5,verbose=2,random_state=3)      1027
```

Lung Random Forest Model Hyperparameters   1028

```
rf = RandomForestClassifier(n_estimators=25,min_samples_split=3,   1029
                            max_depth = 11,                        1030
                            max_features = .8,                     1031
                            n_jobs=5,verbose=2,random_state=3)      1032
```

Breast Neural Network Model Architecture   1033

The archictecture of the Keras multilayer perceptron neural network model trained on   1034
the breast cancer data is given explicitly below: 1035

```
modelbreast = Sequential()                                                1036
modelbreast.add(Dense(114, input_shape=(66,) ,init='normal'))             1037
modelbreast.add(Activation('relu'))                                       1038
modelbreast.add(Dropout(0.05))                                            1039
modelbreast.add(Dense(50, init='normal'))                                 1040
modelbreast.add(Activation('relu'))                                       1041
modelbreast.add(Dropout(0.05))                                            1042
                                                                          1043
modelbreast.add(Dense(36, init='normal'))                                 1044
modelbreast.add(Activation('relu'))                                       1045
modelbreast.add(Dropout(0.05))                                            1046
                                                                          1047
modelbreast.add(Dense(2, init='normal'))                                  1048
modelbreast.add(Activation('softmax'))                                    1049
                                                                          1050
rms = RMSprop(lr=0.001)                                                   1051
                                                                          1052
modelbreast.compile(loss='binary_crossentropy',                          1053
            optimizer=rms, class_mode="binary")                           1054
                                                                          1055
```

and trained with a batch size of 1500 for 200 epochs. 1056

Colon Cancer Neural Network Model Architecture 1057

The archictecture of the Keras multilayer perceptron neural network model trained on 1058
the colon cancer data is given explicitly below: 1059

1060
1061

```
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")
```
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082

and trained with a batch size of 1500 for 200 epochs. 1083

Lung Cancer Neural Network Model Architecture 1084

The archictecture of the Keras multilayer perceptron neural network model trained on 1085
the lung cancer data is given explicitly below: 1086

1087

```
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'))
```
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102

```
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.

S1 Video

Bold the first sentence. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S1 Text

Lorem Ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S1 Fig

Lorem Ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S2 Fig

Lorem Ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S1 Table

Lorem Ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

Acknowledgments

References

1. Sebastian Raschka. Python Machine Learning Essentials. Packt Publishing; 2015.

2. Cam Davidson-Pilon. Quickstart – lifelines 0.8.0.1 documentation; 2016 (accessed 14 Jan 2016).
   `http://lifelines.readthedocs.org/en/latest/Quickstart.html`.

3. 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`.

4. National Cancer Institute, the Surveillance, Epidemiology, and End Results Program. About the SEER Program - SEER; 2016 (accessed 14 Jan 2016). `http://seer.cancer.gov/about`.

5. 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.

6. 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.

7. 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`.

8. 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`.

9. 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`.

10. 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`.

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

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

13. 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`.

14. 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`.

15. 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`.

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

17. Kaplan EL, Meier P. Nonparametric Estimation from Incomplete Observations. Journal of the American Statistical Association. 1958;53(282):457–481. Cited By 34216. Available from: `http://www.scopus.com/inward/record.url?eid=2-s2.0-33845382806&partnerID=40&md5=1b3d05fab93fcd0e37f176c799b3cff6`.

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

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

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

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

22. 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`.

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

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

25. 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`.

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

27. Armin Ronacher. Welcome — Flask (A Python Microframework), howpublished=`http://flask.pocoo.org/`, year="2014 (accessed 29 Jan 2016)";.

28. 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`.