AI-Powered Socioeconomic Prediction of Lifespan

Rustom Ichhaporia [rustomi2@illinois.edu]*

2020-12-12

Abstract

How long will you live? This age-old question has extensive implications in the billions of risk estimations made by individuals planning for the future every day. Although never certain, a stronger approximation of an indidual's lifespan can enable more reliable future planning and greater sense of stability

1 Background

2 Dataset Selection

3 Preprocessing

The appendix of this report contains the code for training the model and saving the results in a file. It does not include the code for statistical plots.

4 Modeling

5 Results

Last week, I registered for HAL access so that I could run the hyperparameter optimization script remotely because my computer overheated and could not run it for the appropriate number of trials. I was able to upload my files and run the first half of the script, but unfortunately when running the `fmin` optimization function, the program crashes after the first of 150 loops with the error:

5.1 Limitations

[1] One significant drawback of the approach taken to estimating

*This research project was completed during my time as a research intern at the Illinois Risk Lab (https://irisklabuiuc.wixsite.com/) during the Fall of 2020. My research was a part of the AI-Powered Lifecycle Financial Planning project, which is still under development. I appreciate the help of Dr. Runhuan Feng, Dr. Frank Quan, Dr. Yong Xie, Dr. Linfeng Zhang, and my fellow interns throughout the process. Thank you!

[1] https://github.com/microsoft/LightGBM/issues/2696

6 Appendix

6.0.0.1 This section will have three columns

script.py

```
'''Imports'''

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LogisticRegressionCV
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import plot_precision_recall_curve
from sklearn.metrics import plot_roc_curve

from imblearn.over_sampling import SMOTE

sns.set()

'''Preprocessing'''

df_raw = pd.read_csv('data/11.csv')
print('Data successfully loaded.')

df_raw = df_raw.drop(columns=['smok100', 'agesmk', 'smokstat', 'smokhome', 'curruse', 'everuse'])

df_raw['indmort'] = df_raw['inddea'][(df_raw['inddea'] == 1) & (df_raw['indalg'] == 1)]
df_raw['indmort'] = df_raw['indmort'].fillna(0)

used_numerical = ['age', 'hhnum']
used_ordinal = ['povpct', 'adjinc']
used_categorical = ['stater', 'pob', 'sex', 'race', 'urban', 'smsast']
used_special = ['wt', 'indmort']

used_features = used_numerical + used_ordinal + used_categorical + used_special

df_raw = df_raw[used_features]

df_raw[used_categorical] = df_raw[used_categorical].astype('category')

df_raw = df_raw.dropna(axis=0)

df = pd.get_dummies(df_raw)

X = df.drop(columns=['indmort'])
y = df['indmort']

'''Sampling'''

X_train, X_test, y_train, y_test = train_test_split(X, y)

print('Proportion of data from minority class before SMOTE
X_train, y_train = SMOTE().fit_resample(X_train, y_train)
print('Proportion of data from minority class after SMOTE:

'''Modeling'''

model = LogisticRegressionCV(scoring='roc_auc', random_sta
print(classification_report(model.predict(X_test.drop(colu

pred_probs = model.predict_proba(X_test.drop(columns=['wt'

print(classification_report(np.round(pred_probs + 0.25), y
```