Analysis to Identify Underperforming Medicare Health Plans

January 6, 2021

0.0.1 Report

Purpose

Find the five worst plans for Medicare patients that warrant further investigation.

Notes

I used the contract_id column to filter plans because there are 267 unique contract_ids, but only 227 unique contract_names. Since I evaluated each row in the health plan dataframe separately, I chose to use the unique identifier contract_id.

There are no null values in the data. However, the data shows numerous suspicious values of 0.00 across the columns. This may warrant further research to check the data sources.

Approach

My approach to answering this question was to find the 5 plans that perform poorly year-over-year across all three measures: complaints, patients choosing to leave, and patients getting needed care. I opted to utilize data from all 4 years because consistently poor performance is a powerful indicator that the health plan is responsible for poor performance instead of external conditions. Also, this indicates a lack of positive change.

Process

I first found the health plans that remained at or above the 75th percentile each year for patient complaints. Next, I found the health plans that remained at or above the 75th percentile each year for patients choosing to leave the plan. Finally, I found the health plans that remained at or below the 25th percentile each year for patients getting the needed care. 7 health plans were in all three groups. In order to chisel the number to the 5 worst health plans, I decided to only include the health plans that remained at or above the 90th percentile for 2019 patient complaints. All other filters remained the same. I made this decision because a high ratio of patient complaints in 2019 indicates that the health plan's most recent performance is bad. Secondly, complaints often precede patients leaving a plan, so a high complaint ratio bodes poorly for the second measure of patients who choose to leave the plan.

Results

Based on this analysis, the top five plants to investigate further due to their consistently poor performance are:

Health Net of Arizona, Inc. (ID: H0351), Wellcare Health Plans of New Jersey, Inc. (ID: H0913), Harmony Health Plan, Inc. (ID: H1416), Human Insurance of Puerto Rico, Inc. (ID: H2029), and Vns Choice (ID: H5549).

The code is below for your reference.

0.0.2 Analysis

[1]:
```
#Import necessary packages
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
```

[2]:
```
#Create dataframe and perform preliminary observation
df = pd.read_csv('emendata_analysis_data.csv')
df.head()
```

[2]:
```
  contract_id                                      contract_name  \
0       H0104              Blue Cross And Blue Shield Of Alabama
1       H0154                                  Viva Health, Inc.
2       H0251    Unitedhealthcare Plan Of The River Valley, Inc.
3       H0294  Care Improvement Plus Wisconsin Insurance Company
4       H0302                                      Medisun, Inc.

   complaints_2016  complaints_2017  complaints_2018  complaints_2019  \
0             0.09             0.04             0.04             0.04
1             0.06             0.06             0.06             0.05
2             0.16             0.12             0.13             0.12
3             0.25             0.22             0.17             0.04
4             0.16             0.20             0.14             0.13

   leaving_2016  leaving_2017  leaving_2018  leaving_2019  care_2016  \
0          0.06          0.03          0.04          0.03       0.89
1          0.06          0.07          0.05          0.05       0.87
2          0.10          0.09          0.08          0.05       0.88
3          0.13          0.22          0.16          0.11       0.84
4          0.05          0.10          0.08          0.19       0.82

   care_2017  care_2018  care_2019
0       0.86       0.85       0.85
1       0.87       0.86       0.87
2       0.86       0.85       0.87
3       0.86       0.85       0.85
4       0.82       0.84       0.79
```

[3]:
```
#Find number of unique contract id's
df['contract_id'].unique().shape
```

[3]:
```
(267,)
```

[4]:
```
#Find number of unique contract names
df['contract_name'].unique().shape
```

[4]:

```
(229,)
```

[5]:

```
#Check for missing values
df.isnull().sum()
```

[5]:

```
contract_id        0
contract_name      0
complaints_2016    0
complaints_2017    0
complaints_2018    0
complaints_2019    0
leaving_2016       0
leaving_2017       0
leaving_2018       0
leaving_2019       0
care_2016          0
care_2017          0
care_2018          0
care_2019          0
dtype: int64
```

[6]:

```
#Check for suspicious values. Here this is defined as 0.0
df[df.eq(0.00).any(1)].head()
```

[6]:

```
   contract_id                                     contract_name  \
48       H1651              Medical Associates Health Plan, Inc.
59       H2224                          Senior Whole Health, Llc
70       H2416  Primewest Rural Mn Health Care Access Initiative
71       H2419                     South Country Health Alliance
72       H2422                              Healthpartners, Inc.

    complaints_2016  complaints_2017  complaints_2018  complaints_2019  \
48             0.00             0.00             0.00             0.01
59             0.00             0.01             0.04             0.05
70             0.04             0.00             0.08             0.00
71             0.05             0.00             0.00             0.05
72             0.00             0.00             0.00             0.00

    leaving_2016  leaving_2017  leaving_2018  leaving_2019  care_2016  \
48          0.01          0.01          0.01          0.01       0.88
59          0.04          0.05          0.04          0.03       0.84
70          0.02          0.01          0.02          0.02       0.87
71          0.01          0.02          0.02          0.02       0.88
72          0.03          0.02          0.02          0.02       0.89

    care_2017  care_2018  care_2019
48       0.87       0.88       0.88
```

```
59        0.84        0.80        0.81
70        0.88        0.88        0.86
71        0.84        0.86        0.88
72        0.87        0.83        0.86
```

[7]:

```
#Create lists that include the contract_id's that fall at or above the 75th␣
 ↪percentile for complaints for each year
complaints_19 = list(df[df['complaints_2019'] >=  df['complaints_2019'].
 ↪quantile(q = 0.90)].sort_values(by=['complaints_2019'], ascending =␣
 ↪False)['contract_id'].values)
complaints_18 = list(df[df['complaints_2018'] >=  df['complaints_2018'].
 ↪quantile(q = 0.75)].sort_values(by=['complaints_2018'], ascending =␣
 ↪False)['contract_id'].values)
complaints_17 = list(df[df['complaints_2017'] >=  df['complaints_2017'].
 ↪quantile(q = 0.75)].sort_values(by=['complaints_2017'], ascending =␣
 ↪False)['contract_id'].values)
complaints_16 = list(df[df['complaints_2016'] >=  df['complaints_2016'].
 ↪quantile(q = 0.75)].sort_values(by=['complaints_2016'], ascending =␣
 ↪False)['contract_id'].values)
```

[8]:

```
#Find plans that consistently perform in the worst 25% for their years for␣
 ↪2016-2019
complaints_bad_performers = []
for plan in complaints_19:
    if plan in complaints_18 and plan in complaints_17 and plan in␣
 ↪complaints_16:
        complaints_bad_performers.append(plan)


print(complaints_bad_performers)
```

```
['H6528', 'H0913', 'H2029', 'H1666', 'H8552', 'H5422', 'H8748', 'H5322',
'H5549', 'H1111', 'H3342', 'H3822', 'H1416', 'H7787', 'H0351', 'H6328', 'H1112']
```

[9]:

```
#Create lists that include the contract_id's that fall at or above the 75th␣
 ↪percentile for patients who leave plan for each year
leaving_19 = list(df[df['leaving_2019'] >=  df['leaving_2019'].quantile(q = 0.
 ↪75)].sort_values(by=['leaving_2019'], ascending = False)['contract_id'].
 ↪values)
leaving_18 = list(df[df['leaving_2018'] >=  df['leaving_2018'].quantile(q = 0.
 ↪75)].sort_values(by=['leaving_2018'], ascending = False)['contract_id'].
 ↪values)
leaving_17 = list(df[df['leaving_2017'] >=  df['leaving_2017'].quantile(q = 0.
 ↪75)].sort_values(by=['leaving_2017'], ascending = False)['contract_id'].
 ↪values)
```

```
leaving_16 = list(df[df['leaving_2016'] >=  df['leaving_2016'].quantile(q = 0.
 ↪75)].sort_values(by=['leaving_2016'], ascending = False)['contract_id'].
 ↪values)
```

[10]:
```
leaving_bad_performers = []
for plan in leaving_19:
    if plan in leaving_18 and plan in leaving_17 and plan in leaving_16:
        leaving_bad_performers.append(plan)

print(leaving_bad_performers)
```

```
['H8554', 'H5549', 'H2486', 'H0913', 'H3347', 'H4007', 'H5322', 'H5471',
'H4523', 'H0423', 'H1666', 'H5928', 'H5431', 'H5656', 'H5577', 'H4227', 'H1032',
'H1416', 'H0351', 'H3387', 'H1111', 'H7787', 'H5991', 'H5475', 'R6801', 'H7245',
'H4003', 'H1019', 'H3815', 'H5087', 'H2029', 'H1415', 'H3312']
```

[11]:
```
#Create lists that include the contract_id's that fall at or below the 25th␣
 ↪percentile percentage of patients who get needed care for each year
care_19 = list(df[df['care_2019'] <=  df['care_2019'].quantile(q = 0.25)].
 ↪sort_values(by=['care_2019'])['contract_id'].values)
care_18 = list(df[df['care_2018'] <=  df['care_2018'].quantile(q = 0.25)].
 ↪sort_values(by=['care_2018'])['contract_id'].values)
care_17 = list(df[df['care_2017'] <=  df['care_2017'].quantile(q = 0.25)].
 ↪sort_values(by=['care_2017'])['contract_id'].values)
care_16 = list(df[df['care_2016'] <=  df['care_2016'].quantile(q = 0.25)].
 ↪sort_values(by=['care_2016'])['contract_id'].values)
```

[12]:
```
care_bad_performers = []
for plan in care_19:
    if plan in care_18 and plan in care_17 and plan in care_16:
        care_bad_performers.append(plan)

print(care_bad_performers)
```

```
['H5608', 'H5928', 'H3204', 'H3206', 'H3328', 'H2174', 'H3379', 'H3387',
'H3822', 'H4346', 'H5087', 'H5587', 'H3330', 'H5810', 'H0545', 'H9082', 'H0423',
'H0351', 'H2593', 'H5746', 'H5823', 'H5826', 'H5991', 'H3347', 'H3251', 'H9207',
'H5656', 'H1415', 'H0571', 'H5475', 'H5433', 'H3815', 'R3175', 'H2241', 'H3307',
'H0321', 'H2491', 'H0712', 'H0913', 'H2029', 'H1416', 'H5008', 'H5430', 'H2108',
'H4527', 'H5549']
```

[13]:
```
all_bad_performers = []

for plan in complaints_bad_performers:
    if plan in leaving_bad_performers and plan in care_bad_performers:
        all_bad_performers.append(plan)

print(all_bad_performers)
```

```
['H0913', 'H2029', 'H5549', 'H1416', 'H0351']
```

```
[14]: df_filtered = df[df['contract_id'].isin(all_bad_performers)]
      df_filtered
```

```
[14]:     contract_id                              contract_name  complaints_2016  \
      7         H0351                Health Net Of Arizona, Inc.             0.22
      24        H0913  Wellcare Health Plans Of New Jersey, Inc.             0.22
      42        H1416                 Harmony Health Plan,  Inc.             0.28
      55        H2029      Humana Insurance Of Puerto Rico, Inc.             0.50
      204       H5549                                 Vns Choice             0.22

           complaints_2017  complaints_2018  complaints_2019  leaving_2016  \
      7               0.27             0.22             0.25          0.17
      24              1.23             0.22             0.64          0.19
      42              0.31             0.28             0.28          0.16
      55              0.81             0.52             0.52          0.18
      204             0.20             0.29             0.34          0.19

           leaving_2017  leaving_2018  leaving_2019  care_2016  care_2017  \
      7            0.38          0.20          0.15       0.74       0.76
      24           0.28          0.23          0.25       0.78       0.75
      42           0.17          0.16          0.15       0.75       0.74
      55           0.32          0.19          0.12       0.80       0.81
      204          0.22          0.18          0.31       0.75       0.80

           care_2018  care_2019
      7         0.79       0.78
      24        0.80       0.81
      42        0.80       0.81
      55        0.74       0.81
      204       0.81       0.82
```