Predicting Mortality:

a Novel Approach Using Detailed Medical Data

Final Report

ORIE 4741: Fall 2017

Kenneth Lipke (kel89), Charlotte Wang (xw476)

***Abstract*— With mortality data released by the Centers for Disease Control and Prevention, we are trying to explore the underlying patterns, and answer the question of, "when someone will die." In this project, we aim to develop a prediction tool for insurance companies to calculate premiums and individuals to plan asset management ahead of time.**

I. INTRODUCTION

A large number of industries rely on fixed payments from customers over time in return for a service—or the promise of a service—at a later date. Consider a bank making a loan; the bank must be sure that the borrower will be able to continue making payments over the life of the loan. This business model, however, requires that the customer will be alive for the entire term of the loan.

Our goal is to develop a prediction tool for predicting life expectancy. We have data from the CDC for all the deaths in the United States and its Territories in 2015. Along with standard data points, like race, gender, and education, it also includes a selection of past medical conditions—we feel this is the most interesting and potentially impactful dimension of the data, and it will allow us to develop a worthwhile tool.

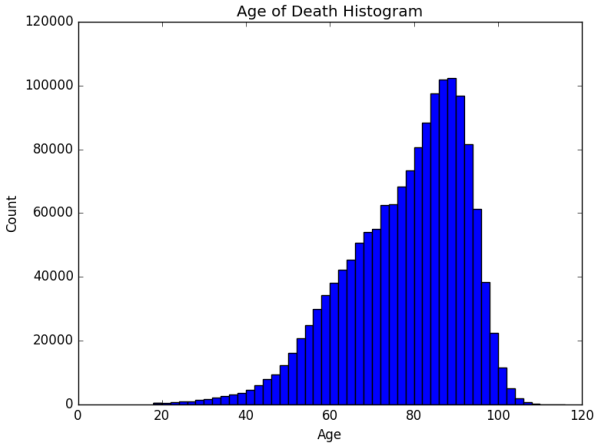
We apply a variety of techniques, including tree based models, along with a number of linear models using different loss functions and regularizes. We find that random forest and ridge regression approaches yield the best predictions, however, their accuracy is limited.

II. DATA

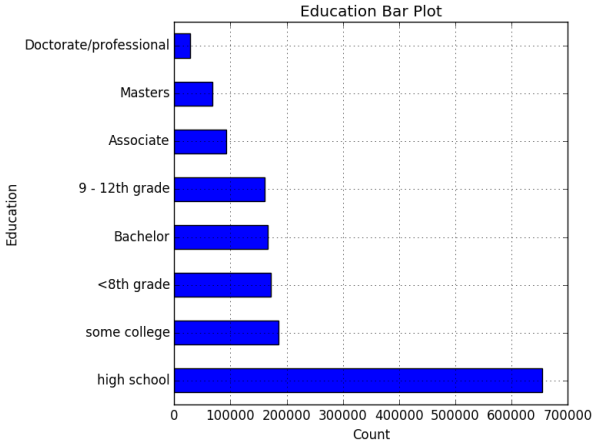
Our data represents deaths in the United States and its territories in 2015. Before cleaning, this includes 2,718,198 observations, each corresponding to a different person. The dataset also contains 77 variables, however, many of them are useless—re-codes of the same quantity or values that are irrelevant for our analysis, such as method of disposal of the body (e.g., cremation of burial). We plan to focus on the following key variables: the age of death, education, sex, marital status, race (with an extra breakout for Hispanic origin), and "entity condition," or the health conditions of the individual. In cleaning the data, we removed all observations with ages below 18, as we want to focus on adults, as well as errors in age coding (which presented themselves as 999)– there were 34,764. Next, we removed the education levels that were missing or coded as "unknown," of which there were 192,874. We removed the observations with unknown places of death (1,290 of them), as well as the 20,104 with unknown marital status. We were comfortable treating our missing entries this way as our data is quite large, so we could afford to simply eliminate observations.

A. Summary Statistics

Lets first explore the classic variables. The distribution of ages are largely was one would expect: The average



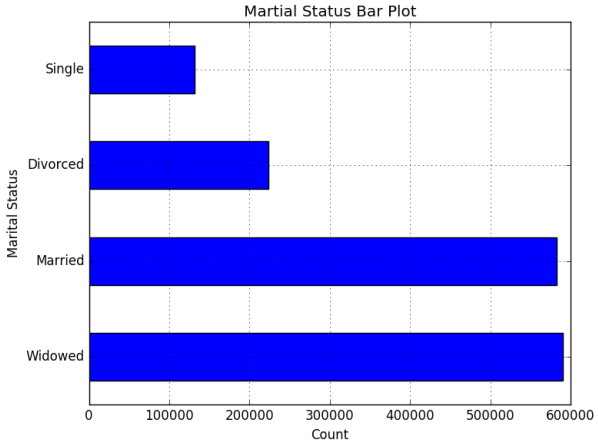
age of death is 77.41 years. Broken down by sex, women have an average of 79.74, while men are slightly lower at 74.87. Focusing on education: from the plot we see—rather



unsurprisingly—that as education goes up, there are fewer

Education	Age
Associate	74.61
some college	75.18
9 - 12th grade	76.20
Bachelor	77.17
high school	77.49
Masters	77.49
Doctorate/professional	79.36
<8th grade	82.06

people (with the exception that there are few people with less than an 8th grade education). More interesting, though, is the cross tab between education and age of death, oddly, those that live the longest tend to be those with less than an eighth grade education. Beyond that anomaly, however, there seems to be a loose correlation between education and life expectancy. We also consider marital status:



Marital Status	Age
Divorced	64.81
Married	70.81
Single	73.93
Widowed	86.16

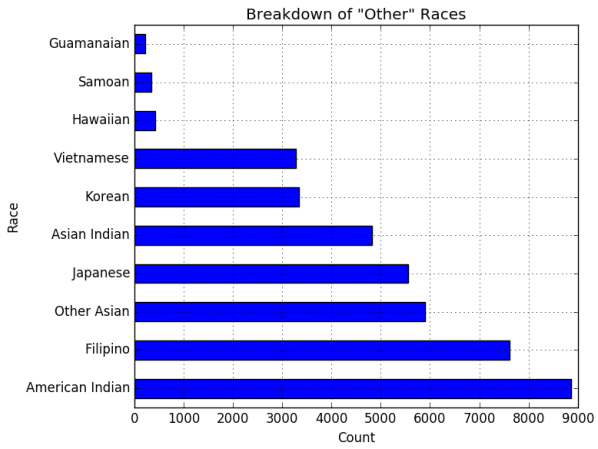
The distribution is not especially surprising, however the cross tab shows that widows and widowers live significantly longer than the rest of the population, while single people live significantly shorter. This is something we investigate further in our models.

Lastly, before the medical data, we consider race. 85% of those represented in the dataset are white, 11% are black, and the remaining 4% are as shown in the following figure (on the top of the next column):

One of the most important takeaways from those graphs and statistics is that there is significant variability in our data set, which gives us considerable freedom to try complicated models.

B. Medical Data

We have data on not only the main cause of death, but also on up to four diseases that afflicted each individual. Every



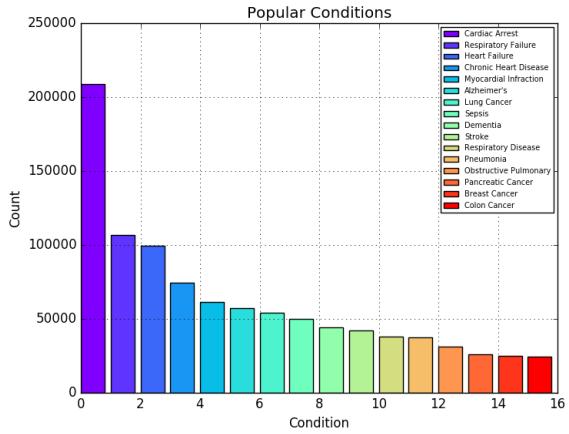
Race	Age
Samoan	65.34
American Indian	68.69
Guamanaian	69.94
Other Asian	69.94
Hawaiian	70.66
Black	71.81
Asian Indian	73.20
Vietnamese	74.12
Filipino	75.21
Korean	76.76
White	78.26
Chinese	79.46
Japanese	83.68

observation has an entry for the main cause of death, but not all of them have second, third, and fourth conditions. Specifically, there are 515 thousand observations without a second condition, 905 thousand without a third condition, and well over a million without a fourth condition (in the pre-cleaned dataset). In the coming analysis we treat these conditions in a variety of ways, however, the most common we explored was creating a dummy column for each of the diseases in the dataset, regardless of its place as the first, second, third or fourth condition listed. We feel this is appropriate as the documentation for the data is vague regarding the order in which conditions are listed, i.e., it does not appear that the condition in the second slot is appreciably more important than the condition in the third or fourth.

The condition data is coded according to the World Health Organization's ICD 10 protocol. Each condition is coded as a letter, followed by two numbers. In total, our data has 1,470 different conditions present. We remove the observations corresponding to death from non-natural causes (e.g., car accidents and gun violence), as these, we feel, are too random to be predicted with the given data.

We also limited our analysis to conditions that were present more than 2,000 times. Practically, this reduced our dimensionality, giving us a more feasible number of features with which to do large scale analysis. After removing the obscure conditions, we are left with our final dataset of 1,528,922 observations.

To give a feel for the breadth and frequency of conditions, the following figure depicts the top 16 most popular in our dataset:



C. Analytical Setup

We split our data into a training and a test set. The training set is comprised of 75% of the data. We then further divided the training set creating a validation set, which is 25% of the original data. As that still leaves over 750 thousand observations for the fitting processes, we feel this is an appropriate division that takes advantage of the large scale of our data, while balancing computational limitations. We restrict the test set for the final iterations of each of our model types in order to limit out-of-sample error estimates.

III. Baseline Analysis

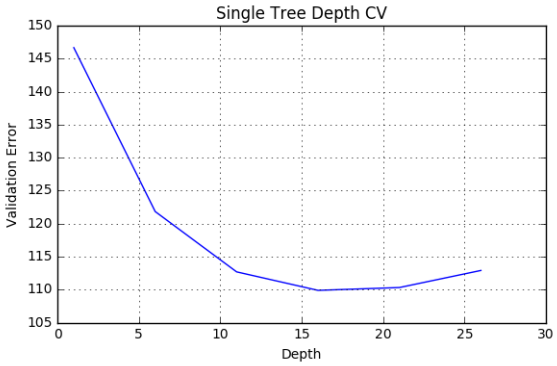
In order to judge the success of our various modeling methods we considered the naive approach of guessing the median age of death for observations in the training set for each observation in the test set. This will provide us with a baseline performance metric to which we can compare our other models. The median age in the training set is 80 years old. When we apply this guess to the test set we get a mean squared error of 212.8, and a mean absolute error of 11.4. To further get a feel for the distribution of errors we look at the error quantiles: the 25th error quantile is -12, and the 75th error quantile is 8. This tells us that this naive method has left skewed errors. We can also consider the histogram of errors. We will include such plots for our other methods, but for this method as we are just "guessing" a single number for all observations, the error histogram is simply a leftwards translation of the age distribution we showed in the summary statistics section. Moving forward, any prediction worthwhile should, at worst, do better than this naive guess.

IV. Tree Based Analysis

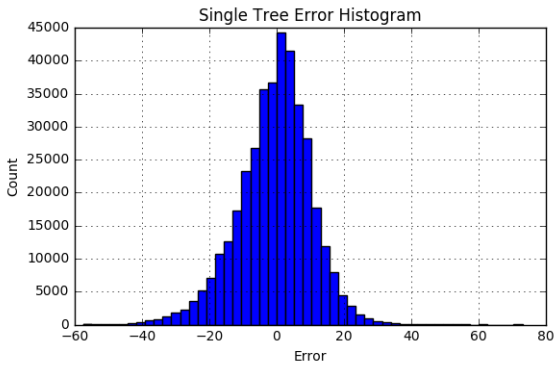
A. Single Tree

We decided to pursue tree based methods as we anticipate that they will preform very well on this data as they succinctly capture interactions between variables, without us having to explicitly enumerate them. As the medical conditions are coded in individual dummies, capturing all possible interactions in a regression setting is very difficult, however trees should capture these automatically. As a baseline, we consider a solitary regression tree.

In fitting the single tree we have one main parameter to choose: the depth of the tree (as determined by the cost complexity pruning parameter). To do so, we fit a number of models, then assessed their accuracy on the validation set. Based on the figure we choose to predict with a tree



depth of 15. We see that after a depth of 15 the trade-off between prediction accuracy and complexity flattens, and later prediction accuracy actually gets worse. Our aim is to balance a desire for a simple model with the desire for one that has low error. The plot shows us that a depth of 15 balances these goals well. Fitting a tree on the whole of the training data and running it on the test set we found a mean squared error of 116.8, and a mean absolute error of 8.3. The



25th error percentile is -6.89 years, and the 75th percentile is 6.2 years. We see that this error distribution is significantly tighter (and more symmetric) than the baseline naive guess.

We can also use a single tree to inform more advanced analysis. Running the single tree analysis using the `rpart` package in R gives information on other potential branches: when fitting the tree, the software iterates over all possible features on which to branch, and it chooses the one that yields the greatest decrease in the chosen error metric (in our cases, the mean squared error). The algorithm keeps track of

the features that are not selected as the best on which to split, and it ranks them. For example, at the first split it chooses to use marital status, however, it notes that sex, the presence of Alzheimer's, or heart conditions are other splits that would have had significant impact. In manually looking through its chosen splits, we see that often the medical information is second or third, but it is rarely chosen as the best split. This is not what we would have expected; a priori we thought the medical conditions would by far be the best predictors of mortality, when it instead appears that other factors are more important. Expanding on that we can look at the reduction in the GINI coefficient (the purity of each end node) related to each different feature. The larger the reduction caused by a given feature, the more important the feature. The table shows the top few features. For this reason, we feel that a

Feature	GINI Reduction
Marital Status	0.6092
Dementia	0.0347
Alzheimer's	0.0335
Heart Disease	0.0299
Sex	0.0250
Race	0.0234
Alcoholism	0.0191
Education	0.0179

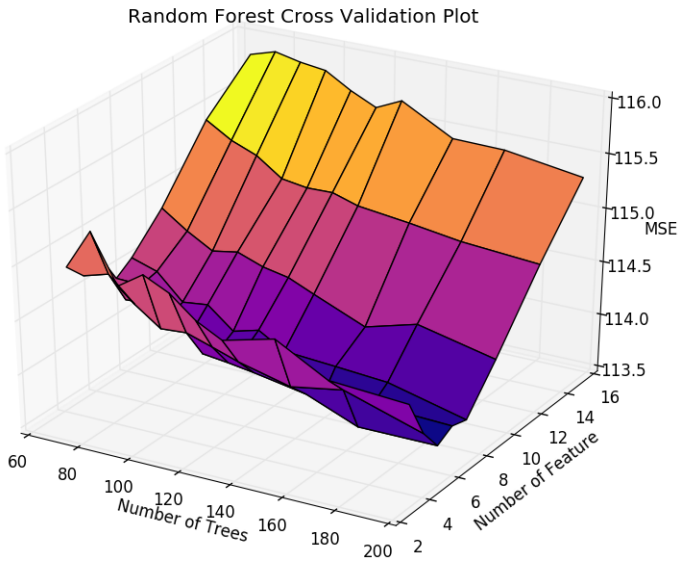
random forest approach may yield significant improvement in prediction accuracy, as at each step the number of potential splits is randomly limited. This will force different trees to explore other variables (i.e., more medical conditions) that may initially lead to less of a prediction improvement, but it may down the road result in a superior model.

B. Random Forest

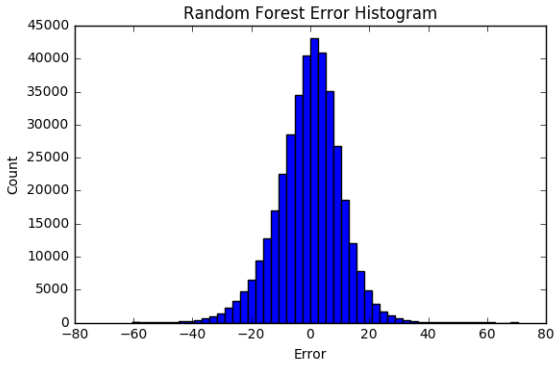
We decided to apply a random forest based on our findings from the single tree analysis. In fitting a random forest, at each step only a limited number of features are considered. As we saw in the single tree, despite all features being considered, over and over only a few were actually chosen for the split. We think that limiting feature choice at each step may lead to better predictions.

We have two main parameters to choose when fitting a random forest: the number of feature to consider at each split, and the number of trees in the forest. In order to decide on these numbers we fit a large number of random forests on our training set, and assessed their fit on the validation set; the result can be seen in the Random Forest Cross Validation Plot.

In the plot, we have number of trees and number of features on the x and y axes and the validation error on the z axis (measured as mean squared error). Unsurprisingly, as the number of trees increases the error decreases, however, we see that after 150 trees the decrease in the error is negligible. Looking at the number of features there is a very clear "U" shape, with the minimum falling at 6 features. Based on that, we choose to fit our final random forest with 150 trees, and limiting the splitting process to choosing from 6 random features at each step.



The final random forest has a mean squared error on the test set of 113.7, and a mean absolute error of 8.16 years. The distribution of errors is as follows:



The 25th percentile of the error is -6.71, and the 75th percentile is 6.28. Comparing this to the single tree, the improvement is rather underwhelming. There is a decrease in mean squared error and mean absolute error, but the magnitude of improvement is significantly smaller than the increase in accuracy we got when we moved from the naive guess to the single tree. Looking at the error histogram, the random forest seems to make more evenly distributed predictions: we see that the random forest histogram takes a very normal shape, while the single tree histogram is less clean. However, the 75th error percentile for the random forest is actually higher than for the single tree (6.28 years versus 6.20 years).

Like with a single tree, we can attempt to judge the importance of individual features by looking at the reduction in the GINI coefficient. The top coefficients are as shown in the next table. We see this is significantly different than with a single tree. The top three most important features are not medical in nature. Specifically, the most important

Feature	GINI Reduction
Marital Status	0.2721
Education	0.0802
Race	0.0372
Alcoholism	0.0283
Dementia	0.0259
Sex	0.0247
Heart Disease	0.0212
Hispanic Origin	0.0206

feature is marital status. After our summary statistics, this is not too surprising; recall, we saw that divorced people lived on average more than 20 years less than widowed people. Looking at a higher level, we are seeing roughly the same variables (just in a different order) as we saw in the first single tree, confirming their importance.

C. Conclusion on Trees

While the random forest is better than a single tree, the prediction accuracy is still quite limited: looking at the width of prediction errors, we still get an error margin of 12 years most of the time. Next we will consider a series of linear models that will hopefully improve on these trees.

V. REGRESSION ANALYSIS

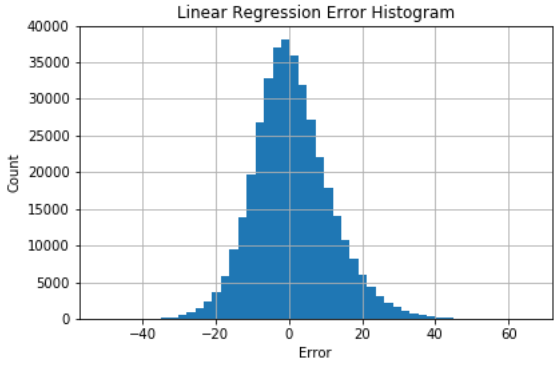
A. Introduction

Unlike the tree models we previously saw, linear models do not natively capture interactions between features. As all of our prediction data is categorical, we were not concerned with non-linearities which would otherwise be captured in polynomial regression, instead, we focused on interactions as our main feature engineering challenge. As our medical data is the most interesting and novel aspect of the data (the one which we feel hast he potential to add the most to current prediction methods) we focused on potential interactions with these. As our dataset has over 100 medical conditions, each coded as its own dummy, any interaction would increase the number of features by at least 100. This is a problem for two reasons, firstly despite our dataset being quite large, we are still concerned about the possibility of over fitting. Secondly, because our dataset is so large, adding more features slows our modeling fitting substantially, as most of the algorithms used to fit our models are quadratic in the number of features.

As a result we were very selective in the features we choose to interact with the health data. To decide, we considered the cross tabulations presented in the summary statistics section. From these, it appeared that marital status, race, and education had the greatest correlation with age of death. Therefore we decided to interact with these.

B. Simple Linear Regression

As a baseline for our other regression techniques, we ran a simple linear regression. After fitting the ordinary least square model on the training set and testing on test set, we found a mean squared error of 114, and the mean absolute error is 8.18 years. This is quite good: it does significantly better than the naive guess, and it even beats a single regression tree. The distribution of errors is as follows:



The 25th percentile of the error is -6.06, and the 75th percentile is 7.11. Comparing this to the naive guess, the improvement is significant and the errors' distribution is tighter. One way to improve on a linear model like this is to add a regularization term; this will introduce bias to the model, but it will hopefully reduce model variance and lend itself to better generalization—reducing the likelihood of over fitting.

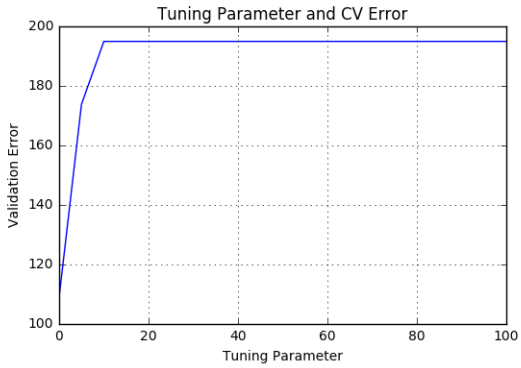
C. Quadratic Loss with ℓ_1 Regularization

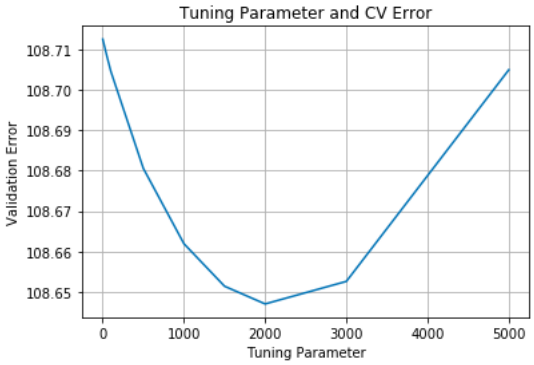
Next, we consider a linear model using quadratic loss with ℓ_1 regularization, also known as lasso regression. We decided to employ this approach because, as discussed in the introduction to this section, we have a large number of features due to the interactions, and in order to reduce model variability, it would be nice to eliminate some of those features. Lasso, unlike ridge regression (which we will see later), will enforce strict subset selection—which means that it will force some coefficients all the way to zero. For this model, our objective is:

$$\min_w \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i \cdot w)^2 + \lambda \sum_{j=1}^{d}|w_j|$$

Where w are our regression coefficients, n is the number of observations, x and y are our input and output data, respectively, and d is the number of features considered. As shown above, our only tuning parameter is λ which controls how aggressively we wish to penalize large coefficients.

As with any regularization method that focuses on the size of coefficients, we first ensured that all of our data was appropriately scaled, as to not create any unnecessary bias. Next we cross validated over a selection of λ's. We first started with a wide range, then narrowed down to those shown in the following figure. This is an interesting and unusual cross validation plot. We tried a wide range of λ's ranging from very nearly zero, to quite large. We see that the error is more or less monotonically increasing in λ, and the smallest validation error comes when λ is quite nearly zero. Now if we look back at the objective we aim to minimize here, we see that $\lambda = 0$ makes it equivalent to





an ordinary linear regression. This cross validation is telling us, therefore, that the best predictions come when we are not regularizing (at least not with an ℓ_1 regularizer). As discussed before, the ℓ_1 regularization encourages a sparse solution. The fact that our testing is showing that we should ignore this regularizer can be interpreted as meaning that either a linear model is not a good fit for this data at all (which does not seem to be the case based on how well the standard linear model performed), or that the true linear model is not sparse, i.e., that all predictors are important. To evaluate these possibilities we will consider another type of regularizer that does not encourage sparsity.

D. Quadratic Loss with ℓ_2 Regularization

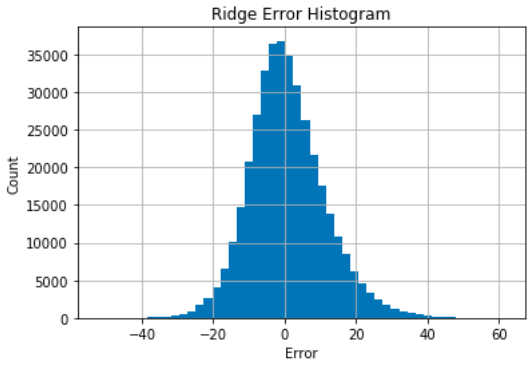
Next we consider quadratic loss with ℓ_2 regularization, also know as ridge regression. Unlike standard, unregularized, linear regression which yields unbiased regression coefficients, ridge regression allows us to regularize coefficients. Though, different from the lasso described previously, ridge will not force any coefficients all the way to zero. After analyzing the performance of the lasso above, and noticing that it does not provide any benefit over standard linear regression, we decided to try a new regularizer in the form of the ridge. Like most forms of regularization, this will introduce some bias into our model, but it should decrease model variance and the possibility of over-fitting, and hopefully improve prediction accuracy. For this model, our objective is:

$$\min_w \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i \cdot w)^2 + \lambda \sum_{j=1}^{d} w_j^2$$

We cross validated to find the best tuning parameter λ which minimized validation mean square error.

This graph shows the cross validation error given different parameters. As we can see, when the ℓ_2 regularization parameter is around 2000 we obtained the lowest mean squared error on the validation set.[1] This large parameter defines the strength of regularization, indicating that we heavily penalize the size of the regression coefficients.

We fit the final ridge model with this tuning parameter and obtained a mean squared error of 113.1. Unfortunately this is but a tiny, likely spurious, improvement over the standard linear regression. In choosing the tuning parameter, our validation errors were significantly lower (around 108—as shown in the figure), and it was our hope that our test error would be similarly good, however, that was clearly not the case. Nevertheless, we can look at other error metrics for this model: the mean absolute error is 8.18 years. The distribution of errors is as follows:



The 25th percentile of the error is -6.26, and the 75th percentile is 6.87. Comparing this to the naive guess, the improvement is significant and the error's distribution is tighter. Though, as already mentioned, ridge regression does not provide a huge improvement over unregularized linear regression.

E. Huber Loss with ℓ_2 Regularization

Lastly, we decided to use a robust regression technique. It is possible that our data contains some large outliers which may be having undue influence on our results. The quadratic loss functions that we have seen in all models up to this point can be greatly impacted by outliers. We will now use a Huber loss function which is robust to outliers:

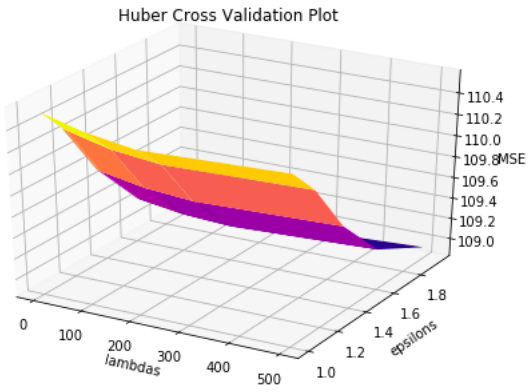
$$\text{huber}(z) = \begin{cases} \frac{1}{2}z^2 & |z| \le \epsilon \\ |z| - \frac{1}{2} & |z| > \epsilon \end{cases}$$

[1] The careful reader may notice that the various cross validation plots show a wide variety of scales for tuning parameters in the different models. One may object and say we should try the same large range on all our models to ensure nothing is missing. In fact, we did, but to make the graphs more meaningful, we are only showing the ranges of importance for each.

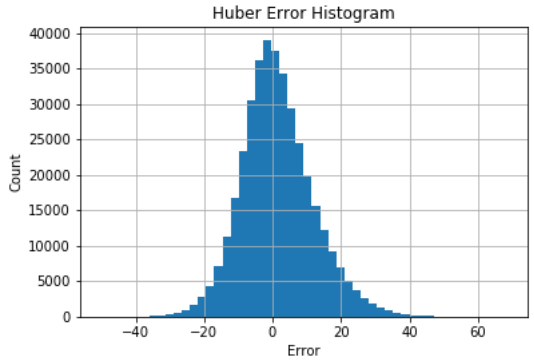
We see that for very small errors (where z is the error) the loss is quadratic, however for large errors, $|z| > \epsilon$, the loss is robust. We decided to include and ℓ_2 regularization here as well. Our resulting objective is:

$$\min \frac{1}{n} \sum_{i=1}^{n} \text{huber}(y_i - x_i \cdot w) + \lambda \sum_{j=1}^{d} w_j^2$$

We cross validated over each combination of λ and ϵ once again to determine the optimal value for the tuning parameters that provides best solution for the model. From the following plot, we can see the smallest validation mean squared error comes when λ = 200 and ϵ = 1.95.



Finally, using the tuning parameters to fit Huber model again, we obtained a mean square error on the test set of 115.1. This actually preforms worse than standard linear regression, leading us to believe that a quadratic loss is superior in this application. Nevertheless, we can look at other error metrics for this model: the mean absolute error is 8.21 years. The distribution of errors is as follows:



The 25th percentile of the error is -5.70, and the 75th percentile is 7.48, implies that the Huber model does not have a relatively symmetric error distribution. As already mentioned, Huber loss regression model does not provide any improvements over unregularized quadratic loss linear regression, suggesting that our data set may have been cleaned well enough to exclude significant outliers.

VI. Model Evaluation

In total, we applied six different models to this problem. To determine prediction accuracy, we traced four main metrics, the mean squared error, the mean absolute error, as well as the 25th and 75th percentiles of prediction error. Mean squared error is the most often used, but for this problem mean absolute error and the percentiles may be more meaningful. Our goal is to predict the age at which someone will die. The mean absolute error tells us on average how far off we are, and the percentiles indicate how fat the error distribution is, as well as if it is skewed, i.e., if we usually predict people die older or younger than average.

Based on these metrics we found that both a random forest approach and ridge regression do about equally well. The random forest has slightly lower mean absolute error (8.16 versus 8.17), and its 25th to 75th percentile width was slightly narrower (12.99 versus 13.31). All of of models did significantly better than a naive method of simply guessing the median age of death for everyone indicating that all our efforts were not futile.

However, from a practically significant standpoint, our two top methods are both trivially close in accuracy. This is interesting as it does little to inform about the underlying structure of the data. Trees split up the input space in fundamentally different ways than regression techniques, so the fact that neither is the clear winner indicates that likely neither accurately represent the true form of the data. This leaves room for further analysis using other more advanced techniques. In terms of confidence in our predictions, one can look at the error percentiles. These are meant to give a sense of how far off we are from the truth, most of the time (50% based on these percentiles). Even for our best models, these are quite wide. Our random forest has 50% accuracy width that is over 10% of the average life span in our data set. Although, it is important to note that these prediction errors do appear to be symmetric about the mean. Nevertheless, we are not overly confident in our predictions.

As we saw, none of these models are remarkably accurate, which limits how they can be applied. As we were not able to predict age of death within a year or two, this may not be a tool that could be used in a high stakes industry, like finance. However, it may be fine for personal use, giving individuals a better sense of their life expectancy for retirement planning purposes.

VII. Conclusions

Accurate death prediction can be used as a prediction tool for insurance companies to calculate premiums and maximize profits, and as a reference for individuals to plan asset management. Using medical data on over 1 million people we tried a variety of tree based and linear models. The best of which were a random forest approach, and a ridge regression, though we did not have tremendous success with either. We were able to predict the age of death within roughly 8 years, which is much better than using the median age of death in the dataset as the prediction for everyone, which was only able to predict death within 20 years most of

the time. As a result, we feel that we have made worthwhile progress in answering this question. We also feel, though, there is much room for further analysis applying other learning methods. Some methods we considered but did not have a chance to apply were methods like K-Nearest Neighbors, which may better be able to capture non-linearities in our data, or deep neural nets, which also may be better able to handle interactions and connections which we were unable to consider.

REFERENCES

[1] Centers for Disease Control and Prevention. National Vital Statistics System. Mortality Data, 2015. https://www.cdc.gov/nchs/nvss/deaths.htm