Assignment4HuahaoShang

November 2, 2022

[1]:
```
# Generic inputs for most ML tasks
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn import tree
#import graphviz
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import BaggingRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor

pd.options.display.float_format = '{:,.2f}'.format

# setup interactive notebook mode
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

from IPython.display import display, HTML

from sklearn.preprocessing import StandardScaler
```

[2]:
```
vote_data = pd.read_csv('president_county_candidate.csv')
vote_data.head()
vote_data_dem = vote_data.groupby('party').get_group('DEM')
vote_data_dem = vote_data_dem.sort_values(by=['state','county'],ascending =␣
↪True)
vote_data_dem.head()
print(vote_data_dem)

vote_data = vote_data.assign(SC=lambda x: x.state + x.county)
vote_data.head()
```

```python
fraction = vote_data.groupby('SC')['total_votes'].sum()
totalv =[]
for x, val in enumerate(fraction):
    totalv.append(val)

vote_data_dem['votes'] = totalv

vote_data_dem = vote_data_dem.assign(DEM_fraction=lambda x: x.total_votes / x.
↪votes)
vote_data_dem.head()
```

```
[2]:       state             county      candidate party  total_votes    won
     0  Delaware        Kent County      Joe Biden   DEM        44552   True
     1  Delaware        Kent County   Donald Trump   REP        41009  False
     2  Delaware        Kent County   Jo Jorgensen   LIB         1044  False
     3  Delaware        Kent County  Howie Hawkins   GRN          420  False
     4  Delaware  New Castle County      Joe Biden   DEM       195034   True
```

```
[2]:          state           county  candidate party  total_votes    won
     27999  Alabama  Autauga County  Joe Biden   DEM         7503  False
     28043  Alabama  Baldwin County  Joe Biden   DEM        24578  False
     28087  Alabama  Barbour County  Joe Biden   DEM         4816  False
     28131  Alabama     Bibb County  Joe Biden   DEM         1986  False
     28175  Alabama   Blount County  Joe Biden   DEM         2640  False
```

```
               state             county  candidate party  total_votes    won
     27999   Alabama     Autauga County  Joe Biden   DEM         7503  False
     28043   Alabama     Baldwin County  Joe Biden   DEM        24578  False
     28087   Alabama     Barbour County  Joe Biden   DEM         4816  False
     28131   Alabama        Bibb County  Joe Biden   DEM         1986  False
     28175   Alabama      Blount County  Joe Biden   DEM         2640  False
     …           …                  …        …    …       …     …
     27934   Wyoming  Sweetwater County  Joe Biden   DEM         3823  False
     27938   Wyoming       Teton County  Joe Biden   DEM         9848   True
     27944   Wyoming       Uinta County  Joe Biden   DEM         1591  False
     27949   Wyoming    Washakie County  Joe Biden   DEM          651  False
     27954   Wyoming      Weston County  Joe Biden   DEM          360  False

     [4633 rows x 6 columns]
```

```
[2]:       state             county      candidate party  total_votes    won  \
     0  Delaware        Kent County      Joe Biden   DEM        44552   True
     1  Delaware        Kent County   Donald Trump   REP        41009  False
     2  Delaware        Kent County   Jo Jorgensen   LIB         1044  False
     3  Delaware        Kent County  Howie Hawkins   GRN          420  False
     4  Delaware  New Castle County      Joe Biden   DEM       195034   True
```

```
                          SC
0        DelawareKent County
1        DelawareKent County
2        DelawareKent County
3        DelawareKent County
4  DelawareNew Castle County
```

[2]:
```
          state          county  candidate party  total_votes    won   votes  \
27999  Alabama  Autauga County  Joe Biden   DEM         7503  False   27770
28043  Alabama  Baldwin County  Joe Biden   DEM        24578  False  109679
28087  Alabama  Barbour County  Joe Biden   DEM         4816  False   10518
28131  Alabama     Bibb County  Joe Biden   DEM         1986  False    9595
28175  Alabama   Blount County  Joe Biden   DEM         2640  False   27588

       DEM_fraction
27999          0.27
28043          0.22
28087          0.46
28131          0.21
28175          0.10
```

[3]:
```python
covid_data = pd.read_csv('../Assignment1/covid_data.csv')

us_country = pd.read_csv('../Assignment1/us_county.csv')

land_area = pd.read_excel('../Assignment1/LND01_land_area_columnH.xls')

land_area.rename(columns = {'STCOU' : 'fips'})
land_area.isna().any()
us_country.isna().any()
covid_data.isna().any()
us_country = us_country.dropna()
covid_data = covid_data.dropna()
joint_data = pd.merge(us_country,land_area.rename(columns = {'STCOU' :␣
 ↪'fips'}), on=['fips'], how='inner')
dataframe = pd.merge(joint_data, covid_data, on='fips')

subset_one = dataframe[dataframe['LND010200D'] > 0]
subset_end = subset_one[subset_one['population'] < 1000000]
#subset_end.loc[:,'population-density'] = subset_end['population']/
 ↪subset_end['LND010200D']
#subset_end.loc[:,'case-ratio'] = subset_end['cases']/subset_end['population']
subset_end = subset_end.assign(population_density=lambda x: x.population / x.
 ↪LND010200D)
subset_end = subset_end.assign(case_ratio=lambda x: x.cases / x.population)
print(subset_end)
```

```
subset_end = subset_end[subset_end['cases'] > 0]
subset_end['cases'] = np.log(subset_end['cases'])
subset_end['male'] = np.log(subset_end['male'])
subset_end['female'] = np.log(subset_end['female'])
subset_end['population'] = np.log(subset_end['population'])
#subset_end = subset_end.drop(columns = ['cases','male'],axis = 1)
cols = subset_end.columns
print(cols)
subset_end = subset_end.assign(log_popl_density=lambda x: x.population - np.
 ↪log(x.LND010200D))
print(subset_end)

hosp = pd.read_csv("../Assignment1/Hospital_Beds_per_County_and_per_capita.csv")
hosp.head()
hosp = hosp[['CoSt','St','ICUBeds','BedsPC','StaffPC','FoodInsc']]
data_final = pd.merge(subset_end, hosp, left_on=['county_x','state_code_x'],␣
 ↪right_on = ['CoSt','St'], how='left')

data_final.isna().any()
data_final = data_final.dropna()

data_final['BedsPC'] = data_final['BedsPC']+1
data_final['StaffPC'] = data_final['StaffPC']+1

data_final['BedsPC'] = np.log(data_final['BedsPC'])
data_final['StaffPC'] = np.log(data_final['StaffPC'])
data_final['ICUBeds'] = data_final['ICUBeds']**0.25

print(data_final)
```

[3]:

```
             Areaname   fips   LND010200D
0       UNITED STATES      0 3,794,083.06
1             ALABAMA   1000    52,419.02
2         Autauga, AL   1001       604.45
3         Baldwin, AL   1003     2,026.93
4         Barbour, AL   1005       904.52
…                   …     …             …
3193  Sweetwater, WY  56037    10,491.17
3194       Teton, WY  56039     4,221.80
3195       Uinta, WY  56041     2,087.56
3196    Washakie, WY  56043     2,242.75
3197      Weston, WY  56045     2,400.07

[3198 rows x 3 columns]
```

[3]:
```
Areaname       False
STCOU          False
LND010200D     False
dtype: bool
```

[3]:
```
fips                  False
county                False
state                 False
state_code             True
male                  False
female                False
median_age            False
population            False
female_percentage     False
lat                   False
long                  False
dtype: bool
```

[3]:
```
fips           True
county         True
state         False
lat           False
long          False
date          False
cases         False
state_code     True
deaths        False
dtype: bool
```

```
       fips           county_x  state_x state_code_x    male  female  \
0      1001     Autauga County  Alabama           AL   26874   28326
1      1003     Baldwin County  Alabama           AL  101188  106919
2      1005     Barbour County  Alabama           AL   13697   12085
3      1007        Bibb County  Alabama           AL   12152   10375
4      1009      Blount County  Alabama           AL   28434   29211
…       …                …        …           …      …      …
3134  56037  Sweetwater County  Wyoming           WY   22882   21235
3135  56039       Teton County  Wyoming           WY   11911   11148
3136  56041       Uinta County  Wyoming           WY   10505   10104
3137  56043    Washakie County  Wyoming           WY    4137    3992
3138  56045      Weston County  Wyoming           WY    3768    3332

      median_age  population  female_percentage  lat_x  …    county_y  \
0          37.80       55200              51.32  32.53  …     Autauga
1          42.80      208107              51.38  30.73  …     Baldwin
2          39.90       25782              46.87  31.87  …     Barbour
3          39.90       22527              46.06  33.00  …        Bibb
4          40.80       57645              50.67  33.98  …      Blount
```

```
…          …          …                …     …  …          …
3134       34.60       44117             48.13  41.66  …  Sweetwater
3135       39.30       23059             48.35  43.93  …       Teton
3136       35.50       20609             49.03  41.29  …       Uinta
3137       43.50        8129             49.11  43.90  …    Washakie
3138       42.90        7100             46.93  43.84  …      Weston

      state_y  lat_y  long_y        date  cases  state_code_y deaths  \
0     Alabama  32.54  -86.64  2021-02-14   6023            AL     84
1     Alabama  30.73  -87.72  2021-02-14  19105            AL    252
2     Alabama  31.87  -85.39  2021-02-14   2042            AL     48
3     Alabama  33.00  -87.13  2021-02-14   2395            AL     57
4     Alabama  33.98  -86.57  2021-02-14   5961            AL    121
…       …    …     …           …    …           …    …
3134  Wyoming  41.66 -108.88  2021-02-14   3595            WY     34
3135  Wyoming  43.94 -110.59  2021-02-14   3278            WY      8
3136  Wyoming  41.29 -110.55  2021-02-14   1994            WY     12
3137  Wyoming  43.90 -107.68  2021-02-14    873            WY     26
3138  Wyoming  43.84 -104.57  2021-02-14    617            WY      5

      population_density case_ratio
0                  91.32       0.11
1                 102.67       0.09
2                  28.50       0.08
3                  35.98       0.11
4                  88.60       0.10
…                  …         …
3134                4.21       0.08
3135                5.46       0.14
3136                9.87       0.10
3137                3.62       0.11
3138                2.96       0.09

[3092 rows x 23 columns]
Index(['fips', 'county_x', 'state_x', 'state_code_x', 'male', 'female',
       'median_age', 'population', 'female_percentage', 'lat_x', 'long_x',
       'Areaname', 'LND010200D', 'county_y', 'state_y', 'lat_y', 'long_y',
       'date', 'cases', 'state_code_y', 'deaths', 'population_density',
       'case_ratio'],
      dtype='object')
       fips           county_x  state_x state_code_x  male  female  \
0      1001     Autauga County  Alabama           AL 10.20   10.25
1      1003     Baldwin County  Alabama           AL 11.52   11.58
2      1005     Barbour County  Alabama           AL  9.52    9.40
3      1007        Bibb County  Alabama           AL  9.41    9.25
4      1009      Blount County  Alabama           AL 10.26   10.28
…     …                 …       …            …   …      …
3134  56037  Sweetwater County  Wyoming           WY 10.04    9.96
```

```
3135  56039       Teton County  Wyoming          WY  9.39    9.32
3136  56041       Uinta County  Wyoming          WY  9.26    9.22
3137  56043    Washakie County  Wyoming          WY  8.33    8.29
3138  56045      Weston County  Wyoming          WY  8.23    8.11

      median_age  population  female_percentage  lat_x  …  state_y lat_y  \
0          37.80       10.92              51.32  32.53  …  Alabama 32.54
1          42.80       12.25              51.38  30.73  …  Alabama 30.73
2          39.90       10.16              46.87  31.87  …  Alabama 31.87
3          39.90       10.02              46.06  33.00  …  Alabama 33.00
4          40.80       10.96              50.67  33.98  …  Alabama 33.98
…            …           …                  …    …  …      …    …
3134       34.60       10.69              48.13  41.66  …  Wyoming 41.66
3135       39.30       10.05              48.35  43.93  …  Wyoming 43.94
3136       35.50        9.93              49.03  41.29  …  Wyoming 41.29
3137       43.50        9.00              49.11  43.90  …  Wyoming 43.90
3138       42.90        8.87              46.93  43.84  …  Wyoming 43.84

      long_y        date cases  state_code_y  deaths population_density  \
0     -86.64  2021-02-14  8.70            AL      84              91.32
1     -87.72  2021-02-14  9.86            AL     252             102.67
2     -85.39  2021-02-14  7.62            AL      48              28.50
3     -87.13  2021-02-14  7.78            AL      57              35.98
4     -86.57  2021-02-14  8.69            AL     121              88.60
…        …           …    …             …     …                 …
3134 -108.88  2021-02-14  8.19            WY      34               4.21
3135 -110.59  2021-02-14  8.09            WY       8               5.46
3136 -110.55  2021-02-14  7.60            WY      12               9.87
3137 -107.68  2021-02-14  6.77            WY      26               3.62
3138 -104.57  2021-02-14  6.42            WY       5               2.96

      case_ratio log_popl_density
0           0.11             4.51
1           0.09             4.63
2           0.08             3.35
3           0.11             3.58
4           0.10             4.48
…             …                …
3134        0.08             1.44
3135        0.14             1.70
3136        0.10             2.29
3137        0.11             1.29
3138        0.09             1.08

[3065 rows x 24 columns]
```

[3]:
```
   FID  GEOID         NAME          Co  St  Pop18              CoSt  \
0    1   1059     Franklin    Franklin  AL  31844     Franklin County
1    2  13111       Fannin      Fannin  GA  26644       Fannin County
2    3  19109      Kossuth     Kossuth  IA  15201      Kossuth County
3    4  40115       Ottawa      Ottawa  OK  31795       Ottawa County
4    5  42115  Susquehanna  Susquehanna  PA  42315  Susquehanna County

   UnwelPct  pct65pls  Staffed  …  F_ICUBeds  F_BedsPC   F_ICUPC  F_StaffPC  \
0     22.41     16.47       74  …          7    254.75  4,549.14     430.32
1     15.89     28.73       50  …          5    532.88  5,328.80     532.88
2     11.88     23.35       24  …          0    608.04      0.00     633.37
3     22.96     18.12       94  …          9    271.75  3,532.78     338.24
4     15.15     22.92       50  …          4    846.30 10,578.75     846.30

   BedsPC      ICUPC  StaffPC  FoodInsc  SHAPE_Length  SHAPE_Area
0  254.75   4,549.14   430.32     13.00          1.78        0.16
1  532.88   5,328.80   532.88     11.40          1.77        0.10
2  608.04       0.00   633.37      9.80          2.13        0.28
3  271.75   3,532.78   338.24     17.20          1.48        0.13
4  846.30  10,578.75   846.30     11.40          2.03        0.23

[5 rows x 25 columns]
```

[3]:
```
fips                  False
county_x              False
state_x               False
state_code_x          False
male                  False
female                False
median_age            False
population            False
female_percentage     False
lat_x                 False
long_x                False
Areaname              False
LND010200D            False
county_y              False
state_y               False
lat_y                 False
long_y                False
date                  False
cases                 False
state_code_y          False
deaths                False
population_density    False
case_ratio            False
log_popl_density      False
```

```
CoSt                  True
St                    True
ICUBeds               True
BedsPC                True
StaffPC               True
FoodInsc              True
dtype: bool

      fips          county_x  state_x state_code_x  male  female  \
0     1001    Autauga County  Alabama           AL 10.20   10.25
1     1003    Baldwin County  Alabama           AL 11.52   11.58
2     1005    Barbour County  Alabama           AL  9.52    9.40
3     1007       Bibb County  Alabama           AL  9.41    9.25
4     1009     Blount County  Alabama           AL 10.26   10.28
…     …                  …      …          …   …     …
3060  56037  Sweetwater County  Wyoming           WY 10.04    9.96
3061  56039       Teton County  Wyoming           WY  9.39    9.32
3062  56041       Uinta County  Wyoming           WY  9.26    9.22
3063  56043    Washakie County  Wyoming           WY  8.33    8.29
3064  56045      Weston County  Wyoming           WY  8.23    8.11

      median_age  population  female_percentage  lat_x  …  deaths  \
0          37.80       10.92              51.32  32.53  …      84
1          42.80       12.25              51.38  30.73  …     252
2          39.90       10.16              46.87  31.87  …      48
3          39.90       10.02              46.06  33.00  …      57
4          40.80       10.96              50.67  33.98  …     121
…            …         …                  …      …  …       …
3060       34.60       10.69              48.13  41.66  …      34
3061       39.30       10.05              48.35  43.93  …       8
3062       35.50        9.93              49.03  41.29  …      12
3063       43.50        9.00              49.11  43.90  …      26
3064       42.90        8.87              46.93  43.84  …       5

     population_density  case_ratio log_popl_density               CoSt  St  \
0                 91.32        0.11             4.51     Autauga County  AL
1                102.67        0.09             4.63     Baldwin County  AL
2                 28.50        0.08             3.35     Barbour County  AL
3                 35.98        0.11             3.58        Bibb County  AL
4                 88.60        0.10             4.48      Blount County  AL
…                   …         …               …                …  ..
3060               4.21        0.08             1.44  Sweetwater County  WY
3061               5.46        0.14             1.70       Teton County  WY
3062               9.87        0.10             2.29       Uinta County  WY
3063               3.62        0.11             1.29    Washakie County  WY
3064               2.96        0.09             1.08      Weston County  WY
```

```
      ICUBeds BedsPC  StaffPC FoodInsc
0        1.57   6.51     6.94    13.40
1        2.58   6.44     6.59    12.30
2        1.50   5.89     6.79    23.20
3        0.00   6.49     6.83    15.80
4        1.57   7.28     7.75    11.00
…         …     …       …       …
3060     1.78   5.99     6.43    11.10
3061     1.57   6.20     6.20     9.90
3062     1.57   6.24     6.51    14.10
3063     0.00   6.15     6.15    12.00
3064     0.00   6.43     6.43    13.80

[3018 rows x 30 columns]
```

[4]:
```
data_final = pd.merge(data_final,vote_data_dem ,
 ↪left_on=['county_x','state_x'], right_on = ['county','state'], how='inner')
data_final.head()
data_final.isna().sum()
```

[4]:
```
   fips         county_x  state_x state_code_x  male  female  median_age  \
0  1001  Autauga County  Alabama           AL 10.20   10.25       37.80
1  1003  Baldwin County  Alabama           AL 11.52   11.58       42.80
2  1005  Barbour County  Alabama           AL  9.52    9.40       39.90
3  1007     Bibb County  Alabama           AL  9.41    9.25       39.90
4  1009   Blount County  Alabama           AL 10.26   10.28       40.80

   population  female_percentage  lat_x  …  StaffPC FoodInsc    state  \
0       10.92              51.32  32.53  …     6.94    13.40  Alabama
1       12.25              51.38  30.73  …     6.59    12.30  Alabama
2       10.16              46.87  31.87  …     6.79    23.20  Alabama
3       10.02              46.06  33.00  …     6.83    15.80  Alabama
4       10.96              50.67  33.98  …     7.75    11.00  Alabama

           county  candidate  party  total_votes    won    votes DEM_fraction
0  Autauga County  Joe Biden    DEM         7503  False    27770         0.27
1  Baldwin County  Joe Biden    DEM        24578  False   109679         0.22
2  Barbour County  Joe Biden    DEM         4816  False    10518         0.46
3     Bibb County  Joe Biden    DEM         1986  False     9595         0.21
4   Blount County  Joe Biden    DEM         2640  False    27588         0.10

[5 rows x 38 columns]
```

[4]:
```
fips                    0
county_x                0
state_x                 0
state_code_x            0
```

```
male                  0
female                0
median_age            0
population            0
female_percentage     0
lat_x                 0
long_x                0
Areaname              0
LND010200D            0
county_y              0
state_y               0
lat_y                 0
long_y                0
date                  0
cases                 0
state_code_y          0
deaths                0
population_density    0
case_ratio            0
log_popl_density      0
CoSt                  0
St                    0
ICUBeds               0
BedsPC                0
StaffPC               0
FoodInsc              0
state                 0
county                0
candidate             0
party                 0
total_votes           0
won                   0
votes                 0
DEM_fraction          0
dtype: int64
```

[5]:
```
data_final.to_csv('finaldata.csv')
vote_data_dem.to_csv('vote.csv')
```

[6]:
```
print("dem_draction mean: ",data_final['DEM_fraction'].mean())
print("dem_draction std: ",data_final['DEM_fraction'].std(),"\n")
print("lat mean: ",data_final['lat_x'].mean())
print("lat std: ",data_final['lat_x'].std(),"\n")
print("long mean: ",data_final['long_x'].mean())
print("long std: ",data_final['long_x'].std(),"\n")
print("log-cases mean: ",data_final['cases'].mean())
print("log-cases std: ",data_final['cases'].std(),"\n")
```

```python
print("log-male mean: ",data_final['male'].mean())
print("log-male std: ",data_final['male'].std(),"\n")
print("log-female mean: ",data_final['female'].mean())
print("log-female std: ",data_final['female'].std(),"\n")
print("median age mean: ",data_final['median_age'].mean())
print("median age std: ",data_final['median_age'].std(),"\n")
print("log-popl mean: ",data_final['population'].mean())
print("log-popl std: ",data_final['population'].std(),"\n")
print("famale percent mean: ",data_final['female_percentage'].mean())
print("famale percent std: ",data_final['female_percentage'].std(),"\n")
print("landarea mean: ",data_final['LND010200D'].mean())
print("landarea std: ",data_final['LND010200D'].std(),"\n")
print("log-popl-density mean: ",data_final['log_popl_density'].mean())
print("log-popl-density std: ",data_final['log_popl_density'].std(),"\n")
print("fourth_root_ICUbeds mean: ",data_final['ICUBeds'].mean())
print("fourth_root_ICUbeds std: ",data_final['ICUBeds'].std(),"\n")
print("log_BedsPC mean: ",data_final['BedsPC'].mean())
print("log_BedsPC std: ",data_final['BedsPC'].std(),"\n")
print("log_StaffPC mean: ",data_final['StaffPC'].mean())
print("log_StaffPC std: ",data_final['StaffPC'].std(),"\n")
print("FoodInsc mean: ",data_final['FoodInsc'].mean())
print("FoodInsc std: ",data_final['FoodInsc'].std(),"\n")
```

```
dem_draction mean:  0.3226172432069653
dem_draction std:  0.15146763111955566

lat mean:  38.16898000383613
lat std:  4.863662739182393

long mean:  -92.04365222777348
long std:  11.242335561791965

log-cases mean:  7.716188796109285
log-cases std:  1.4284738314227694

log-male mean:  9.505843792052362
log-male std:  1.382695324573225

log-female mean:  9.50237246599053
log-female std:  1.4050056346616568

median age mean:  41.392472024415035
median age std:  5.363516032699041

log-popl mean:  10.198415159837902
log-popl std:  1.3923910174789456
```

```
famale percent mean:  49.919112079179946
famale percent std:  2.376065301863634

landarea mean:  972.8838826720925
landarea std:  1249.1304886112187

log-popl-density mean:  3.6819763791293756
log-popl-density std:  1.6395156335162768

fourth_root_ICUbeds mean:  0.9172825428334056
fourth_root_ICUbeds std:  1.1092327950795027

log_BedsPC mean:  4.764812627656603
log_BedsPC std:  2.5823368157296005

log_StaffPC mean:  4.902102517129053
log_StaffPC std:  2.6573348541788446

FoodInsc mean:  13.740217022719582
FoodInsc std:  4.217487591102986
```

[7]:
```python
finaltest =␣
↪data_final[['cases','DEM_fraction','lat_x','long_x','male','female','median_age','populatio
finaltest.head()
```

[7]:

	cases	DEM_fraction	lat_x	long_x	male	female	median_age	population	\
0	8.70	0.27	32.53	-86.64	10.20	10.25	37.80	10.92	
1	9.86	0.22	30.73	-87.72	11.52	11.58	42.80	12.25	
2	7.62	0.46	31.87	-85.39	9.52	9.40	39.90	10.16	
3	7.78	0.21	33.00	-87.13	9.41	9.25	39.90	10.02	
4	8.69	0.10	33.98	-86.57	10.26	10.28	40.80	10.96	

	female_percentage	LND010200D	log_popl_density	ICUBeds	BedsPC	StaffPC	\
0	51.32	604.45	4.51	1.57	6.51	6.94	
1	51.38	2,026.93	4.63	2.58	6.44	6.59	
2	46.87	904.52	3.35	1.50	5.89	6.79	
3	46.06	626.16	3.58	0.00	6.49	6.83	
4	50.67	650.60	4.48	1.57	7.28	7.75	

	FoodInsc
0	13.40
1	12.30
2	23.20
3	15.80
4	11.00

[8]:
```
X_train, X_test, y_train, y_test = train_test_split(finaltest.drop(columns =
 ['cases'],axis = 1),finaltest['cases'], test_size=0.20, random_state = 2)
X_train
X_test
y_train
y_test

finaltest = StandardScaler()
X_train = pd.DataFrame(finaltest.fit_transform(X_train), columns = X_train.
 columns,index=X_train.index)
X_test = pd.DataFrame(finaltest.transform(X_test), columns = X_test.columns,
 index=X_test.index)
```

[8]:
```
      DEM_fraction  lat_x  long_x  male  female  median_age  population  \
2632          0.35  40.33 -111.17  9.65    9.61       33.40       10.33
1567          0.21  42.49  -96.87  7.97    7.96       42.00        8.66
2702          0.28  37.06  -80.71  9.75    9.75       46.30       10.44
2843          0.26  37.66  -80.86  8.70    8.86       48.20        9.47
369           0.26  32.71  -83.99  8.75    8.71       44.40        9.42
…              …     …      …     …      …        …           …
2514          0.19  34.07 -102.35  8.81    8.79       36.00        9.49
2347          0.14  36.14  -84.65  9.40    9.16       41.10        9.98
1608          0.23  40.13  -96.24  7.15    7.24       50.00        7.89
2541          0.11  31.50  -98.60  7.75    7.85       50.60        8.50
2575          0.22  32.11  -94.76 10.25   10.14       38.50       10.89

      female_percentage  LND010200D  log_popl_density  ICUBeds  BedsPC  \
2632              49.07    1,209.17              3.23     0.00    7.46
1567              49.81      482.72              2.48     0.00    0.00
2702              49.94      329.59              4.64     1.68    5.95
2843              53.96      367.72              3.57     0.00    6.33
369               48.97      326.45              3.63     0.00    0.00
…                   …          …                 …        …      …
2514              49.40    1,017.73              2.57     0.00    5.19
2347              44.20      522.40              3.72     0.00    0.00
1608              52.28      432.95              1.82     0.00    5.52
2541              52.49      749.89              1.88     0.00    0.00
2575              47.05      938.62              4.04     0.00    6.33

      StaffPC  FoodInsc
2632     7.46     11.50
1567     0.00     10.10
2702     5.95     11.00
2843     6.33     13.80
369      0.00     15.00
…       …        …
2514     5.77     13.20
```

```
2347     0.00     15.00
1608     5.09     13.70
2541     0.00     14.80
2575     7.18     19.30

[2359 rows x 14 columns]
```

[8]:

```
      DEM_fraction  lat_x  long_x  male  female  median_age  population  \
1581          0.31  40.87  -98.50 10.34   10.33       36.00       11.02
994           0.17  37.59  -83.72  8.16    8.09       44.10        8.82
456           0.20  34.55  -83.29  9.37    9.54       41.50       10.15
596           0.32  41.03  -89.34  8.68    8.69       46.30        9.38
2234          0.31  43.67  -98.15  9.21    9.20       38.80        9.90
…                …      …       …     …       …           …           …
1948          0.30  40.39  -80.76 10.39   10.45       44.80       11.11
1127          0.40  38.21  -75.33 10.14   10.18       50.10       10.85
2654          0.42  39.11  -78.00  8.90    8.86       46.60        9.57
2392          0.26  30.27  -98.40  8.66    8.61       49.90        9.33
202           0.38  39.27 -121.35 10.56   10.52       32.50       11.23

      female_percentage  LND010200D  log_popl_density  ICUBeds  BedsPC  \
1581              49.74      552.22              4.71     2.00    5.98
994               48.14      211.22              3.46     0.00    0.00
456               54.18      184.23              4.94     1.57    5.62
596               50.20      398.52              3.39     0.00    0.00
2234              49.94      436.78              3.82     1.68    5.73
…                     …           …                 …        …       …
1948              51.43      410.87              5.09     1.86    5.58
1127              50.97      694.73              4.31     1.57    7.02
2654              48.95      178.19              4.39     0.00    0.00
2392              48.75      713.41              2.76     0.00    0.00
202               49.04      643.73              4.76     2.21    5.85

      StaffPC  FoodInsc
1581     6.11     10.20
994      0.00     21.50
456      5.88     14.20
596      0.00      9.50
2234     6.02     11.30
…           …         …
1948     5.98     16.50
1127     7.36     11.60
2654     0.00      8.10
2392     0.00     13.30
202      6.25     16.10

[590 rows x 14 columns]
```

[8]:
```
2632    8.35
1567    6.32
2702    7.72
2843    6.54
369     6.39
        …
2514    7.58
2347    7.71
1608    5.30
2541    6.41
2575    8.19
Name: cases, Length: 2359, dtype: float64
```

[8]:
```
1581    8.84
994     7.07
456     7.97
596     6.71
2234    7.98
        …
1948    8.43
1127    8.07
2654    6.59
2392    6.50
202     8.63
Name: cases, Length: 590, dtype: float64
```

[9]:
```
#model_1 with linear regression
model1 = LinearRegression(fit_intercept = True)
model1.fit(X_train, y_train)

#model1.score(X_test, y_test)
model1.score(X_train, y_train)
model1.coef_
model1.intercept_

test_output1 = pd.DataFrame(model1.predict(X_test), index = X_test.index,␣
↪columns = ['predict LR log_case'])
test_output1.head()

test_output1 = test_output1.merge(y_test, left_index = True, right_index = True)
test_output1.head()
mean_absolute_error = abs(test_output1['predict LR log_case'] -␣
↪test_output1['cases']).mean()
print('Mean absolute error is ')
print(mean_absolute_error)
print('Fraction MAE is ')
print(mean_absolute_error / test_output1['cases'].mean())
```

[9]:
```
LinearRegression()
```

[9]:
```
0.9475409620160818
```

[9]:
```
array([-6.63616590e-02, -1.93380952e-02,  9.04713449e-02, -4.59761729e+00,
       -5.92762931e+00, -1.67850017e-01,  1.18685674e+01,  5.00778762e-02,
       -1.62878826e-02, -7.07644408e-02,  5.12765671e-03,  9.25101505e-03,
        4.13306688e-02, -1.54885762e-02])
```

[9]:
```
7.704374445314362
```

[9]:

	predict LR log_case
1581	8.62
994	6.30
456	7.81
596	6.70
2234	7.46

[9]:

	predict LR log_case	cases
1581	8.62	8.84
994	6.30	7.07
456	7.81	7.97
596	6.70	6.71
2234	7.46	7.98

```
Mean absolute error is
0.23230412410523965
Fraction MAE is
0.029922886992130117
```

[10]:
```
model2 = Lasso(fit_intercept = True, alpha = 0.05)

model2.fit(X_train, y_train)

model2.score(X_train, y_train)

model2.coef_ # this is beta 1, the slope of the regression function

model2.intercept_ # this is beta 0

test_output2 = pd.DataFrame(model2.predict(X_test), index = X_test.index,␣
 ↪columns = ['predict Lasso log_case'])
test_output2.head()

test_output2 = test_output2.merge(y_test, left_index = True, right_index = True)
test_output2.head()
```

```
mean_absolute_error = abs(test_output2['predict Lasso log_case'] -␣
 ↪test_output2['cases']).mean()
print('Mean absolute error is ')
print(mean_absolute_error)
print('Fraction MAE is ')
print(mean_absolute_error / test_output2['cases'].mean())
```

[10]:
```
Lasso(alpha=0.05)
```

[10]:
```
0.9416591618432251
```

[10]:
```
array([-0.        , -0.        ,  0.03452448,  1.04896106,  0.22070689,
       -0.1193119 ,  0.00150945,  0.        , -0.        ,  0.        ,
        0.        ,  0.        ,  0.0151458 , -0.        ])
```

[10]:
```
7.704374445314363
```

[10]:

	predict Lasso log_case
1581	8.58
994	6.41
456	7.65
596	6.83
2234	7.48

[10]:

	predict Lasso log_case	cases
1581	8.58	8.84
994	6.41	7.07
456	7.65	7.97
596	6.83	6.71
2234	7.48	7.98

```
Mean absolute error is
0.2381560514876638
Fraction MAE is
0.030676668537872723
```

[11]:
```
model3 = BaggingRegressor(random_state=2, max_samples = 10)

model3 = model3.fit(X_train, y_train)
model1.score(X_train, y_train)
```

[11]:
```
0.9475409620160818
```

[12]:
```
test_output3 = pd.DataFrame(model3.predict(X_test), index = X_test.index,␣
 ↪columns = ['predict Bagging log_case'])
test_output3.head()
```

```
test_output3 = test_output3.merge(y_test, left_index = True, right_index = True)
test_output3.head()
mean_absolute_error = abs(test_output3['predict Bagging log_case'] -␣
 ↪test_output3['cases']).mean()
print('Mean absolute error is ')
print(mean_absolute_error)
print('Fraction MAE is ')
print(mean_absolute_error / test_output3['cases'].mean())
```

[12]:

	predict Bagging log_case
1581	8.27
994	6.88
456	7.65
596	7.00
2234	7.43

[12]:

	predict Bagging log_case	cases
1581	8.27	8.84
994	6.88	7.07
456	7.65	7.97
596	7.00	6.71
2234	7.43	7.98

```
Mean absolute error is
0.5240060559801898
Fraction MAE is
0.06749675261547958
```

[13]:

```
model4 = RandomForestRegressor(random_state=2, min_samples_leaf = 3,␣
 ↪max_features = "sqrt")

model4 = model4.fit(X_train, y_train)

model4.score(X_train, y_train)

test_output4 = pd.DataFrame(model4.predict(X_test), index = X_test.index,␣
 ↪columns = ['predict RF log_case'])
test_output4.head()

test_output4 = test_output4.merge(y_test, left_index = True, right_index = True)
test_output4.head()
mean_absolute_error = abs(test_output4['predict RF log_case'] -␣
 ↪test_output4['cases']).mean()
print('Mean absolute error is ')
print(mean_absolute_error)
print('Fraction MAE is ')
print(mean_absolute_error / test_output4['cases'].mean())
```

[13]: 0.9819210229117767

[13]:
```
      predict RF log_case
1581                 8.61
994                  6.26
456                  7.73
596                  6.81
2234                 7.61
```

[13]:
```
      predict RF log_case  cases
1581                 8.61   8.84
994                  6.26   7.07
456                  7.73   7.97
596                  6.81   6.71
2234                 7.61   7.98
```

```
Mean absolute error is
0.21110661563252212
Fraction MAE is
0.027192454835632984
```

[14]:
```
model5 = GradientBoostingRegressor(random_state=2, min_samples_split = 5,␣
 ↪min_samples_leaf = 3)

model5 = model5.fit(X_train, y_train)

model5.score(X_train,y_train)
```

[14]: 0.9790725269242372

[15]:
```
test_output5 = pd.DataFrame(model5.predict(X_test), index = X_test.index,␣
 ↪columns = ['predict Gard Boos log_case'])
test_output5.head()

test_output5 = test_output5.merge(y_test, left_index = True, right_index = True)
test_output5.head()
mean_absolute_error = abs(test_output5['predict Gard Boos log_case'] -␣
 ↪test_output5['cases']).mean()
print('Mean absolute error is ')
print(mean_absolute_error)
print('Fraction MAE is ')
print(mean_absolute_error / test_output5['cases'].mean())
```

[15]:
```
      predict Gard Boos log_case
1581                        8.70
994                         6.27
456                         7.81
```

```
596                         6.82
2234                        7.68
```

[15]:

```
      predict Gard Boos log_case  cases
1581                        8.70   8.84
994                         6.27   7.07
456                         7.81   7.97
596                         6.82   6.71
2234                        7.68   7.98
```

```
Mean absolute error is
0.19108585673046655
Fraction MAE is
0.024613598741578046
```

[16]:

```python
from functools import reduce
alloutput = [test_output1,test_output2,test_output3,test_output4,test_output5]
predict_df = reduce(lambda  left,right: pd.merge(left,right,on=['cases'],
                                            how='inner'), alloutput)
predict_df = predict_df[['cases','predict LR log_case','predict Lasso log_case',
                         'predict Bagging log_case','predict RF␣
 ↪log_case','predict Gard Boos log_case']]
predict_df.head()
```

[16]:

```
   cases  predict LR log_case  predict Lasso log_case  \
0   8.84                 8.62                    8.58
1   7.07                 6.30                    6.41
2   7.97                 7.81                    7.65
3   6.71                 6.70                    6.83
4   7.98                 7.46                    7.48

   predict Bagging log_case  predict RF log_case  predict Gard Boos log_case
0                      8.27                 8.61                        8.70
1                      6.88                 6.26                        6.27
2                      7.65                 7.73                        7.81
3                      7.00                 6.81                        6.82
4                      7.43                 7.61                        7.68
```

[17]:

```python
read_pred = predict_df.copy()
read_pred['predict LR log_case'] = np.exp(read_pred['predict LR log_case'])
read_pred['predict Lasso log_case'] = np.exp(read_pred['predict Lasso␣
 ↪log_case'])
read_pred['predict Bagging log_case'] = np.exp(read_pred['predict Bagging␣
 ↪log_case'])
read_pred['predict RF log_case'] = np.exp(read_pred['predict RF log_case'])
read_pred['predict Gard Boos log_case'] = np.exp(read_pred['predict Gard Boos␣
 ↪log_case'])
```

```
read_pred.head()


mean_absolute_error = abs(read_pred['predict LR log_case'] -␣
 ↪read_pred['cases']).mean()
print('LR Mean absolute error is ')
print(mean_absolute_error)
print('LR ratio MAE is ')
print(mean_absolute_error / read_pred['cases'].mean())

mean_absolute_error = abs(read_pred['predict Lasso log_case'] -␣
 ↪read_pred['cases']).mean()
print('Lasso Mean absolute error is ')
print(mean_absolute_error)
print('Lasso ratio MAE is ')
print(mean_absolute_error / read_pred['cases'].mean())

mean_absolute_error = abs(read_pred['predict Bagging log_case'] -␣
 ↪read_pred['cases']).mean()
print('Bagging Mean absolute error is ')
print(mean_absolute_error)
print('Bagging ratio MAE is ')
print(mean_absolute_error / read_pred['cases'].mean())

mean_absolute_error = abs(read_pred['predict RF log_case'] -␣
 ↪read_pred['cases']).mean()
print('RF Mean absolute error is ')
print(mean_absolute_error)
print('RF ratio MAE is ')
print(mean_absolute_error / read_pred['cases'].mean())

mean_absolute_error = abs(read_pred['predict Gard Boos log_case'] -␣
 ↪read_pred['cases']).mean()
print('GB Mean absolute error is ')
print(mean_absolute_error)
print('GB ratio MAE is ')
print(mean_absolute_error / read_pred['cases'].mean())
```

[17]:

```
   cases  predict LR log_case  predict Lasso log_case  \
0   8.84             5,556.92                5,312.88
1   7.07               543.58                  606.40
2   7.97             2,468.50                2,094.99
3   6.71               811.34                  923.31
4   7.98             1,737.72                1,779.16
```

```
   predict Bagging log_case  predict RF log_case  predict Gard Boos log_case
0                  3,900.56             5,480.41                    6,030.12
1                    975.54               525.74                      527.50
2                  2,106.49             2,274.08                    2,475.86
3                  1,100.77               903.33                      915.45
4                  1,683.11             2,024.72                    2,159.12
```

[17]:

```
      cases  predict LR log_case  predict Lasso log_case  \
0      8.84                 8.62                    8.58
1      7.07                 6.30                    6.41
2      7.97                 7.81                    7.65
3      6.71                 6.70                    6.83
4      7.98                 7.46                    7.48
…      …                  …                     …
1455   8.43                 8.55                    8.49
1456   8.07                 8.21                    8.17
1457   6.59                 6.90                    7.05
1458   6.50                 6.55                    6.70
1459   8.63                 8.71                    8.78

      predict Bagging log_case  predict RF log_case  \
0                         8.27                 8.61
1                         6.88                 6.26
2                         7.65                 7.73
3                         7.00                 6.81
4                         7.43                 7.61
…                       …                    …
1455                      8.13                 8.56
1456                      8.14                 8.10
1457                      7.23                 6.80
1458                      6.77                 6.72
1459                      8.34                 8.64

      predict Gard Boos log_case
0                           8.70
1                           6.27
2                           7.81
3                           6.82
4                           7.68
…                          …
1455                        8.46
1456                        8.03
1457                        6.73
1458                        6.64
1459                        8.63

[1460 rows x 6 columns]
```

```
LR Mean absolute error is
3302.862809748904
LR ratio MAE is
456.3681064653794
Lasso Mean absolute error is
3200.329211059047
Lasso ratio MAE is
442.2006805144576
Bagging Mean absolute error is
2224.4940224014936
Bagging ratio MAE is
307.36611943143447
RF Mean absolute error is
3312.335424841095
RF ratio MAE is
457.6769708239409
GB Mean absolute error is
3362.73916571482
GB ratio MAE is
464.6414319918155
```

[18]:

```python
print("for number of cases")
sAMPLE_LR = abs(read_pred['cases'] - read_pred['predict LR log_case'] ).mean()
LR1 = (abs(read_pred['cases'])+abs(read_pred['predict LR log_case'])).mean()
print('LR sAMPLE',sAMPLE_LR/LR1)
sAMPLE_Lasso = abs(read_pred['cases'] - read_pred['predict Lasso log_case'] ).
 ↪mean()
Lasso1 = (abs(read_pred['cases'])+abs(read_pred['predict Lasso log_case'])).
 ↪mean()
print('Lasso sAMPLE',sAMPLE_Lasso/Lasso1)
sAMPLE_Bagging = abs(read_pred['cases'] - read_pred['predict Bagging log_case']␣
 ↪).mean()
Bagging1 = (abs(read_pred['cases'])+abs(read_pred['predict Bagging log_case'])).
 ↪mean()
print('Bagging sAMPLE',sAMPLE_Bagging/Bagging1)
sAMPLE_RF = abs(read_pred['cases'] - read_pred['predict RF log_case'] ).mean()
RF1 = (abs(read_pred['cases'])+abs(read_pred['predict RF log_case'])).mean()
print('RF sAMPLE',sAMPLE_RF/RF1)
sAMPLE_GB = abs(read_pred['cases'] - read_pred['predict Gard Boos log_case'] ).
 ↪mean()
GB1 = (abs(read_pred['cases'])+abs(read_pred['predict Gard Boos log_case'])).
 ↪mean()
print('GB sAMPLE',sAMPLE_GB/GB1)

print("\nfor number of log cases")
sAMPLE_LR = abs(predict_df['cases'] - predict_df['predict LR log_case'] ).mean()
LR1 = (abs(predict_df['cases'])+abs(predict_df['predict LR log_case'])).mean()
```

```python
print('LR sAMPLE',sAMPLE_LR/LR1)
sAMPLE_Lasso = abs(predict_df['cases'] - predict_df['predict Lasso log_case'] ).
 ↪mean()
Lasso1 = (abs(predict_df['cases'])+abs(predict_df['predict Lasso log_case'])).
 ↪mean()
print('Lasso sAMPLE',sAMPLE_Lasso/Lasso1)
sAMPLE_Bagging = abs(predict_df['cases'] - predict_df['predict Bagging␣
 ↪log_case'] ).mean()
Bagging1 = (abs(predict_df['cases'])+abs(predict_df['predict Bagging␣
 ↪log_case'])).mean()
print('Bagging sAMPLE',sAMPLE_Bagging/Bagging1)
sAMPLE_RF = abs(predict_df['cases'] - predict_df['predict RF log_case'] ).mean()
RF1 = (abs(predict_df['cases'])+abs(predict_df['predict RF log_case'])).mean()
print('RF sAMPLE',sAMPLE_RF/RF1)
sAMPLE_GB = abs(predict_df['cases'] - predict_df['predict Gard Boos log_case']␣
 ↪).mean()
GB1 = (abs(predict_df['cases'])+abs(predict_df['predict Gard Boos log_case'])).
 ↪mean()
print('GB sAMPLE',sAMPLE_GB/GB1)
```

```
for number of cases
LR sAMPLE 0.9956366946744619
Lasso sAMPLE 0.9954975305357755
Bagging sAMPLE 0.9935351679632073
RF sAMPLE 0.9956491185616423
GB sAMPLE 0.9957140539547416

for number of log cases
LR sAMPLE 0.014578333457077552
Lasso sAMPLE 0.016216942731567312
Bagging sAMPLE 0.034120573607639974
RF sAMPLE 0.014630249389655183
GB sAMPLE 0.013037112223976032
```

[ ]:

```python

```

[ ]:

```python

```