Mortality Group Midterm Report - 10/09/2020

Rustom Ichhaporia

Overview

The mortality section of the project aims to predict the mortality of a person given their current socioeconomic factors such as health, income, and family status. Although no long-term mortality prediction can be perfectly accurate, getting a probabilistic sense of life expectancy can be useful for financial planning. For example, it allows someone to optimize their portfolio with a more accurate expectation of when they will enter retirement.

At the beginning, I was assigned to work in a group, but my partner dropped out of the research program so I have been working by myself.

Preliminary Exploration

For the first few weeks, I attempted to work with a dataset from the CDC recommended by Linfeng Zhang. The dataset was called Mortality Multiple Cause-of-Death Public Use Record and is linked below. It contained location, family, date, and cause of death features for almost every death in the United States for the past few decades. I began by working with reading in and transforming the data, which was time consuming because of the complex nature of the data guide (it was not written in a standard table). Some of this work was documented in earlier weekly reports. However, this dataset did not contain information about the income or occupation of the deceased, which is an important element of our analysis. We considered simply combining predictions from an income model and a health model, but decided against this because it is statistically dubious and may have ignored interaction terms that would affect results. The code from this portion of my work has been omitted from this report, but some of it can be found in my earlier weekly reports.

https://www.cdc.gov/nchs/nvss/mortality_public_use_data.htm

New Dataset Search

After deciding to move on from the CDC dataset, I began looking into datasets that combined all three of our relevant items: economic status, health status, and the target variable, mortality. This was somewhat difficult, because most publicly available, large-scale datasets in this area only contain two of those three variables. At one weekly meeting, we discussed the potential of combining datasets as described above using a statistical methods such as hierarchical modeling and multiple frames as described in the paper linked below which was recommended by Yong Xie. However, I decided to continue looking for datasets and found the dataset from the National Longitudinal Mortality Study that incorporates all three of the variables mentioned above. After applying for approval and getting access to it, I began to work with the data. While it has more limited health information and fewer data points, I will attempt to extract meaningful insights from the data.

https://projecteuclid.org/euclid.ss/1494489817

National Longitudinal Mortality Study (NLMS) Dataset

The most recent iteraction of the NLMS dataset linked below contains roughly 3.8 million records with 550 thousand 1.8 million mortality entries, but my sample of the data contains roughly 1.8 million entries, and 100 thousand mortality entries. The dataset follows the format of a follow-up study, in which data about the participants was collected in 1990 and their potential mortality was observed 11 years later. There are more datasets with more recent data, but they are smaller, and the process for incorporating the more recent data should not be difficult once the old data has been properly modeled.

The data contains roughly 43 features, including age, race, sex, zip code, marital status, number of members of the household, highest education, health rating, employment status/occupation, income, smoking status, and cause of death (if applicable). I have used a selection of these variables which seem most relevant, leaving out items like health insurance type and day of week of death. These can be added later if deemed useful.

https://www.census.gov/topics/research/nlms.Reference_Manual.html

In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import tree
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import mean_squared_error
from sklearn.metrics import auc # to be implemented
sns.set()
```

In [5]:
```
df_raw = pd.read_csv('NLMS_PublicUse_Release5b/11.csv').drop(['smok100', 'agesmk', 'smokstat', 'smokhom
e', 'curruse', 'everuse'], axis=1)
df_raw
```

Out[5]:

	record	age	race	sex	ms	hisp	adjinc	educ	pob	wt	...	vt	histatus	hitype	povpct	stater	rcow	tenure	citizen	he
0	88426	70	1.0	2	5.0	3.0	11.0	4.0	909	151	...	0.0	NaN	NaN	18	16	4.0	1.0	NaN	
1	88427	79	1.0	2	2.0	3.0	11.0	4.0	909	132	...	0.0	NaN	NaN	18	16	3.0	1.0	NaN	
2	88428	34	1.0	1	1.0	3.0	8.0	4.0	909	155	...	0.0	NaN	NaN	10	16	1.0	2.0	NaN	
3	88429	32	1.0	2	1.0	3.0	8.0	1.0	909	155	...	0.0	NaN	NaN	10	16	1.0	2.0	NaN	
4	88430	2	1.0	2	NaN	3.0	8.0	NaN	909	145	...	NaN	NaN	NaN	10	16	NaN	2.0	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1835067	666	19	1.0	1	5.0	2.0	4.0	8.0	909	60	...	0.0	1.0	4.0	6	16	2.0	2.0	NaN	
1835068	667	33	1.0	2	1.0	2.0	11.0	6.0	909	56	...	0.0	1.0	4.0	10	16	NaN	2.0	1.0	
1835069	668	16	1.0	2	5.0	2.0	11.0	6.0	909	60	...	0.0	1.0	4.0	10	16	NaN	2.0	NaN	
1835070	669	7	1.0	2	NaN	2.0	11.0	NaN	909	51	...	NaN	1.0	4.0	10	16	NaN	2.0	1.0	
1835071	670	6	1.0	1	NaN	2.0	11.0	NaN	909	56	...	NaN	1.0	4.0	10	16	NaN	2.0	NaN	

1835072 rows × 37 columns

Because this data takes a long time to compile and verify from multiple smaller studies, the creators included both a definite and an algorithmic indicator of death to speed up the release of the data. These features, `'inddea'` and `'indalg'`, respectively, have been combined through intersection into one target variable, `'indmort'` in the way that the reference manual recommends. The expanded list of features is below.

In [6]:
```
numerical = ['age', 'hhnum']
uneven_numerical = ['adjinc', 'health', 'follow']
categorical = ['race', 'sex', 'ms', 'hisp', 'educ', 'pob', 'hhid', 'reltrf', 'occ', 'majocc', 'ind', 'e
sr', 'urban', 'smsast', 'inddea', 'cause113', 'dayod', 'hosp', 'hospd', 'ssnyn', 'vt', 'histatus', 'hit
ype', 'povpct', 'stater', 'rcow', 'tenure', 'citizen', 'indalg']
smoking = ['smok100', 'agesmk', 'smokstat', 'smokhome', 'curruse', 'everuse']
misc = ['record', 'wt']
```

```
List of all the features and their full names is pasted below. For the full description of the f
eatures, refer to the read_pubfile5.dat file.
@1 record     $   7.     /*Record Number (page no. 6)                               */
    @8 age            2.     /*Age at Time of Interview (page no. 10)                  */
    @10 race      $   1.     /*Race  (page no.12)                                      */
    @11 sex       $   1.     /*Sex   (page no.10)                                      */
    @12 ms        $   1.     /*Marital Status (page no.13)                             */
    @13 hisp      $   1.     /*Hispanic Origin (page no. 12)                           */
    @14 adjinc    $   2.     /*Inflation Adjusted Income (page no.20)                  */
    @16 educ      $   2.     /*Highest Grade Completed (page no.14)                    */
    @18 pob       $   3.     /*Region of Birth (page no. 11)                           */
    @21 wt            4.     /*Adjusted Weight (page no. 6 )                           */
    @25 hhid      $   7.     /*Household ID No. (page no. 6)                           */
    @32 hhnum         2.     /*Number of People in HH (page no. 14)                    */
    @34 reltrf    $   1.     /*Relationship to Reference Person (page no.13)           */
    @35 occ       $   4.     /*4 Digit Occupation Code (page no. 18)                   */
    @39 majocc    $   2.     /*Major Occupation Code (page no. 18 )                    */
    @41 ind       $   4.     /*4 Digit Industry Code (page no. 17)                     */
    @45 majind    $   2.     /*Major Industry Code (page no. 18)                       */
    @47 esr       $   1.     /*Employment Status Recode (page no. 17)                  */
    @48 urban     $   1.     /*Urban/Rural Status (page no. 8)                         */
    @49 smsast    $   1.     /*SMSAST Status (page no.9)                               */
    @50 inddea    $   1.     /*Death Indicator (page no. 23)                           */
    @51 cause113  $   3.     /*Cause of Death (page no. 23)                            */
    @54 follow        4.     /*Length of Follow-up (page no. 24)                       */
    @58 dayod     $   1.     /*Day of Week of Death (page no. 24)                      */
    @59 hosp      $   1.     /*Hospital Type (page no.25)                              */
    @60 hospd     $   1.     /*Hospital Death Indicator (page no. 26)                  */
    @61 ssnyn     $   1.     /*Presence of SSN (page no. 7)                            */
    @62 vt        $   1.     /*Veteran Status (page no. 16)                            */
    @63 histatus  $   1.     /*Health Insurance Status (page no. 22)                   */
    @64 hitype    $   1.     /*Health Insurance Type (page no. 22)                     */
    @65 povpct    $   2.     /*Income as Percent of Poverty Level (page no. 21)        */
    @67 stater    $   2.     /*State Recode (page no. 8)                               */
    @69 rcow      $   2.     /*Recoded Class of Worker (page no.19)                    */
    @71 tenure    $   1.     /*Housing Tenure (page no. 14)                            */
    @72 citizen   $   1.     /*Citizenship (page no. 16)                               */
    @73 health    $   2.     /*Health (page no. 16)                                    */
    @75 indalg        1.     /*Indicator of Algorithmic Death (page no. 27)            */
    @76 smok100   $   1.     /*Smoked More than 100 Cigarettes (page no. 28)           */
    @77 agesmk    $   2.     /*Age Started Smoking (page no. 28)                       */
    @79 smokstat  $   1.     /*Cigarette Smoking Status (page no.28)                   */
    @80 smokhome  $   1.     /*Rules for Smoking Cigarettes in the Home (page no. 29 ) */
    @81 curruse   $   5.     /*Currently Use Smokeless Tobacco (page no. 30)           */
    @86 everuse   $   5.     /*Ever Use Smokeless Tobacco (page no. 31)                */
```

In [8]:
```
# indmort is the recommended combination feature of both confirmed deaths and computer-predicted deaths
 based on the data collection agency
df_raw['indmort'] = df_raw['inddea'][(df_raw['inddea'] == 1) & (df_raw['indalg'] == 1)]
df_raw['indmort'] = df_raw['indmort'].fillna(0)
df_raw['indmort'].sum()
```

Out[8]:
```
94107.0
```

Each entry also has a weight that is meant to highlight the fact that not every entry is representative of the same number of people. Hidden statistical smoothing has been used so that the applying the weights to each entry will yield a more accurate estimate of societal data, but I have not yet made use of the weights.

In [9]:
```
# "Weight" of entry, roughly 50-200. I am not sure how to apply these to the data.
df_raw.wt.describe()
```

Out[9]:
```
count    1.835072e+06
mean     1.328667e+02
std      7.247297e+01
min      0.000000e+00
25%      7.600000e+01
50%      1.340000e+02
75%      1.790000e+02
max      1.522000e+03
Name: wt, dtype: float64
```

In [11]:
```
# Selection of fewer variables for EDA purposes
used_features = ['age', 'hhnum', 'adjinc', 'health', 'occ', 'ind', 'esr', 'cause113', 'ms', 'indmort']
df_short = df_raw[used_features]
```

In [15]:
```
df_short.isna().sum() / df_short.shape[0]
```

Out[15]:
```
age         0.000000
hhnum       0.000000
adjinc      0.024124
health      0.790674
occ         0.466099
ind         0.466219
esr         0.191220
cause113    0.000000
ms          0.196846
indmort     0.000000
dtype: float64
```

In [13]:
```
mort_corr = df_short.corr()['indmort'].sort_values()
mort_corr
```

Out[13]:
```
hhnum      -0.169388
adjinc     -0.098752
ms         -0.073020
ind        -0.010118
occ         0.004227
esr         0.195555
health      0.282516
age         0.336753
cause113    0.686527
indmort     1.000000
Name: indmort, dtype: float64
```

Correlations can be useful for numerical features, but most of these features are categorical, so I decided to begin one-hot encoding and imputation of missing values. Most of the health rating entries are still missing.

In [16]:
```
df_short = df_short.astype({'occ':'category', 'ind': 'category', 'esr': 'category', 'cause113': 'catego
ry', 'ms': 'category'})
df_short.dtypes
```

Out[16]:
```
age            int64
hhnum          int64
adjinc       float64
health       float64
occ         category
ind         category
esr         category
cause113    category
ms          category
indmort      float64
dtype: object
```

In [18]:
```
df_short = pd.get_dummies(df_short, dummy_na=True)
```

Recently, I created a decision tree classifier to attempt to model the data but my results were likely very unhelpful considering the number of other items I have yet to consider to polish the data before training a model on it. I used a train-test-split and calculated the mean squared error, but I will use more formal cross-fold validation later. I had previously been filling the NaN values in the Dataframe with the means of their respective features, but I was advised that this would provide a poor estimation, especially considering the large number of missing entries. To fix this, I will begin looking through the weekly reports of other groups to see what methods they used to resolve these issues and try to implement some of them myself.

It seems like the method of imputation that other groups were using was most commonly filling in the NaN values with a specific NaN substitute, but I'm not sure how to apply this to my numerical values or what the threshold should be for dropping entries, since many of them do not contain entries for the health feature.

I am still unsure whether to use a classifier or a regression model. Although the output of death is binary, the it would be more useful in the construction of a death age probability distribution to get a likelihood score of death (e.g. 0.65% chance dead at age 70) instead of a binary response (e.g. dead or not dead at age 70), and so a logistic regression would be useful. However, it was suggested that because death is a categorical variable, I should use a classifier. I would appreciate feedback on which one I should attempt to begin working with.

I apologize for my slow progress, as I am working by myself and have not learned many of the advanced methods that some other groups are using. I have excluded the code for my decision tree and some of the other EDA that I conducted earlier for the sake of clarity, but some of it can be found in my earlier weekly reports. This is part of the reason for the brevity of my report. Some of the methods that Frank Quan recommended I use are lightgbm, auc, pr auc, and recall precision f1 score, all of which I will look into in the coming weeks.

Ultimately, I hope to hope to have a model that, given the input parameters specified, can determine the probability distribution of a person's future date of death. Any help on what type of model would be ideal for this data or what method of imputation would be best would be appreciated. Thank you.