統計データ分析コンペティション 2018

特別賞（高校生の部）

機械学習による 15 歳未満人口の推定

伊藤 寛子（渋谷教育学園幕張高等学校）

審査委員長講評

機械学習により人口統計データを分析することを試み、その過程を詳細に記載した技術的に優れた論文です。論文としては、15 歳未満人口を推計することが、社会的にどんな意義があるか、目的を明確にすると良いでしょう。

入力データが「人口総数」「総面積」「可住地面積」となっていますが、この 3 変数で「15 歳未満人口」を予測することが、なぜ意味があるかなどを示すと良いでしょう。

この特別賞は社会課題の解決という審査の視点とは少し異なりますが、「技術賞」という位置づけと考えて下さい。これからも機械学習の勉強を続けてください。

機械学習による15歳未満人口の推定

伊藤　寛子

渋谷教育学園幕張高等学校１年

1.　はじめに

近年、都市への人口集中が進んでいるとか、村落において少子化が顕著であるといった報道を目にすることが多い。元々機械学習に興味があった私はこのような報道を見聞きして、市町村別の人口総数・総面積・可住地面積の値から、回帰モデルを用いた機械学習による15歳未満人口を推定できるではないかと考えた。

なぜこれらのデータに目をつけたか。それは、少子化が進んでいる地域は村落が多いと言われることから、その地域の村落度合いを測ることができるデータ、すなわち人口総数・総面積・可住地面積を用いれば良いと考えたからだ。

本研究の手法が実現することにより、例えば人口急増時の教育施設や福祉施設の需要を予測することが容易になり、公立学校の運営や、少子化対策に役立つと考えられる。

そこで本研究では、線形回帰を用いた機械学習により、市町村別の人口総数・総面積（北方地域及び竹島を除く）・可住地面積から、15歳未満人口の推定[i]を試みる。

2.　研究の方法と手順

2.1. 環境について

環境は以下の通りである。

・マシン： MacBook Air (Early 2015, macOS High Sierra バージョン 10.13.6 (17G65))
・言語： Python 3.6.5
・実行： Jupiter notebook 5.5.0
・ライブラリ・拡張モジュール：scikit-learn[ii] 0.19.1，Pandas[iii] 0.23.0, NumPy[iv] 1.14.3

また、 Jupiter notebook 5.5.0 に関連して Anaconda Navigator[v] 1.8.7 を用いた。

機械学習した推定値の評価及び、データセットからのデータの抽出には Numbers バージョン 5.1(5683) を用いた。

2.2. 手順の概要

ここでは、手順について簡単に説明する。手順の詳細については、第３章を参照のこと。

1. 必要なデータの抽出
2. プログラミング
3. 機械学習の実行（５回）
4. 推定結果の分析

---

[i] 本論文において、推定は ”（機械学習による）結果の推定・予測 ”という意味で用いている。

[ii] scikit-learn は、 Python の機械学習のためのライブラリ(1)である。

[iii] Pandas は、 Python の データ解析用のライブラリである。

[iv] NumPy は、 Python において数値計算を効率的に行うための拡張モジュールである。

[v] Anaconda Navigator は、 conda パッケージの環境およびチャンネルを簡単に管理できる、 Anaconda に含まれるグラフィカルユーザーインターフェイスである。

3. データの分析手順

3.1. 使用したデータについて

本研究では教育用標準データセットの中から、人口総数、総面積（北方地域及び竹島を除く）[vi]、可住地面積、15歳未満人口のみを抽出し、総面積順に並べ替えたものを使用した。これらのデータは CSV 形式で（テキストエンコーディング： UTF-8 ）ファイル名を test2.csv として書き出した。書き出したファイルは、内蔵ハードディスクのホームディレクトリに保存した。

3.2. プログラミング

本章では、 scikit-learn を用いて、ランダムフォレスト[vii]で教師あり学習を行うためのプログラミングについて述べる。

プログラムのフローチャートは、図 1 の通りである。

Jupiter notebook 5.5.0 を用いて、プログラミングを行なった。ライブラリ、 Python は “2.1. 環境について ” で示したものが、既にインストールされているものとする。

今回のプログラムのアルゴリズムを図 1 のフローチャートに示す。

ソースコードは資料 1 の通りである。プログラムのソースコードについては、示したソースコードにコメントとして表記しているため、説明は割愛する。また、プログラム中の任意に設定した値については、値の設定の理由を以下で説明する。

まず、トレーニングデータとテストデータの比率であるが、今回の事例では全体のデータ件数が 1740 件である。トレーニングデータが多いと精度を向上するが、評価に使うテストデータが少なすぎると適切な評価が困難となってしまう。そこで今回は、テストデータを 300 件以上とすることにした。値の桁数が多いとバグを生じやすくなることを踏まえ、有効数字 1 桁の値でテストデータの割合を指定することにした。この時、条件を満たす中で有効数字 1 桁の最小の値は百分率で表すと、 $2*10^1$ %となる。この値を割合に変換し、テストデータの割合を設定する変数である 22 行目の test_size に代入した。

22 行目で設定している、擬似乱数生成に利用する値（変数名： random_state ）は 70 としているが、これは特に意味を持たない。ちなみに、擬似乱数はトレーニングデータとテストデータをランダムに分割するために使用している。

図 1　フローチャート

[vi] 以下、総面積と表記する。また、CSVファイルに書き出して使用したデータのヘッダ名も、総面積と表記されているものとしてプログラミングしている。

[vii] ランダムフォレストとは、2001年に Leo Breiman によって提案された機械学習のアルゴリズムである。このアルゴリズムでは、ランダムサンプリングされたトレーニングデータによって学習した多数の決定木を使用する。

```
1.import pandas as pd
2.import numpy as np
3.%matplotlib inline
4.import matplotlib
5.import matplotlib.pyplot as plt
6.from sklearn import cross_validation
7.from sklearn.preprocessing import StandardScaler
8.from sklearn.ensemble import RandomForestRegressor
9.
10.# CSVファイルの読み込み(ファイル名：test2.csv, テキストエンコーディング：UTF-8)
11.df_test = pd.read_csv("test2.csv",encoding="UTF-8")
12.
13.# 入出力に使用するデータ列の指定
14.X_cols = ["人口総数","総面積","可住地面積"]
15.y_cols = ["１５歳未満人口"]
16.
17.# 入力・出力データの取得
18.X = df_test[X_cols].as_matrix().astype('float')
19.y = df_test[y_cols].as_matrix().astype('int').flatten()
20.
21.# トレーニングデータ：テストデータ＝0.8:0.2になるようにランダムに分割（乱数生成値：70）
22.X_train, X_test, y_train, y_test = cross_validation.train_test_split(X, y,
  test_size=0.2, random_state=70)
23.
24.# 入力データを正規化
25.scaler = StandardScaler()
26.scaler.fit(X_train)
27.
28.X_train = scaler.transform(X_train)
29.X_test = scaler.transform(X_test)
30.
31.# ランダムフォレストで学習
32.model = RandomForestRegressor()
33.model.fit(X_train, y_train)
34.
35.# 推定値のスコア（決定係数）を出力
36.print(model.score(X_test,y_test))
37.
38.# 推定結果
39.result = model.predict(X_test)
40.
41.# データフレームに変換
42.df_result = pd.DataFrame({
43.    "test":y_test,
44.    "result":result
45.})
46.
47.# CSVを出力（ファイル名：answer_number.csv）
48.df_result.to_csv("answer_number.csv")
```

資料１　機械学習によるデータの分析のためのアルゴリズムのソースコード

3.3. 実行

資料１のソースコードを Jupiter notebook 5.5.0 上で実行した。

この時、同じアルゴリズムを実行しても、機械学習による推定結果は変化する。そこで今回は、偶然誤差を減らすため、資料１のソースコードのアルゴリズムを連続して５回実行し、その平均値をとることにした。それに際し、ダウンロードされるファイルのデータが上書きされてしまうのを防ぐため、資料１の 48 行目、 df_result.to_csv(“answer_number.csv”) を、 df_result.to_csv(“answer_1.csv”), df_result.to_csv(“answer_2.csv”) …というように書き換える操作、すなわち結果の出力ファイル名の変更をする操作を行った。これにより、推定結果が格納されたファイルのファイル名は、１回目の結果から順に answer_1.csv, answer_2.csv, answer_3.csv, answer_4.csv, answer_5.csv となる。

ちなみに、ここで出力された CSV ファイルでは、 A 列にサンプリングの際のナンバリングされた値、 B 列（ヘッダ名： test ）に[viii]実際の 15 歳未満人口、 C 列（ヘッダ名： result ）に機械学習で推定された15歳未満人口が格納されている。

また、実行すると標準出力で出力される推定値のスコア[ix]を毎回記録した。

4.　推定結果とその評価

4.1. 推定値の結果

図２は、テストデータについて、縦軸に５回の推定値の平均、横軸に実際の15歳未満人口をとったグラフである。また、そのグラフに最小二乗法によるトレンドラインを引いたところ、その方程式は $y = 0.9635x + 272.21$ となり、正の相関が見られた。

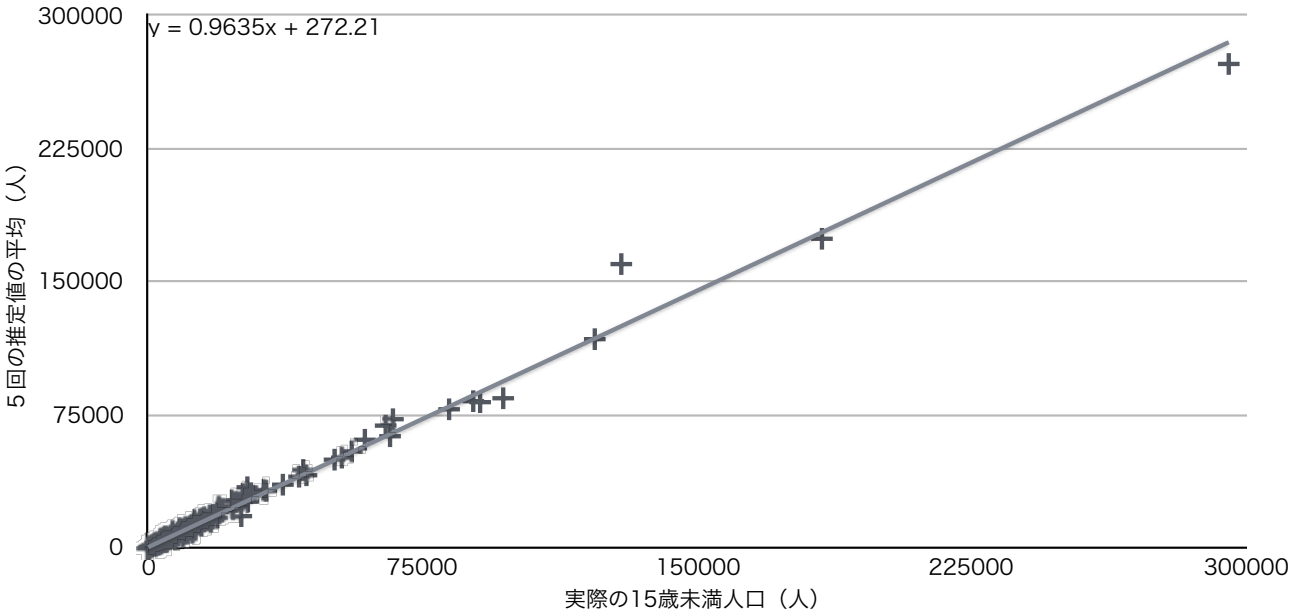

図２　５回の推定値の平均と実際の15歳未満人口

4.2. 相関関係

テストデータについて、５回の推定値の平均と実際の15歳未満人口の相関関係を調べた[x]。相関係数は 0.9948 、決定係数は 0.9896 であった。

[viii] 以下、推定値との混同を避けるため、15歳未満人口は実際の15歳未満人口と表記する。

[ix] 本論文において、スコアとして示される値は全て決定係数 (1) である。

[x] 本来であれば有効数字を意識して計算すべきではあるが、テストデータにおける市町村別の15歳未満人口の値に大きなばらつきが見られ、一番小さい桁数にあわせると、十分な評価が行えなくなるため、今回は小数点以下第５位を四捨五入して計算した。これ以降の章で求められている値についても同じ扱いを行った。

4.3. 誤差

テストデータにおいて、推定値について実際の15歳未満人口との相対誤差[xi]を求めると、図３のようになった。

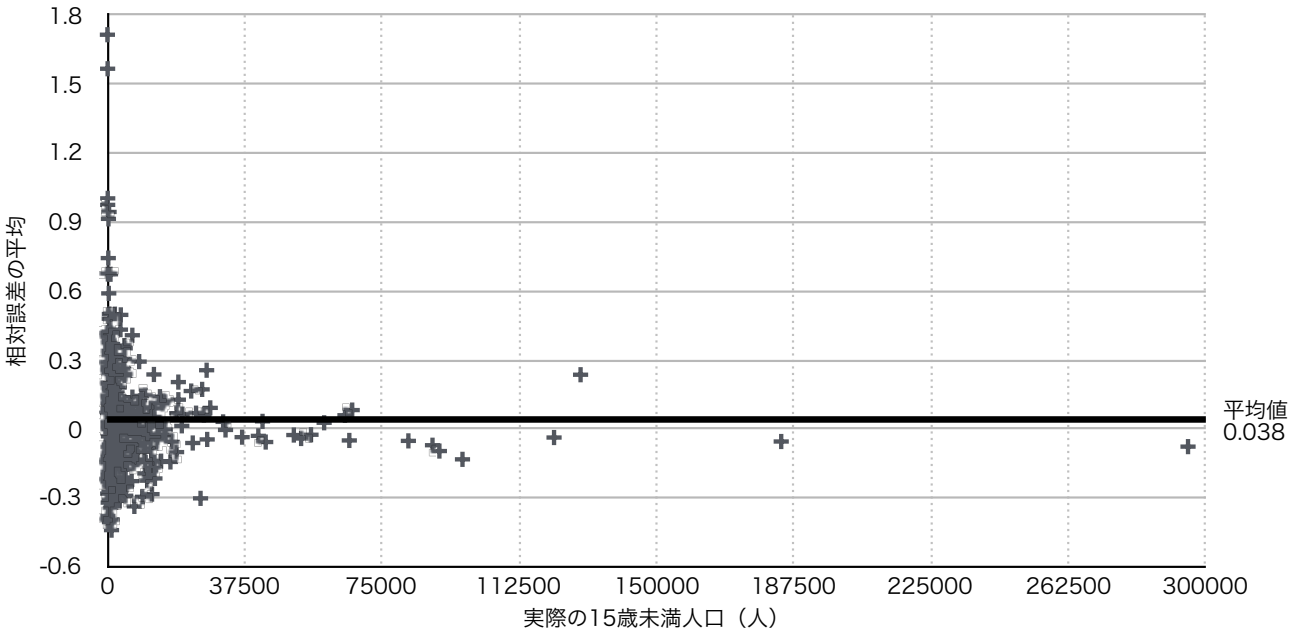

図３　実際の15歳未満人口と推定値の相対誤差

4.4. プログラム中で出力されるスコア

回帰モデルによる機械学習プログラム中で測定した推定値のスコアは表１の通りである。

表１ スコア

回数	スコア
１回目	0.9886389795692969
２回目	0.9908011704351442
３回目	0.9854055839021446
４回目	0.9814707632552758
５回目	0.9905348525568049
平均	0.987370269943733

5.　結論・今後の展望

5.1. 15歳未満人口の推定の精度

５回平均の相関係数・決定係数共に 0.9 を超える高い数値が出ており、概ね推定精度は良好であった。また、１回ごとの結果で見ても、決定係数は 0.98 を超える高い数値を安定して出しており、精度はかなり高いといえよう。

[xi] 15歳未満人口が０人の市町村については、相対誤差による推定値の評価をできないため、ここでは考えないものとする。また、以下の相対誤差やそれを用いた指標による評価を行っている部分についても同様である。なお、今回の分析においてテストデータとなった市町村のうち、15歳未満人口が０人の市町村は２市町村であった。

5.2. 課題

図３からもわかるように、実際の15歳未満人口が少ない、概ね 100 人未満の市町村で、推定値の相対誤差が大きい市町村が 1.0 を超える地域があり、予想精度が著しく低い傾向が見られた。また、テストデータに含まれる実際の15歳未満人口が 1 人以上100 人未満の８市町村について相対誤差の平均は 0.5769 であり、全テストデータの相対誤差の平均である 0.9948 に比べ大幅に低い数値であった。これは、実際の15歳未満人口が少ない地域における学習データの不足が原因である。

また、15歳未満人口と推定値の比較以外の視点からの、精度についての考察は行えておらず、精度の評価が十分であるとは言えない。

5.3.結論

前述したことを踏まえると、実際の15歳未満人口が概ね 100 人以上である地域においては、機械学習による15歳未満人口の推定手法として、本研究で示した手法は有効であるといえよう。

また、実際の15歳未満人口が概ね 100 人未満の市町村では、精度が著しく低下するため、本研究の手法により求められた推定値の利用には注意が必要である。

5.4. 今後の展望

改善すべき点として、まず、人口が少ない市町村における予測精度の向上が挙げられる。これについては、ランダムフォレストを多層にしたディープ・フォレストと呼ばれる手法による機械学習アルゴリズムを用いるのが有効だと考えられる。

また、実際の15歳未満人口と推定値の比較以外の視点から、精度の評価を行えていない。これに関しては、人口などの他のデータとの比較による評価が行えるようにプログラムの改善すること、そして多様な視点で精度を評価し精度をより向上させていくことが必要である。

そして、この手法を利活用するにはプログラミングの知識が多少なりとも必要となってしまうため、手軽に利活用できないという課題がある。この手法の利活用の促進のため、機械学習によるデータ分析を誰でも簡単に行えるようなソフトウェアの開発が必要であると考える。

他にも、機械学習は、企業誘致政策のサポートなど、より具体的な政策に特化した結果の推定にも使えると思うので、本研究を応用してより実用的なデータを得ることにも取り組みたい。

5.5. さいごに　―社会への提言―

本研究は、一高校生が機械学習により人口統計データを分析することを試みた過程を記録したものである。この論文は、機械学習で15歳未満人口を推定することを実現するだけでなく、私が機械学習を統計学の分野で今まで以上に活用することを社会に提言すべく書いた側面もある。この論文を読んで、機械学習が難しくてよくわからないものではなく、統計データを分析する上で有益な手法となりうることを知ってもらえれば、さらには読者が機械学習を統計データ分析手法の一つとして役立ててみたいと思ってもらえれば、嬉しい。

6.　参考文献

(1)　Pedregosa *et al.*, JMLR 12 “ Scimitar-learn 公式ホームページ ”, http://scikit-learn.org (2018.9.10)