Mortality Report 10/07 - Rustom Ichhaporia

So far, I have applied for the data and received the files. I have begun reading it in below, but I am unsure of the best imputation or encoding method for the categorical variables with large numbers of categories. I have tried resolving this, and have created a model with error below.

In [1]:
```python
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 DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
sns.set()
```

In [2]:
```python
df_raw = pd.read_csv('NLMS_PublicUse_Release5b/11.csv')
```

In [3]:
```python
df_raw
```

Out[3]:

	record	age	race	sex	ms	hisp	adjinc	educ	pob	wt	...	tenure	citizen	health	indalg	smok100	agesmk	smokstat	s
0	88426	70	1.0	2	5.0	3.0	11.0	4.0	909	151	...	1.0	NaN	NaN	1.0	NaN	NaN	NaN	
1	88427	79	1.0	2	2.0	3.0	11.0	4.0	909	132	...	1.0	NaN	NaN	NaN	NaN	NaN	NaN	
2	88428	34	1.0	1	1.0	3.0	8.0	4.0	909	155	...	2.0	NaN	NaN	1.0	NaN	NaN	NaN	
3	88429	32	1.0	2	1.0	3.0	8.0	1.0	909	155	...	2.0	NaN	NaN	1.0	NaN	NaN	NaN	
4	88430	2	1.0	2	NaN	3.0	8.0	NaN	909	145	...	2.0	NaN	NaN	1.0	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1835067	666	19	1.0	1	5.0	2.0	4.0	8.0	909	60	...	2.0	NaN	1.0	NaN	NaN	NaN	NaN	
1835068	667	33	1.0	2	1.0	2.0	11.0	6.0	909	56	...	2.0	1.0	1.0	NaN	NaN	NaN	NaN	
1835069	668	16	1.0	2	5.0	2.0	11.0	6.0	909	60	...	2.0	NaN	1.0	NaN	NaN	NaN	NaN	
1835070	669	7	1.0	2	NaN	2.0	11.0	NaN	909	51	...	2.0	1.0	1.0	NaN	NaN	NaN	NaN	
1835071	670	6	1.0	1	NaN	2.0	11.0	NaN	909	56	...	2.0	NaN	1.0	NaN	NaN	NaN	NaN	

1835072 rows × 43 columns

In [4]:
```python
df_raw = df_raw.drop(['smok100', 'agesmk', 'smokstat', 'smokhome', 'curruse', 'everuse'], axis=1)
```

In [5]:
```python
# 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)
```

In [6]:
```python
# "Weight" of entry, roughly 50-200. I am not sure how to apply these to the data.
df_raw.wt.describe()
```

Out[6]:
```
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 [7]:
```python
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']
```

In [8]:
```python
df_short = df_raw[['age', 'hhnum', 'adjinc', 'health', 'occ', 'ind', 'esr', 'cause113', 'ms', 'indmort'
]]
```

In [9]:
```python
X = df_short.drop(['indmort'], axis=1)
```

In [10]:
```python
y = df_short['indmort']
```

In [11]:
```python
mort_corr = df_short.corr()['indmort'].sort_values()
```

In [12]:
```python
mort_corr
```

Out[12]:
```
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
```

The above numbers are not really accurate for most of the features because they are nonordinal categorical variables, so their correlation is not useful until they have both been imputed and one-hot encoded. As seen below, there are many categories for some of the variables, so algorithmic encoding will be necessary.

In [13]:
```python
X.cause113.unique()
```

Out[13]:
```
array([  0,  95,  52,  64,  55,  76,  23,  19,  73,  59,  54,  34,  67,
        21,  80,  31,  61,  58,  18,  20,  96,  37,  85,  51,  32,  40,
        27,  49,  60,  33,  94,  22,  70,  25, 103,  36, 107,  43,  81,
        62,  71,  77,  38, 106, 111, 104, 105,  30,  48,  41,  26,   9,
        53,  63, 102,  29,  44,  28,  42, 108,  50, 101,  47,  99,  17,
        56,  82,  98,  93,  79,  35, 100,   4,  24,  72,  84, 113,  75,
        65, 110,  46, 109,  66,  74,  87,  57,  97,  83,  14,  68,   8,
        78,  89,  88,  91,  45,   3,  90,   5,  92,  86,  12, 112,   7,
         1,  39])
```

In [14]:
```python
X.cause113.describe()
```

Out[14]:
```
count    1.835072e+06
mean     4.845706e+00
std      1.706694e+01
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      0.000000e+00
max      1.130000e+02
Name: cause113, dtype: float64
```

In [15]:
```python
X.dtypes
```

Out[15]:
```
age           int64
hhnum         int64
adjinc      float64
health      float64
occ         float64
ind         float64
esr         float64
cause113      int64
ms          float64
dtype: object
```

In [16]:
```python
X.isna().sum() / X.shape[0]
```

Out[16]:
```
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
dtype: float64
```

In [17]:
```python
X
```

Out[17]:

	age	hhnum	adjinc	health	occ	ind	esr	cause113	ms
0	70	2	11.0	NaN	2630.0	5470.0	1.0	0	5.0
1	79	2	11.0	NaN	4700.0	5470.0	1.0	95	2.0
2	34	3	8.0	NaN	8960.0	2980.0	1.0	0	1.0
3	32	3	8.0	NaN	8960.0	5470.0	1.0	0	1.0
4	2	3	8.0	NaN	NaN	NaN	NaN	0	NaN
...	...	...	...	...	...	...	...	...	...
1835067	19	2	4.0	1.0	4760.0	4770.0	1.0	59	5.0
1835068	33	6	11.0	1.0	NaN	NaN	5.0	0	1.0
1835069	16	6	11.0	1.0	NaN	NaN	5.0	0	5.0
1835070	7	6	11.0	1.0	NaN	NaN	NaN	0	NaN
1835071	6	6	11.0	1.0	NaN	NaN	NaN	55	NaN

1835072 rows × 9 columns

In [18]:
```python
X = X.astype({'occ':'category', 'ind': 'category', 'esr': 'category', 'cause113': 'category', 'ms': 'ca
tegory'})
X.dtypes
```

Out[18]:
```
age            int64
hhnum          int64
adjinc       float64
health       float64
occ         category
ind         category
esr         category
cause113    category
ms          category
dtype: object
```

In [19]:
```python
# encoder = OneHotEncoder()
# encoder.fix(X)
```

In [20]:
```python
X = pd.get_dummies(X, dummy_na=True)
```

In [21]:
```python
X = X.fillna(X.mean())
```

In [22]:
```python
X.isna().sum() / X.shape[0]
```

Out[22]:
```
age         0.0
hhnum       0.0
adjinc      0.0
health      0.0
occ_10.0    0.0
           ...
ms_2.0      0.0
ms_3.0      0.0
ms_4.0      0.0
ms_5.0      0.0
ms_nan      0.0
Length: 717, dtype: float64
```

At this point, work must be done to convert the categorical variables using one-hot-encoding and NaN values must be imputed before creating a model (optionally, the data types of the dataframe above can be converted to categorical). K-fold cross validation will also be added afterwards.

In [23]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, y)
```

In [24]:
```python
model = DecisionTreeRegressor()
```

In [25]:
```python
model.fit(X_train, y_train)
```

Out[25]:
```
DecisionTreeRegressor()
```

In [26]:
```python
mean_squared_error(model.predict(X_test), y_test)
```

Out[26]:
```
0.031503972172323404
```

In [27]:
```python
def print_full(df):
    with pd.option_context('display.max_rows', None, 'display.max_columns', None):
        print(df)
```

In [28]:
```python
print(y.sum(), y_train.sum(), y_test.sum())
```

```
94107.0 70607.0 23500.0
```

I am not yet sure how to interpret the error of the model. I think I may be improperly handling NaN values.

In [29]:
```python
model.tree_.node_count
```

Out[29]:
```
75155
```

In [33]:
```python
prediction = model.predict(X_test)
```

In [34]:
```python
prediction[prediction != 0]
```

Out[34]:
```
array([0.44444444, 0.66666667, 0.8       , ..., 1.        , 0.6       ,
       1.        ])
```

```
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)               */
```