Prepared for submission to JHEP

# Machine Learning for Survival Analysis: A New Approach

**D. Dooling P. Green A. Kim D. Scroggin L. Stevens J. Webster**

*Innovative Oncology Business Solutions,*
*4901 Lang Ave NE,*
*Albuquerque, NM 87109, USA*

*E-mail:* ddooling@innovativeobs.com, pgreen@innovativeobs.com, akim@innovativeobs.com, dscroggin@innovativeobs.com, lstevens@innovativeobs.com, jwebster@innovativeobs.com

Abstract: We have applied a little-known data transformation on 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.

## Contents

## 1 Introduction and Background

Extracting actionable information from data is changing the fabric of modern business. A class of techniques that transforms data into actionable information goes by the name of Machine Learning [1]. Machine Learning has recently become a popular method to answer questions and solve problems that are too complex to solve via traditional methods. The Surveillance, Epidemiolgy, and End Results (SEER) Program of the National Cancer Institute (NCI) has been collecting data because intuitively researchers feel confident that this data is capturing information that has buried within it useful information in the form of relationships between the types of data collected (demographic as well as staging information) and the survival outcomes. Though this relationship evades capture by traditional methods, it is possible to surface it with the two machine learning techniques known as **Random Forests** and **Neural Networks**. These two methods produce very similar results when applied to the SEER dataset, and are based on two almost diametrically opposed learning philosophies, which lends confidence in the validity of the results.

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

Quoting directly from the SEER website [2]:

> The SEER program registries routinely collect data on patient demographics, primary tumor site, tumor morphology and stage at diagnosis, first course of treatment, and follow-up for vital status. This program is the only comprehensive source of population-based information in the United States that includes stage of cancer at the time of diagnosis and patient survival data. The mortality data reported by SEER are provided by the National Center for Health Statistics. The population data used in calculating cancer rates is obtained periodically from the Census Bureau. 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 and that is shared by many datasets in the medical field goes by the name of "censored data." 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's Proportional Hazard models [3]. 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. In addition, the Cox Proportional Hazard models are not able to capture

the nonlinear relationships between the given data fields that go into making predictions; they can only capture the first-order linear relationships.

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." The authors of [4] 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. The authors of [5] 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 trained models.

To overcome these limitations of the traditional methods, IOBS has applied a little-known technique to transform the SEER data to make it amenable to more powerful machine learning methods. 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 simply so as 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. Details of this transformation can be found in this blog post [6].

## 2 Methodology

### 2.1 Data acquisition

We used the publically available 1973-2012 SEER incidence data files corresponding to colon, breast and lung cancer contained in the following list. SEER requires that researchers submit a request for the data, which includes an agreement form. Detailed documentation explaining the contents of both the incidence data files used in this study as well as a data dictionary for the 1973-2012 SEER incidence data files are available without the need to register or submit a data request [7].

- 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

| 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.

- incidence\yr2005.lo_2nd_half\BREAST.txt
- incidence\yr2005.lo_2nd_half\RESPIR.txt

### 2.2 Data preparation and preprocessing

A great deal of data munging is necessary before using these SEER incidence files as input into machine learning algorithms. A preprocessing step common to each of three cancer types studied involves the `STATE-COUNTY RECODE`. 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. This particular field illustrates an important feature of machine learning, that 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 of the real numbers. 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 [8].

As a simple example of how to correctly treat categorical variables in a mchine learning context, consider the SEER variable `SEX`. This variable is encoded 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 [9]. 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,

| Sex | | sex_Male | sex_Female | | sex_Female |
|---|---|---|---|---|---|
| 1 | $\longrightarrow$ | 1 | 0 | $\longrightarrow$ | 0 |

(2.1)

and

| Sex | | sex_Male | sex_Female | | sex_Female |
|---|---|---|---|---|---|
| 2 | $\longrightarrow$ | 0 | 1 | $\longrightarrow$ | 1 |

(2.2)

The procedure outlined in Equations (2.1, 2.2) 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

| 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 |

**Table 2**. Example of the transformation of `STATE-COUNTY RECODE` to `elevation` , `lat` , and `lng` .

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 data 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 [8]. 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 an ordinal variable. There is an ordering among the (state_county) columns, name 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 would 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` . Using the publically available dafafile from the United States Census Bureau [10] to construct query strings like the values of the `address` field in Table ( 2), it is possible to then programmatically query the Google Maps Geocoding API for the latitude and longitude [11], and the Google Maps Elevation API for the corresponding elevation [12]. 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 covereed in the SEER registries. The full lookup table analogous to Table 2 is available from a GitHub repository containing supplemental information for this study [13].

### 2.3 Colon Cancer Data

In this section we describe the data processing steps that were specific to the colon cancer model development. 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 following categorical features were one-hot encoded as described in section 2.2:

| Column | Filter |
|---|---|
| SEQUENCE NUMBER-CENTRAL | $\neq$ "Unspecified" |
| AGE AT DIAGNOSIS | $\neq$ "Unknown age" |
| BIRTHDATE-YEAR | $\neq$ "Unknown year of birth" |
| YEAR OF DIAGNOSIS | $\geq 2004$ |
| 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 3**. Filters applied to the Colon Cancer data.

- 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 .

The STATE-COUNTY RECODE variable was dropped and replaced with the elevation , lat , and lng variables as illustrated in Table 2.

With just the above data preparation, it is possible to construct traditional Kaplan-Meier estimates of the survival curves for the colon cancer population represented by this subset of the data. After the above one-hot encoding procedure, the new variable vital_status_recode_Dead indicates that the patient is deceased if this variable = 1, or else that the patient's record is right-censored if this variable = 0. SURVIVAL MONTHS and vital_status_recode_Dead are all that is needed to construct the Kaplan-Meier estimate shown in Figure (1).

### 2.4 Lung Cancer Data

In this section we describe the data processing steps that were specific to the lung cancer model development. The four RESPIR.txt files were imported into a pandas DataFrame object. This data was then filtered according to the conditions in Table 4. The same list of categorical features as in the colon cancer case were then one-hot encoded.

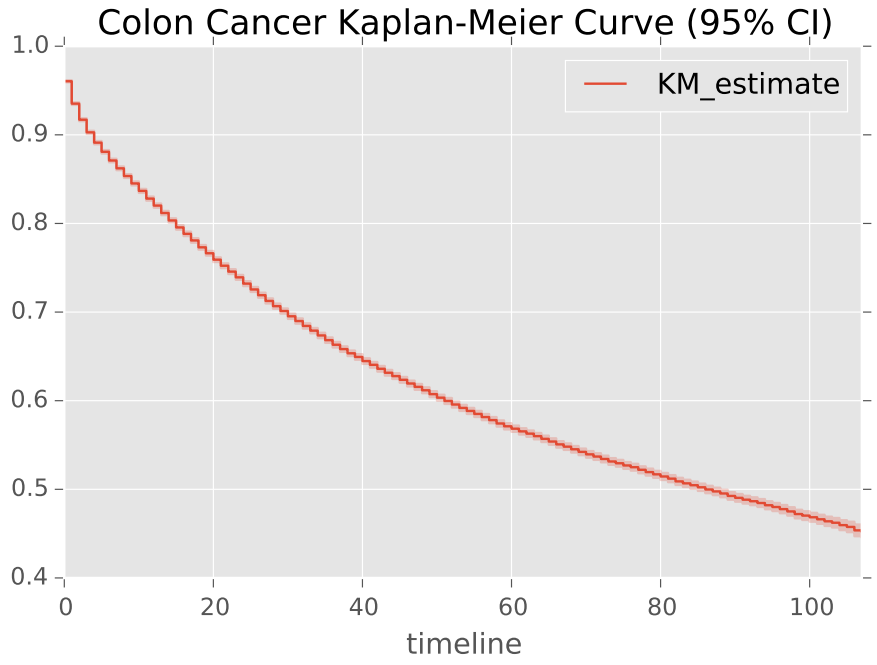


**Figure 1**. Traditional Kaplan-Meier estimate of the survival curve for all colon cancer patients. Fitted with 113072 observations, 71804 censored.

| Column | Filter |
|---|---|
| SEQUENCE NUMBER-CENTRAL | $\neq$ "Unspecified" |
| AGE AT DIAGNOSIS | $\neq$ "Unknown age" |
| BIRTHDATE-YEAR | $\neq$ "Unknown year of birth" |
| YEAR OF DIAGNOSIS | $\geq 2004$ |
| 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 4**. Filters applied to the Lung Cancer data.

With just the above data preparation, it is possible to construct traditional Kaplan-Meier estimates of the survival curves for the colon cancer population represented by this subset of the data. After the above one-hot encoding procedure, the new variable `vital_status_recode_Dead` indicates that the patient is deceased if this variable = 1, or else that the patient's record is right-censored if this variable = 0. `SURVIVAL MONTHS` and

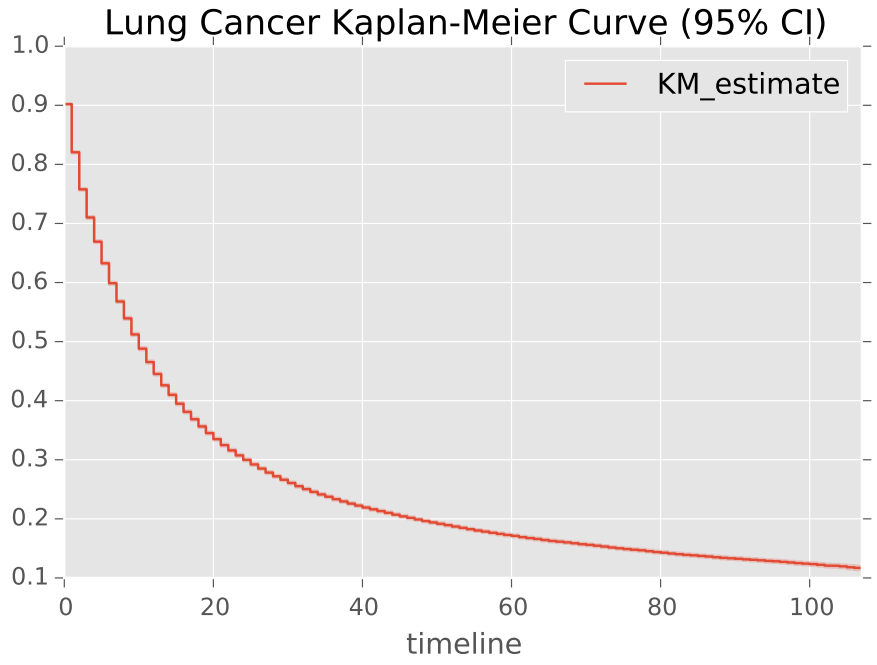


**Figure 2**. Traditional Kaplan-Meier estimate of the survival curve for all lung cancer patients. Fitted with 177089 observatins, 47409 censored.

`vital_status_recode_Dead` are all that is needed to construct the Kaplan-Meier estimate shown in Figure (2).

### 2.5 Breast Cancer Data

In this section we describe the data processing steps that were specific to the lung cancer model development. The four BREAST.txt files were imported into a pandas DataFrame object. This data was then filtered according to the conditions in Table 5. The same list of categorical features as in the colon cancer case were then one-hot encoded.

With just the above data preparation, it is possible to construct traditional Kaplan-Meier estimates of the survival curves for the colon cancer population represented by this subset of the data. After the above one-hot encoding procedure, the new variable `vital_status_recode_Dead` indicates that the patient is deceased if this variable = 1, or else that the patient's record is right-censored if this variable = 0. `SURVIVAL MONTHS` and `vital_status_recode_Dead` are all that is needed to construct the Kaplan-Meier estimate shown in Figure (3).

Before applying machine learning models trained with these data sets, we review in section (3) 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 set of features and target variables listed in sections (A.1, A.2, A.3).

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

**Table 5**. Filters applied to the Breast Cancer data.

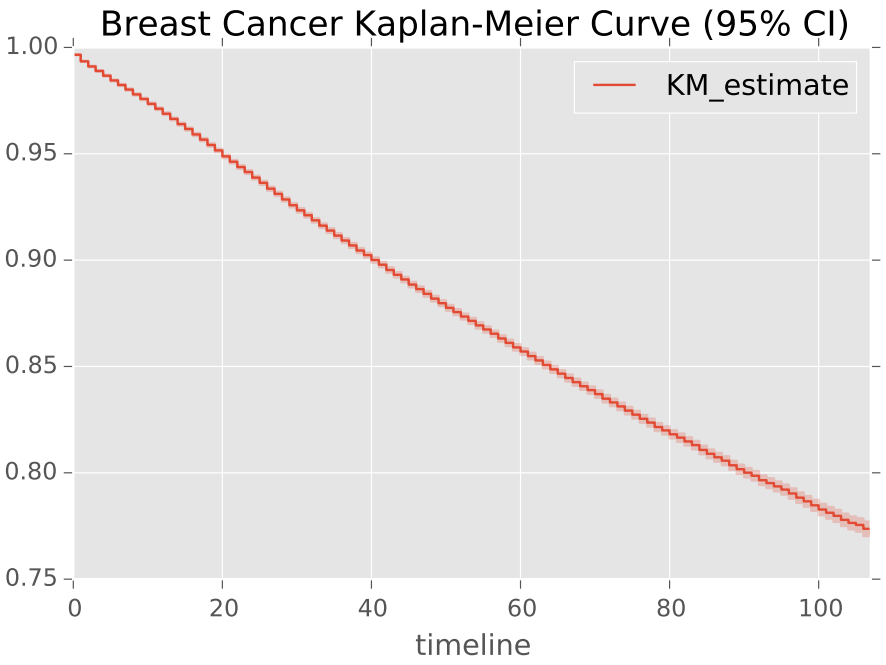


**Figure 3**. Traditional Kaplan-Meier estimate of the survival curve for all breast cancer patients. Fitted with 329949 observatins, 292279 censored.

## 3 Machine Learning Survival Analysis with Censored Data

The above Kaplan-Meier estimates of the survival curves for colon (Figure (1), lung (Figure (2), and breast cancer (Figure (3) are constructed from the full population of cancer patients in the respective datasets. An unsatisfactory consequence is that these estimates are highly course-grained, and not very meaningful to an indivual. Patients with very disparate characteristics are given the same prognosis by these Kaplan-Meier survival curve

estimates. Therefore it is desirable to find robust predictors for survival curves of individual where the input is an individual record as opposed to a population.

### 3.1 Survival Analysis

To understand survival analysis, you first have to understand survival data - that survival times are *intervals* between certain kinds of events, that these intervals are often affected by a peculiar kind of "partial missingness" called *censoring*, and that censored data must be analyzed in a special way to avoid biased estimates and incorrect conclusions.

In the case of the SEER data, the starting point of the time interval is the diagnosis date. Even though survival times are continuous or nearly continuous numerial quantities, they're never almost never normally distributed. If non-normality were the only problem with survival data, you would be able to summarize survival times as medians and centiles instead of means and standard deviations, and you could compare survival between groups with nonparametric Mann-Whitney and Kruksal-Wallis testse instead of t tests and ANOVAs. But time-to-event data is susceptible to a special situation called *censoring*, which the usual parametric and non-parametric methods cannot handle. Therefore special methods have been developed to analyze censored data properly.

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

Statisticians have developed some traditional techniques to utilize the partial information contained in censored observations: the life-table method and the Kaplan-Meier method. To understand these methods, you need to understand two fundamental concepts: - *hazard* and *survival*:

- **The hazard rate** is the probability of dying in the next small interval of time, assuming that the subject is alive right now.
- **The survival rate** is the probability of living for a certain amount of time after some starting point.

The first task when analyzing survival data is usually to describe how the hazard and survival rates vary with time. Here are two ways *not* to handle censored survival data:

- You shouldn't excude subjects with a censored surivival time from any survival analysis.
- You shouldn't *impute* (replace) the censored (last-seen) date with some reasonable substitute value. One commonly used imputation scheme is to replace a missing value with the last observed value for that subject (called *last observation carried forward*, or LOCF imputation).

These techniques for dealing with missing data don't work for censored data. If you simply exclude all subjects with censored death dates from your analysis, you may be left with too few analyzable subjects, which weakens (underpowers) youe study. Worse, it will also bias your results in subtle and unpredictable ways. The problem is that a censored observation time isn't really missing. If you know that a person was last seen alive three years after treatment, you have partial information for that patient. You don't know exactly what the patient's true survival time is, but you do know that it's at least three years.

To estimate survival and hazard rates in a population from a set of observed survival times, some of which are censored, you must combine the information from censored and uncensored observations properly. You have to think of the process in terms of a series of small slices of time, and think of the probability of making it through each time slice, assuming that the subject is alive at the start of that slice. The cumulative survival probability can then be obtained by successively multiplying all these individual time-slice survival probabities together. For example, to survive three years, first the subject has to make it through Year 1, then she has to make it through Year 2, and then she has to make it through Year 3. The probability of making it through all three years is the product of the probabilities of making it through Year 1, Year 2, and Year 3. These calculations can be laid out very systematically in a *life table*, sometimes called an *actuarial life table* because of its early use by insurance comparies. The calculations involce only addition, subtraction, multiplication, and division and are simple enough to do by hand.

To create a life table from your survival-time data, first break the entire range of survival times into convenient time slices (months, quarters, or years, depending on the time scale of the event you're studying). You should try to have at least five slices, otherwise, your survival and hazard estimates will be too coarse to show any useful features. Having very fine slices doesn't hurt the calculations, although the table will have more rows and may become unweildly. Next, count how many people died during each slice and how many were *censored* (that is, last seen alive during that slice, either because they became lost to follow-up or were still alive at the end of the study). For the survival times shown in Table (6), a natural choice would be to use seven one-year time slices.

Next, count how many people died during each slice and how many were *censored* (that is, last seen alive during that slice, either because they became lost to follow-up or were still alive at the end of the study). From Table (6) you see that

- During the first year after surgery, one subjects died (#0) and one subject was censored ( #4)
- During the second year, nothing happened (no deaths, no censoring)
- During the third year, two subjects died (#8 and #3)
- During the fourth year, one subject was censored (#6)
- During the fifth year, one subject died (#5)
- During the sixth year, one subject died (#9) one subject was censored (#7)
- During the seventh year, one subjectdied (#1) and one was censored (#2)

To construct a lifetable, one proceeds as follows:

| | 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 6**. Example data to illustate traditional Survival Analsyis.

| | Died | Censored | Alive at Start | At Risk | Prob of Dying | Prob of Surviving | Cum Survival |
|---|---|---|---|---|---|---|---|
| 0-1 yr | 1 | 1 | 10 | 9.5 | 0.105263 | 0.894737 | 0.894737 |
| 1-2 yr | 0 | 0 | 8 | 8.0 | 0.000000 | 1.000000 | 0.894737 |
| 2-3 yr | 2 | 0 | 8 | 8.0 | 0.250000 | 0.750000 | 0.671053 |
| 3-4 yr | 0 | 1 | 6 | 5.5 | 0.000000 | 1.000000 | 0.671053 |
| 4-5 yr | 1 | 0 | 5 | 5.0 | 0.200000 | 0.800000 | 0.536842 |
| 5-6 yr | 1 | 1 | 4 | 3.5 | 0.285714 | 0.714286 | 0.383459 |
| 6-7 yr | 1 | 1 | 2 | 1.5 | 0.666667 | 0.333333 | 0.127820 |

**Table 7**. Lifetable corresponding to the example data in Table (6).

- Enter the total number of subjects alive at the start into column **Alive at Start**, in the **0-1 yr** row
- Enter the counts of people who died within each time slice into column **Died**
- Enter the counts of people who were censored during each time slice into **Censored**
- The column **At Risk** shows the number of subjects known to be alive at the start of each year after surgery. This is equal to the number of subjects alive at the start of the preceding year minus the number of subjects who died (**Died**) or were censored (**Censored**) during the preceding year.
- The Column **At Risk** shows the number of subjects "at risk for dying" during each year. You may guess that this is the number of people alive at the start of the interval, but there's one minor correction. ( COMPLETELY DISAGREE WITH THIS; A WEAKNESS OF THE METHOD) If any people were censured during that year, then they weren't really "available to die" (to use an awful expression) for the entire year. If you don't know exactly when, during that year, they became censored, then it's reasonable to "split the difference" and consider them at risk for aonly half

the year. So the number at risk can be estimated as the number alive at the start of the year, minus one-half of the number who became censored during that year. (ONLY MAKES SENSE FOR LOST TO FOLLOW UP).

- The column **Prob of Dying** shows the probability of dying during each interval, assuming the subject has survived up to the start of that interval. This is simply the number of people who died divided by the number of people at risk during each interval.
- The column **Prob of Surviving** shows the probability of surviving during each interval, assuming the subject has survived up to the start of that interval. Surviving means not dying, so the probability of surviving is simply 1 - the probability of dying.
- The column **Cum Survival** shows the cumulative probability of surviving from the time of the operation all the way through the end of this time slice. To survive from the time of the operation through the end of any given year (year $N$), the subject must survive each of the years from Year 1 through Year $N$. Because surviving each year is an independent accomplishment, the probability of surviving all $N$ of the years is the product of the individual years' probabilities.

The sample hazard and survival values obtained from a life table are only sample estimates (in this example, at 1-year time slices) of the true population hazard and survival functions. The hazard rate obtained from a life table is equal to the probability of dying during each time slice (column **Prob of Dying**) divided by the width of the slice, so the hazard rate for the first year would be expressed as .105 per year, or 10.5 percent per year. The cumulative survival probability in column **Cum Survival**, is the probability of surviving from the operation date through to the end of the interval. It has no units, and it can be expressed as a fraction or as a percentage.

Using very narrow time slices doesn't hurt life-table calculations. In fact, you can define slices so narrow that each subject's survival time falls within its own private little slice. With $N$ subjects, $N$ rows would have one subject each; all the rest of the rows would be empty. And because empty rows don't affect the life-table calculations, you can delete them entirely, leaving a table with only $N$ rows, one for each subject. (If you happen to have two or more subjects with exactly the same survival or censoring time, it's okay to put each of the subjects in a separate row). The life-table calculations work fine with only one subject per row and produce what's called *Kaplan-Meier (K-M) survival estimates*. You can think of the K-M method as a very fine-grained life table or a life table as a grouped K-M calculation.

The Kaplan-Meier survival estimate corresponding to the data given in Table (6) is shown in Table (8).

### 3.2 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 described in section 3.1. The full details of this transformation, and a large inspiration for this study, can be flound in this blog post [6].

| | Censored Status | Survival Times | Alive at Start | Prob of Dying | Prob of Surv | Cum Survival |
|---|---|---|---|---|---|---|
| 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 |

**Table 8**. Kaplan-Meier table corresponding to the example data in Table (6).

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

$$\hat{S}(t) = \prod_{t_i<t} \frac{n_i - d_i}{n_i} \tag{3.1}$$

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 priot to time $t$. Equation 3.1 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 for all of the individual records in the data set.

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

$$h(\mathbf{X}, t) = P(Y = t | Y \geq t, \mathbf{X}), \tag{3.2}$$

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 $T < Y$.

Treating $T$ is 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

| | 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 an uncensored 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 a censored record in the untransformed 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 |

**Table 11**. Example of four columns in an uncensored 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 |

**Table 12**. Example of four columns in a censored record in the transformed dataset.

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 B.

Explicit examples will help make this transformation clear. The untransformed datapoint represented Table (9) is transformed to the multiple records shown in Table (11). All uncensored data is transformed in this way. All censored data is similarly transformed. The untransformed datapoint represented Table (10) is transformed to the multiple records shown in Table (12).

One obvious side effect of this transformation is that it explodes the data size. For

this study, the original, untransformed colon cancer DataFrame has shape $(113072, 106)$, and the total transformed colon cancer DataFrame has shape $(4165251, 106)$. Similary, the original, untransformed lung cancer DataFrame has shape $(177089, 118)$, and the total transformed lung cancer DataFrame has shape $(3079931, 118)$. The biggest explosion in data size occured with the breast cancer data. The original, untransformed breast cancer DataFrame has shape $(329949, 70)$, and the total transformed breast cancer DataFrame has shape $(15085711, 70)$. 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.

## 4 Prediction Models

With the datasets transformed as described in section (3.2), we are now able to split them into training and testing sets in the usual manner. The classifier models described in this section are learning the hazard function: given all of the data given in sections (A.2, A.1, A.3), which includes the field `months` (the months after diagnosis), the models predict the target variable `newtarget` , which represents the probability of dying in that month, given that the patient represented by the record has survived up to that month. 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 learned and predicted hazard functions. From the resulting hazard functions for each unique patient, we can construct the resulting survival functions as presented in section (B) and explicitly given in python code in the notebooks at the github repository containing supplemental material for this study [13].

In order to evaluate the performance of the learned models, we first construct three binary classifiers corresponding to whether or not a subject survived 6, 12, or 60 months after diagnosis. This is done by looping over all distinct patient indices in the test set, prediciting the full survival function, and capturing the values corresonding to 6, 12, and 60 months.

It is necessary to apply some Boolean filters to the data in order to correctly assess the resulting classifiers, again because of censoring. For example, 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 conditiions 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.

### 4.1 Training and Test Partitions

After performing the data transformation adumbrated in section (3.2), it is necessarily 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 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 $\approx 396$ for the breast cancer data, $\approx 99$ for the colon cancer data, and $\approx 22.75$ for the lung cancer data.

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 *probablity* of `newtarget` being '0' or '1'., and thus learning the hazard function.

Finally, we emphasize the crucial point that 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 (9, 10, 11, 12). 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.

### 4.2 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.

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

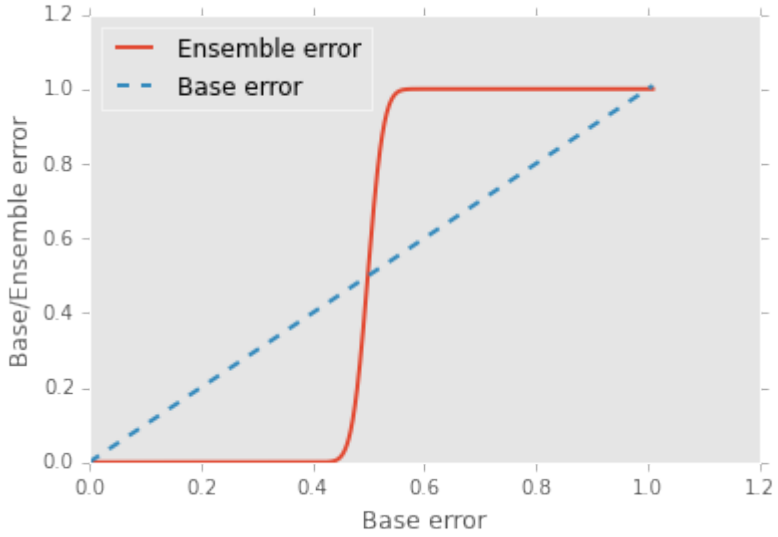


**Figure 4**. Illustration of ensemble methods showing how a collection of base learners with poor accuracy can combine to produce an accurate ensemble learner.

IOBS has chosen to use the Python scikit-learn implemenation of the Random Forest machine learning classifier [14]. Random Forests are frequent winners of the Kaggle machine learning competitions [15]. The model parameters for each cancer type are given in sections (C.1, C.2, C.3).

### 4.3 MLP Neural Networks

Neural networks are a biologically-inspired programming paradigm that enable computers to learn from observational data [16]. Deep learning can be understood as a set of algorithms that were developed to train **artificial neural networks** with many layers most efficiently. 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

| Model | 6 Months AUC | 12 Months AUC | 60 Months AUC |
|---|---|---|---|
| Breast RF | .846 | .885 | .844 |
| Breast NN | .855 | .867 | .836 |
| Colon RF | .804 | .806 | .828 |
| Colon NN | .797 | .804 | .841 |
| Lung RF | .772 | .796 | .874 |
| Lung NN | .765 | .796 | .875 |

**Table 13**. AUC values for the Random Forest and Neural Networks model binary classifiers derived from the full survival curve predictions; see text for details.

networks and deep learning research. As of today, complex neural networks powered by deep learning algorithms are considered as 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 [17].

IOBS has chosen to use the 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) [18]. 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 sections (C.4, C.5, C.6). Training a neural network and choosing an appropriate architecture is as much art as science [16], and the search for a good neural network architecture for the lung cancer case was more demanding than for the breast and colon.

## 5 Results

### 5.1 Performance Metrics

The AUC scores for each of the 18 different binary classifiers are listed in Table (13). We emphasize the treatment explained in section (4) concering the correct treatment of the censored test data when evaluating performance metrics. Namely, when computing the AUC for the 12 month survival curve classifiers, we restrict the test data subjects to those that in the untransformed data set that satisfy either of the following mutually exlusive conditions:

- `survival_months` >= 12 AND `vital_status_recode` == 0
- `vivtal_status_recode` == 1

We limit evaluation data to subsets of the data where we know for certain whether or not the subject survived at least 12 months. Similar considerations apply to the 12 and 60 months AUC calculations. The lowest AUC in Table (13) is .765, corresponding to

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

**Table 14**. 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.

the lung neural network model predictions for 6 months survival, while the highest AUC in Table (13) is .885, corresponding to the breast random forest model predictions for 12 months survival.

### 5.2 Model Agreement

TO DO: Check if comorbidites are contributing to the outliers in the agreement boxplots that follow. Could mesh with the prevous work [5].

An additional means of validating the predictions of these models is by comparing their predictions to each other for the same set of input data. Table (14) shows the strong agreement between the random forest and neural network classifiers for each cancer type. Python code showing how the values in Table (14) are computed is available in the files `NewPatientBreastCF.html` , `NewPatientColonCF.html` , and `NewPatientLung.html` in the GitHub repository containing supplemental matierial for this study [13]. This table is computed as follows. For each cancer type (breast,colon, and lung), do the following:

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

This high level of agreement between two models lends confidence to the notion that they have both learned from the training data and are generalizing well. Figures (7, 6, 8) show box plots of the value of the random forest prediction subtracted from the neural network prediction.

## 6 Survival Curve Prediction Apps

CF this guy `http://kmplot.com/analysis/index.php?p=service&cancer=lung`

Below is a list of some web applications developed by IOBS. For each of the cancer types (colon, breast, lung, and prostate), a model has been developed using random forests and one using neural networks. The models were evaluted using the AUC (Area Under

Curve) performance evaluation metric on test data (data not used in the training of the models), typically achieving AUC scores $\approx .8$, as shown in Table (13).[1]

1. breast cancer
   (a) random forest: http://ming-cancer.herokuapp.com/
   (b) neural network: http://breastcancer-neuralnetwork.herokuapp.com/
2. lung cancer
   (a) random forest: http://lung-cancer.herokuapp.com/
   (b) neural network: http://lungnn.herokuapp.com/
3. colon cancer
   (a) random forest: http://colon-cancer.herokuapp.com/
   (b) neural network: http://coloncancernn.herokuapp.com/
4. prostate cancer
   (a) random forest: http://prostate-cancer.herokuapp.com/

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 (15) results in the survival curve depicted in Figure (9); the predicted probablities of living at least 6, 12, and 60 months are .89, .83, and .50, respectively.

| 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 |

**Table 15**. Example input data to the Colon Cancer neural network app.

[1] "AUC | Kaggle," Kaggle Website, https://www.kaggle.com/wiki/AUC, accessed 11 Jan 2016.

Changing the data in Table (15) 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.

## 7 Further Directions

Discussion of causality. A certain Marital status is not a "cause" of a better prognosis; c.f. Simpson's Paradox. Implementation of Judea Pearl's Causality Calculus.

The leap from observation to causality can be hazardous, however, if not analyzed correctly[2]. IOBS is looking into making these conclusions drawn from evidence-based, machine learning models more rigorous by firmly vetting them within the cutting-edge methods of Causality Calculus as pioneered by Judea Pearl.[3]

## A Selected Features

In this Appendix we explicitly list the features chosen for each of the Colon, Breast and Lung cancer predictive models. For each cancer type, the features chosen for the random forest and neural network models were the same, so as to be best able to compare the two models. IPython notebooks explicitly providing all code, as well as html versions of the notebooks, are available from a GitHub repository providing supplemental material for thus study [13].

### A.1 Colon Cancer Feature Selection

The feature set used as input into both the Random Forest and Neural Network models, after the transformation described in section (3.2) is given below and also available in full detail in the file `NewPatientColonML.html` .

- cs_tumor_size
- elevation
- grade_cell type not determined
- grade_moderately differentiated
- grade_poorly differentiated
- grade_undifferentiated; anaplastic
- grade_well differentiated
- histology_recode_broad_groupings_acinar cell neoplasms

[2] "Simpson's Paradox," http://www.intuitor.com/statistics/SimpsonsParadox.html, accessed 11 Jan 2016.

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

- histology_recode_broad_groupings_adenomas and adenocarcinomas
- histology_recode_broad_groupings_blood vessel tumors
- histology_recode_broad_groupings_complex epithelial neoplasms
- histology_recode_broad_groupings_complex mixed and stromal 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_fibromatuos neoplasms
- histology_recode_broad_groupings_germ cell neoplasms
- histology_recode_broad_groupings_lipomatous neplasms
- histology_recode_broad_groupings_miscellaneous bone tumors
- histology_recode_broad_groupings_myomatous neoplasms
- histology_recode_broad_groupings_neuroepitheliomatous neoplasms
- histology_recode_broad_groupings_nevi and melanomas
- 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_transistional cell papillomas and carcinomas
- histology_recode_broad_groupings_unspecified neoplasms
- lat
- 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_Chinese
- race_ethnicity_Fiji Islander
- race_ethnicity_Filipino
- race_ethnicity_Guamanian
- race_ethnicity_Hawaiian
- race_ethnicity_Hmong
- race_ethnicity_Japanese
- race_ethnicity_Kampuchean
- race_ethnicity_Korean
- race_ethnicity_Laotian
- race_ethnicity_Melanesian
- race_ethnicity_Micronesian
- race_ethnicity_New Guinean
- race_ethnicity_Other
- race_ethnicity_Other Asian
- race_ethnicity_Pacific Islander
- race_ethnicity_Pakistani
- race_ethnicity_Polynesian
- race_ethnicity_Samoan
- race_ethnicity_Thai
- race_ethnicity_Tongan
- race_ethnicity_Unknown
- race_ethnicity_Vietnamese
- race_ethnicity_White
- seer_historic_stage_a_Distant
- seer_historic_stage_a_In situ
- seer_historic_stage_a_Localized
- seer_historic_stage_a_Regional
- seer_historic_stage_a_Unstaged
- sex_Female
- spanish_hispanic_origin_Cuban
- spanish_hispanic_origin_Dominican Republic
- spanish_hispanic_origin_Mexican
- spanish_hispanic_origin_Non-Spanish/Non-hispanic
- spanish_hispanic_origin_Other specified Spanish/Hispanic origin (excludes Domini-

can Repuclic)
- spanish_hispanic_origin_Puerto Rican
- spanish_hispanic_origin_South or Central American (except Brazil)
- spanish_hispanic_origin_Spanish surname only
- spanish_hispanic_origin_Spanish, NOS; Hispanic, NOS; Latino, NOS
- spanish_hispanic_origin_Uknown whether Spanish/Hispanic or not
- 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.

### A.2 Lung Cancer Feature Selection

The feature set used as input into both the Random Forest and Neural Network models, after the transformation described in section (3.2) is given below and also available in full detail in the file `NewPatientLungML.html` .

- cs_tumor_size
- elevation
- grade_cell type not determined
- grade_moderately differentiated
- grade_poorly differentiated
- grade_undifferentiated; anaplastic
- grade_well differentiated
- histology_recode_broad_groupings_acinar cell neoplasms
- histology_recode_broad_groupings_adenomas and adenocarcinomas
- histology_recode_broad_groupings_blood vessel tumors
- histology_recode_broad_groupings_complex epithelial neoplasms
- histology_recode_broad_groupings_complex mixed and stromal 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_fibroepithelial neoplasms
- histology_recode_broad_groupings_fibromatuos neoplasms
- histology_recode_broad_groupings_germ cell neoplasms
- histology_recode_broad_groupings_gliomas
- histology_recode_broad_groupings_granular cell tumors & alveolar soft part sarcomas
- histology_recode_broad_groupings_lipomatous neplasms
- histology_recode_broad_groupings_miscellaneous bone tumors
- 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
- race_ethnicity_Filipino
- race_ethnicity_Guamanian
- race_ethnicity_Hawaiian
- race_ethnicity_Hmong
- race_ethnicity_Japanese
- race_ethnicity_Kampuchean
- race_ethnicity_Korean
- race_ethnicity_Laotian
- race_ethnicity_Melanesian
- race_ethnicity_Micronesian
- race_ethnicity_New Guinean
- race_ethnicity_Other
- race_ethnicity_Other Asian
- race_ethnicity_Pacific Islander
- race_ethnicity_Pakistani
- race_ethnicity_Polynesian
- race_ethnicity_Samoan
- race_ethnicity_Thai
- race_ethnicity_Tongan
- race_ethnicity_Unknown
- race_ethnicity_Vietnamese
- race_ethnicity_White
- seer_historic_stage_a_Distant
- seer_historic_stage_a_In situ
- seer_historic_stage_a_Localized
- seer_historic_stage_a_Regional
- seer_historic_stage_a_Unstaged
- sex_Female
- spanish_hispanic_origin_Cuban
- spanish_hispanic_origin_Dominican Republic
- spanish_hispanic_origin_Mexican
- spanish_hispanic_origin_Non-Spanish/Non-hispanic
- spanish_hispanic_origin_Other specified Spanish/Hispanic origin (excludes Dominican Repuclic)
- spanish_hispanic_origin_Puerto Rican
- spanish_hispanic_origin_South or Central American (except Brazil)
- spanish_hispanic_origin_Spanish surname only

- spanish_hispanic_origin_Spanish, NOS; Hispanic, NOS; Latino, NOS
- spanish_hispanic_origin_Uknown whether Spanish/Hispanic or not
- 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.

### A.3 Breast Cancer Feature Selection

The feature set used as input into both the Random Forest and Neural Network models, after the transformation described in section (3.2) is given below and also available in full detail in the file `NewPatientBreastML.html` .

- cs_tumor_size
- elevation
- grade_moderately differentiated
- grade_poorly differentiated
- grade_ndifferentiated; anaplastic
- grade_well differentiated
- histology_recode_broad_groupings_adenomas and adenocarcinomas
- histology_recode_broad_groupings_adnexal and skin appendage neoplasms
- histology_recode_broad_groupings_basal cell neoplasms
- histology_recode_broad_groupings_complex epithelial neoplasms
- histology_recode_broad_groupings_cystic, mucinous and serous neoplasms
- histology_recode_broad_groupings_ductal and lobular neoplasms
- histology_recode_broad_groupings_epithelial neoplasms, NOS
- histology_recode_broad_groupings_nerve sheath tumors
- histology_recode_broad_groupings_unspecified neoplasms
- lat
- laterality_Bilateral involvement, lateral origin unknown; stated to be single primary
- laterality_Paired site, but no information concerning laterality; midline tumor
- laterality_Right: origin of primary
- lng
- marital_stats_at_dx_Divorced
- marital_stats_at_dx_Married (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
- 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.

## B 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)
```

## C Model Architecture and Python Code

### C.1 Breast Random Forest Model

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

### C.2 Colon Random Forest Model

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

### C.3 Lung Random Forest Model

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

### C.4 Breast Neural Network Model

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

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

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

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

rms = RMSprop(lr=0.001)

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

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

### C.5 Colon Cancer Neural Network Model

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

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

```
modelcolon.add(Activation('relu'))
modelcolon.add(Dropout(0.05))


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

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

rms = RMSprop(lr=0.001)

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

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

### C.6 Lung Cancer Neural Network Model

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

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


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


rms = RMSprop(lr=0.001)

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

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

## Acknowledgments

This is the most common positions for acknowledgments. A macro is available to maintain the same layout and spelling of the heading.

**Note added.** This is also a good position for notes added after the paper has been written.

## References

[1] S. Raschka, *Python Machine Learning Essentials*. Packt Publishing, 2015.

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

[3] Cam Davidson-Pilon, "Quickstart – lifelines 0.8.0.1 documentation." http://lifelines.readthedocs.org/en/latest/Quickstart.html, 2016 (accessed 14 Jan 2016).

[4] H. Shin and Y. Nam, "A coupling approach of a predictor and a descriptor for breast cancer prognosis," *BMC MEDICAL GENOMICS*, vol. 7, MAY 8 2014. 3rd Annual Translational Bioinformatics Conference (TBC) / ISCB-Asia, Seoul, SOUTH KOREA, OCT 02-04, 2013.

[5] H. M. Zolbanin, D. Delen, and A. H. Zadeh, "Predicting overall survivability in comorbidity of cancers: A data mining approach," *DECISION SUPPORT SYSTEMS*, vol. 74, pp. 150–161, JUN 2015.

[6] Ben Kuhn, "Decision trees for survival analysis." http://www.benkuhn.net/survival-trees, year="2016 (accessed 14 Jan 2016)",.

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

[8] M. Bowles, *Machine Learning in Python: Essential Techniques for Predictive Analysis*. Wiley, 2015.

[9] A. Downey, *Think Stats*. O'Reilly Media, 2014.

[10] U. S. C. Bureau, "2010 fips code files for counties - geography - u.s. census bureau." https://www.census.gov/geo/reference/codes/cou.html, 2016 (accessed 18 Jan 2016).

[11] G. Developers, "The google maps geocoding api | google maps geocoding api | google developers." https://developers.google.com/maps/documentation/geocoding/intro, 2016 (accessed 18 Jan 2016).

[12] G. Developers, "The google maps elevation api | google maps elevation api | google developers." https://developers.google.com/maps/documentation/elevation/intro?hl=en, 2016 (accessed 18 Jan 2016).

[13] IOBS, "Supplemental material | paperdata." https://github.com/doolingdavid/PAPERDATA.git, 2016 (accessed 18 Jan 2016.

[14] scikit-learn developers, "3.2.4.3.1. sklearn.ensemble.randomforestclassifier." http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, 2014 (accessed 25 Jan 2016).

[15] K. Inc., "Random forests | kaggle." https://www.kaggle.com/wiki/RandomForests, 2015 (accessed 25 Jan 2016.

[16] M. Nielsen, "Neural networks and deep learning." http://neuralnetworksanddeeplearning.com/, Jan 2016 (accessed 25 Jan 2016).

[17] T. Unterthiner, A. Mayr, G. Klambauer, and S. Hochreiter, "Toxicity prediction using deep learning." http://arxiv.org/abs/1503.01445, 4 Mar 2015 (accessed 25 Jan 2016).

[18] F. Chollet, "Keras documentation." http://keras.io/, 2015 (accessed 25 Jan 2016).

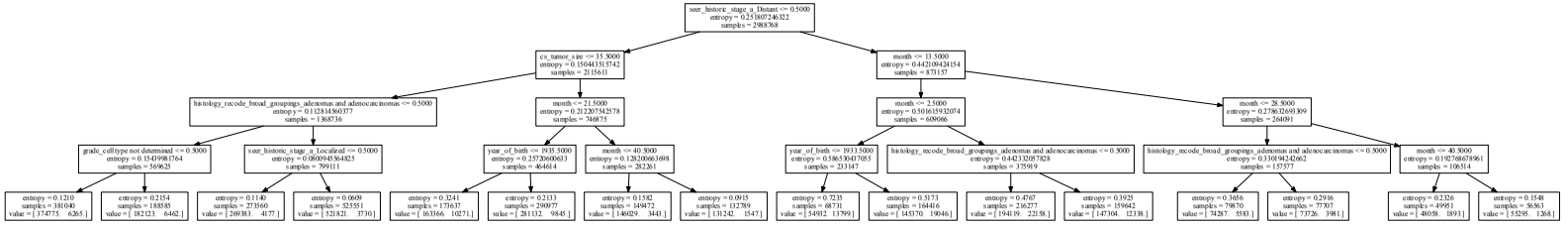


**Figure 5**. The top levels of a decision tree trained on the Lung Cancer training data.

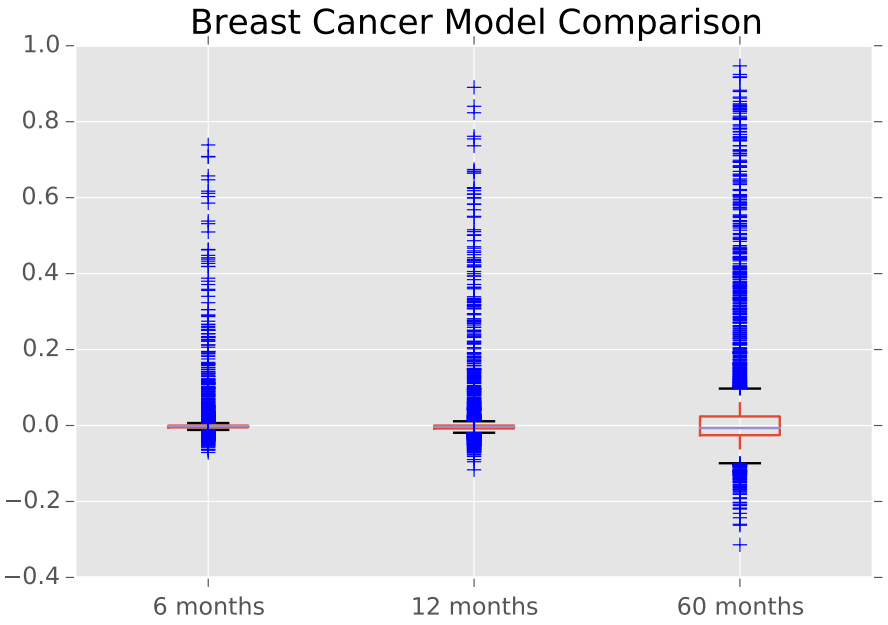


**Figure 6**. Box plots showing the distributions of the signed difference between the MLP model's prediction for the probability of surviving 6 months and the Random Forest model's prediction of the same quantity for breast cancer. The plot shows the same quantity for the 12 and 60 months classifiers.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

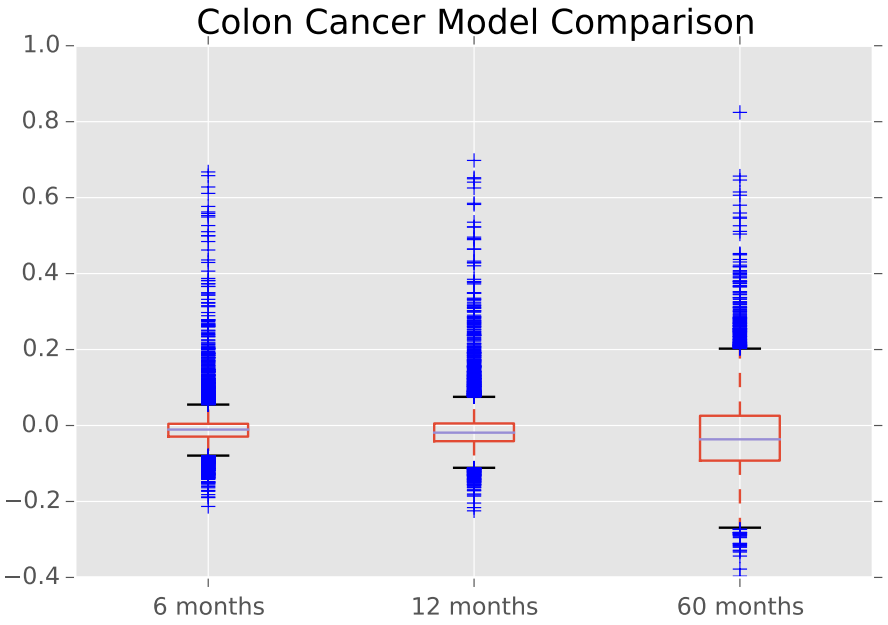


**Figure 7**. Box plots showing the distributions of the signed difference between the MLP model's prediction for the probability of surviving 6 months and the Random Forest model's prediction of the same quantity for colon cancer. The plot shows the same quantity for the 12 and 60 months classifiers.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

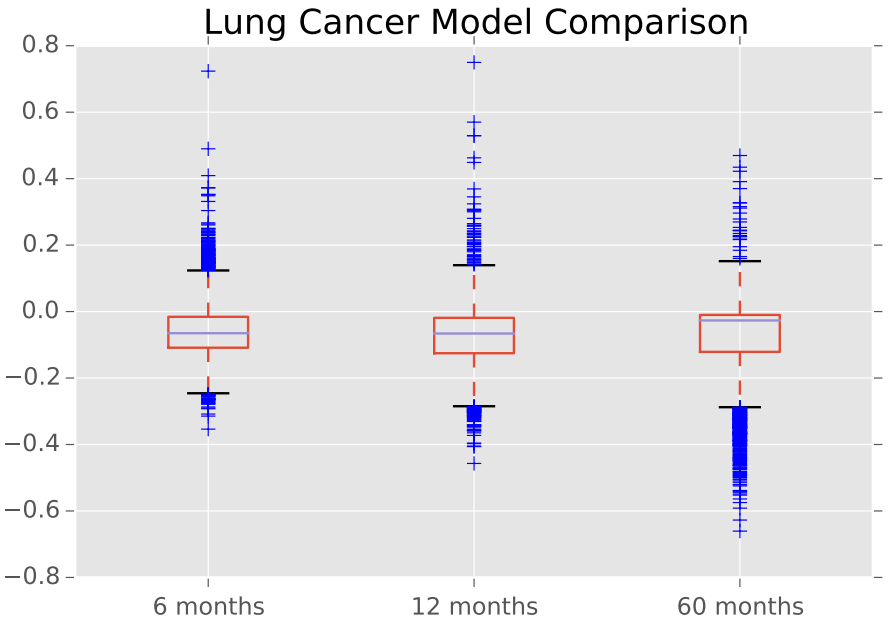


**Figure 8**. Box plots showing the distributions of the signed difference between the MLP model's prediction for the probability of surviving 6 months and the Random Forest model's prediction of the same quantity for lung cancer. The plot shows the same quantity for the 12 and 60 months classifiers. These differences were evaluated for the 5654 test patients in the colon 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.

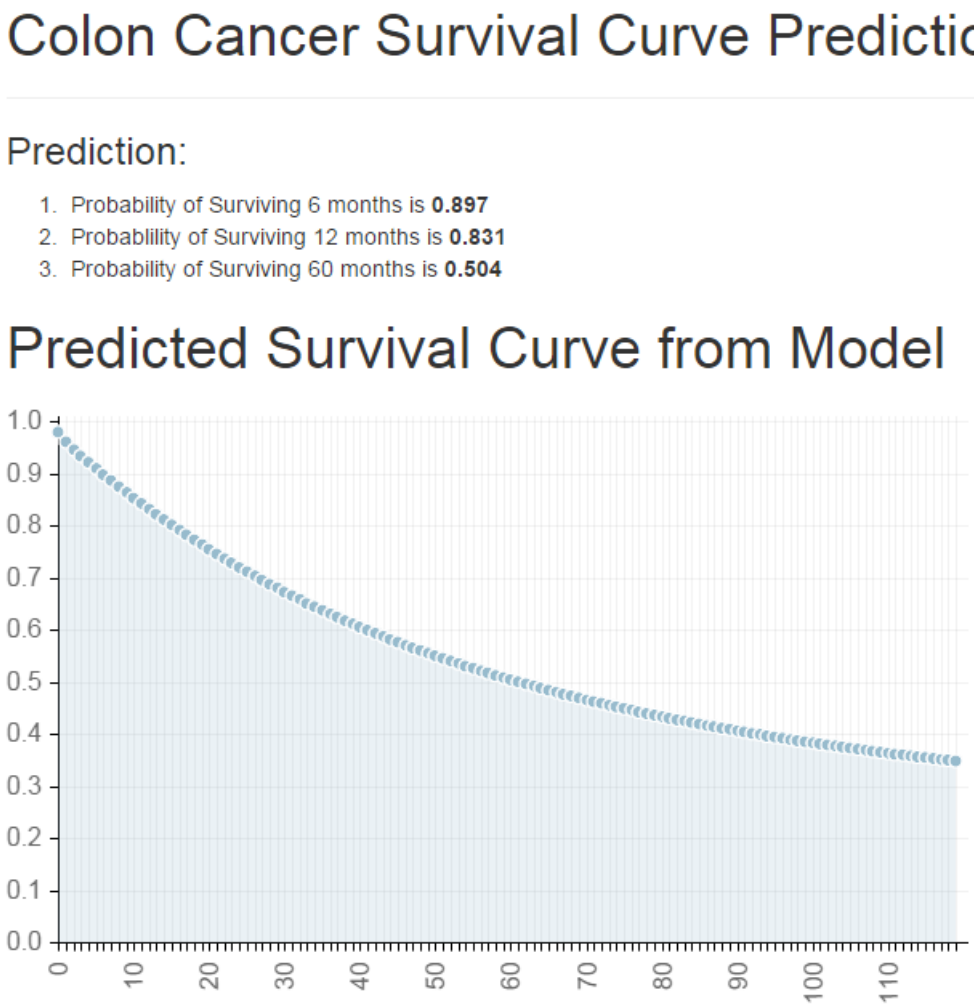


**Figure 9**. Colon Cancer Survival Curve predicted from the data in Table (15) using the neural network web app http://coloncancer.herokuapp.com/.