Benjamin Spiegel

Medicare Capstone Milestone 2

10/09/2019

*Executive Summary*

This project used Gradient Boosting to recognize physicians who have much higher prescription drug costs compared to their predicted cost. The Coefficient of Determination using this algorithm was .664. Additionally the accuracy of the model is demonstrated in that 79.5% of physicians were no more than 40% above their predicted drug costs, while 88.2% were no more than 60% above their predicted costs.

*Description of The Data Set*

I obtained the data for my data set from cms.gov using two different csv files for the year 2016. The first file was titled ‘[medicare-physician-and-other-supplier-national-provider-identifier-npi-aggregate-report-calendar-year-2016.csv](http://localhost:8888/edit/Desktop/springboard/Springboard-20190719T193509Z-002/Springboard/Kagle_Datasets/Medicare_NPI/medicare-physician-and-other-supplier-national-provider-identifier-npi-aggregate-report-calendar-year-2016.csv).’ It included 55 different features for all Medicare contracted doctors. These features provided information ranging from professional background of physicians and on demographic data of each doctor’s patient population. This demographic information included percentage of patients for each doctor who fit into particular demographic groups (age, race, income), and each doctor’s percentage of patients with specific health designations and conditions. Because the files were CSV files I used the ‘pd.read\_csv()’ function.

The second file titled ‘ [Part D Prescriber PUF NPI Drug, CY2016](http://download.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Downloads/PartD_Prescriber_PUF_NPI_DRUG_16.zip)’ included drug prescription data for every medication that a doctor prescribed over the year 2016. Consolidating this data to the most useful features, I used pandas *iloc* function and pandas *drop* function to create a new dataframe, including only ‘total\_drug\_cost’ (the total cost of medicare prescription) ’total\_30\_day\_fill\_count’ (total number of monthly prescriptions filled), ‘specialty description’ of the doctors and each doctor’s ‘National Provider Identifier’ .

Using Pandas’ group by function I combined all data for different drugs into one row per doctor. By doing this, I now had data on each doctor’s total number of 30 day prescriptions written and total cost to Medicare of all medications written using Medicare’s Part D drug plan. After combing all of the rows from the Part D drug based data frame described above into one row for each doctor, I merged this dataset with the first described dataset using Panda’s function ‘dataframe.merge()’.

After merging the files, I created a dictionary with each medical specialty as a key and each specialty’s respective average ‘total\_drug\_cost’ as a value. By using the map function, I then created a feature that was equal to the mean prescription costs for doctors of each physician’s specialty. I created this feature as an initial method for recognizing and calculating the relationships between specialty and total drug costs.

|  |
| --- |
| drug\_avg = df\_fill.loc[:,['specialty\_description','total\_drug\_cost']]  drug\_avg = drug\_avg.groupby(by ='specialty\_description').median()  nested\_dict = drug\_avg['total\_drug\_cost']  nested\_dict = nested\_dict.to\_dict()  dr\_df['specialty\_description\_vector'] = dr\_df['specialty\_description'].map(nested\_dict) |

Although creating a feature equal to the mean annual drug cost of a doctor’s specialty was helpful in predicting a physician’s drug cost, I wanted to take it a step further and make sure the machine learning algorithm was able to handle the specialties as categorical data. As there were over 70 different specialties in the data, the top algorithms could not handle features with this many categorical variables. My solution to this was to filter the data so that only 8 specialties were included. I chose to use the 8 specialties with the most physicians who wrote over 3,000 prescriptions in a year. I did this because I wanted to choose specialties with the largest number of doctors with meaningful data, as opposed to simply the largest number of doctors who had written any prescription. This minimized working the effect of physicians who were, for purposes of this subset, just creating random noise due to their lack of prescription volume. It also emphasized the importance of prescriptions written as opposed to just sheer number of physicians.

Once I filtered the data to eight specialties, I then used Pandas to create dummy variables for each of the 8 specialties so that my algorithms would be able to recognize relationships between specialties and other variables. Below is the code used to implement the steps described above

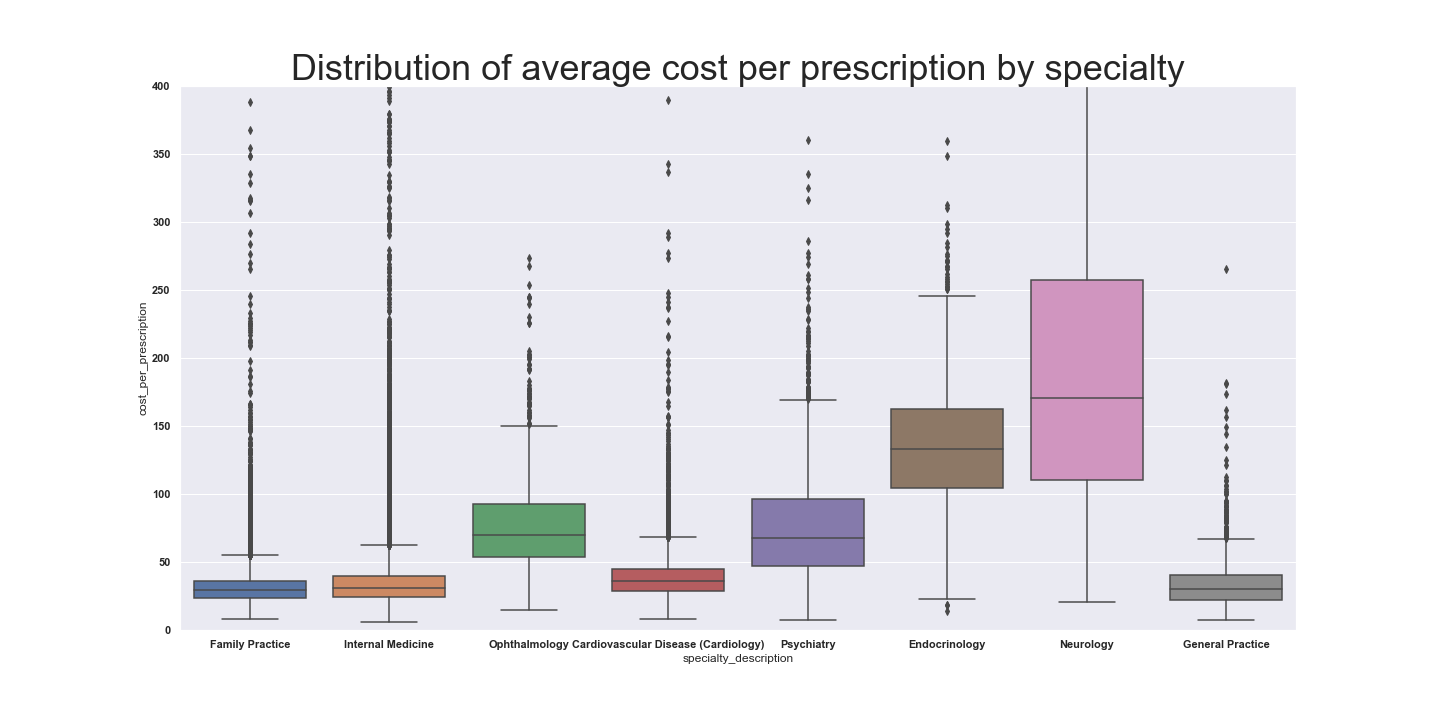
|  |
| --- |
| **from collections import Counter**  **spec\_df = df\_fill[df\_fill['total\_30\_day\_fill\_count']>=3000]**  **V\_count = Counter(spec\_df.specialty\_description).most\_common(8)**  **spec\_list = []**  **for object in V\_count:**  **spec\_list.append(object[0])**  **top\_spec = df\_fill[df\_fill.specialty\_description.isin(spec\_list)]**  **dummy\_df = pd.get\_dummies(top\_spec['specialty\_description'])**  **df\_fill = pd.concat([top\_spec,dummy\_df], axis = 1)** |

Next I dealt with missing values. I did this because of the inability of the machine learning algorithms to run data with na. In order to do so, I used the Pandas’ function ‘dropna()’. I created a threshold of 44 values as the minimum amount of values a row could have without being dropped. This decreased the DataFrame from 622705 to 397766 rows. For those rows that were not dropped, but still had na values, I rab Pandas’ ‘fillna’ function. I replaced each na value with the median value of it’s column. The code used is below:

Before using any machine learning on the dataset needed to replace any categorical values with numerical values. I did this for the feature ‘Gender of the Provider’. I created a dictionary (genderdict) where Male and Female were keys with values of 0 and 1 respectively. I then remapped the feature with the code “dataframe[‘Gender of the Provider’].map(genderdict)”.

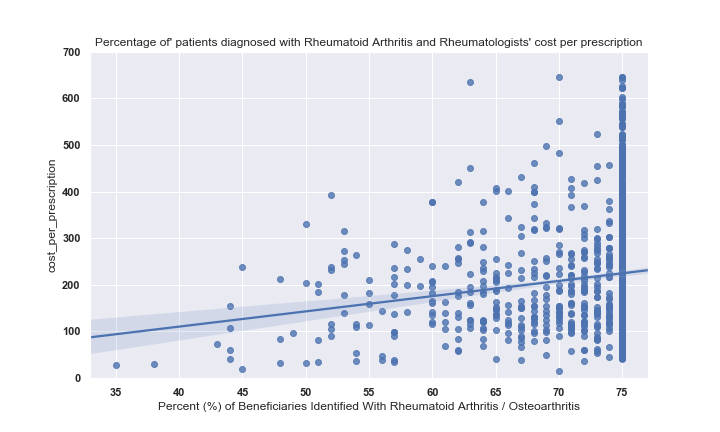
*Data Exploration*

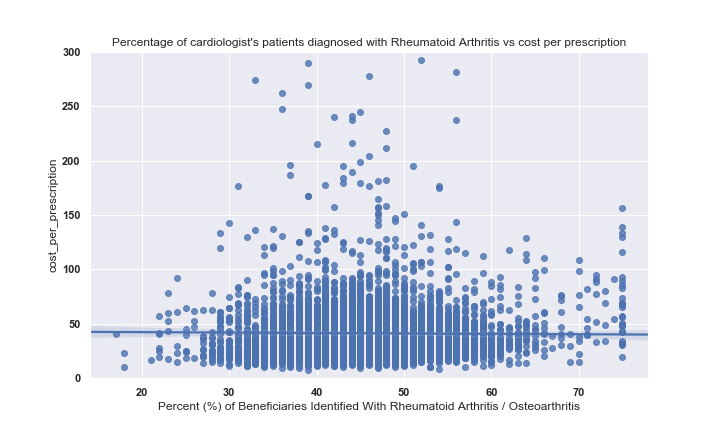
After obtaining and organizing the data, I looked at Data exploration. The first visualization I created of the data looked at how distribution of drug costs differed according to medical specialties.



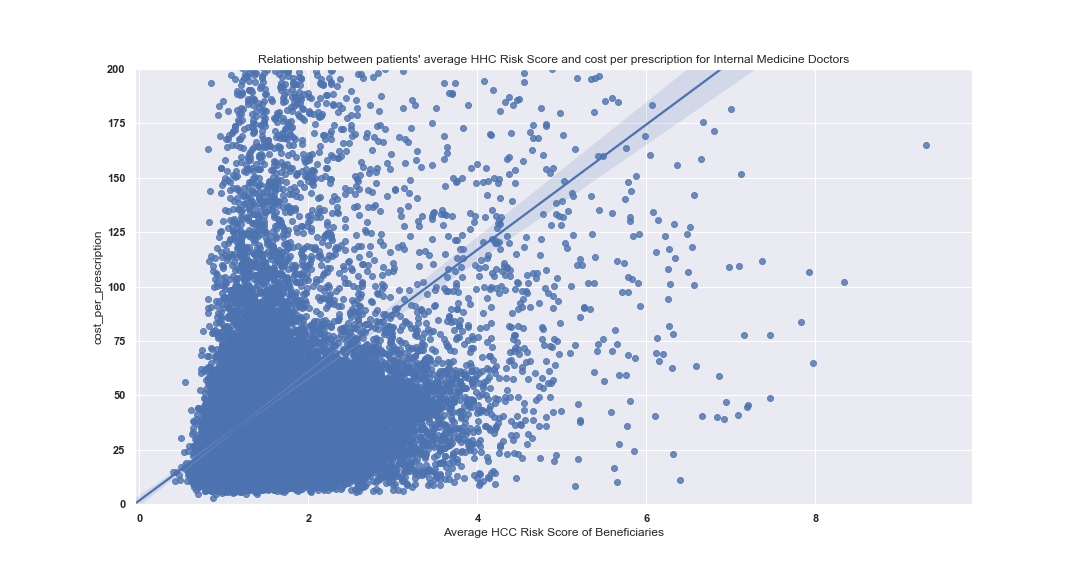
Looking at the plot above, one can see that a doctor’s medical specialty has a large bearing on the range of the average cost per prescription he/ she writes. Of the eight specialties in the plot, it can be seen that Neurology and Endocrinology have the highest average prescription costs. Similarly, it can be seen that Psychiatry and Ophthalmology are also quite high. On the other hand, primary care specialties and Cardiology had much lower average costs per prescription. This likely has to do with certain specialties dealing with illnesses that require more expensive brand name medications, medications that are often targeted to specific chronic illnesses.

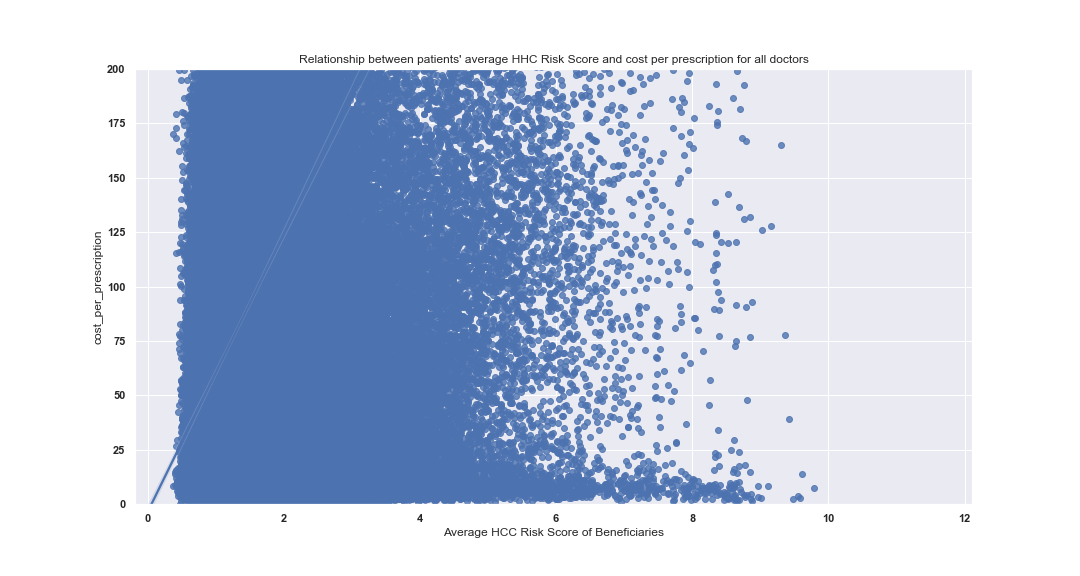
Next, I tried to look at the relationship between the percentage of physician’s patients with specific health conditions and their average cost per prescription. I found however that it was necessary to isolate the distinct specialties to find a correlation. The reason for this is likely because individuals with chronic illness go to specialists that relate to their illnesses. Drugs that physicians prescribe are related to the illnesses they treat and not the other illness the patient has. Therefore, the specific illnesses that specialists treat are correlated with the drugs they prescribe and their costs,. A supporting plot of this theory can be seen in the plots below showing how the percentage of patients with Rheumatoid Arthritis affect distinct specialties’ prescription costs differently. For instance, while the Percentage of patients with Rheumatoid Arthritis has a large effect on the average prescription costs of Rheumatologists, it has very little relation to the average prescription costs of Cardiologists, who do not treat Arthritis.





Finally, I looked at how ‘HCC score’ affects drug cost. HCC score is “a risk adjustment model that is used to calculate risk per individual on the basis of how cumulative chronic illnesses  predict future healthcare costs. Essentially specific chronic illnesses have specific HCC values depending on their average costs. Adding up these different chronic illness HCC values results in a specific HCC score. Physicians with Higher average HCC scores for their patient’s population were correlated with higher average costs. Of interest, family practice physician’s average prescription drug cost correlation with HCC is weaker than that of all doctors combined. One could presume this is because patients who are very sick are going to be getting much of their care and prescription from specialists. It also explains why the annual cost of many specialties are much higher than that of primary care physicians. Below are two charts demonstrating the relationships described above.

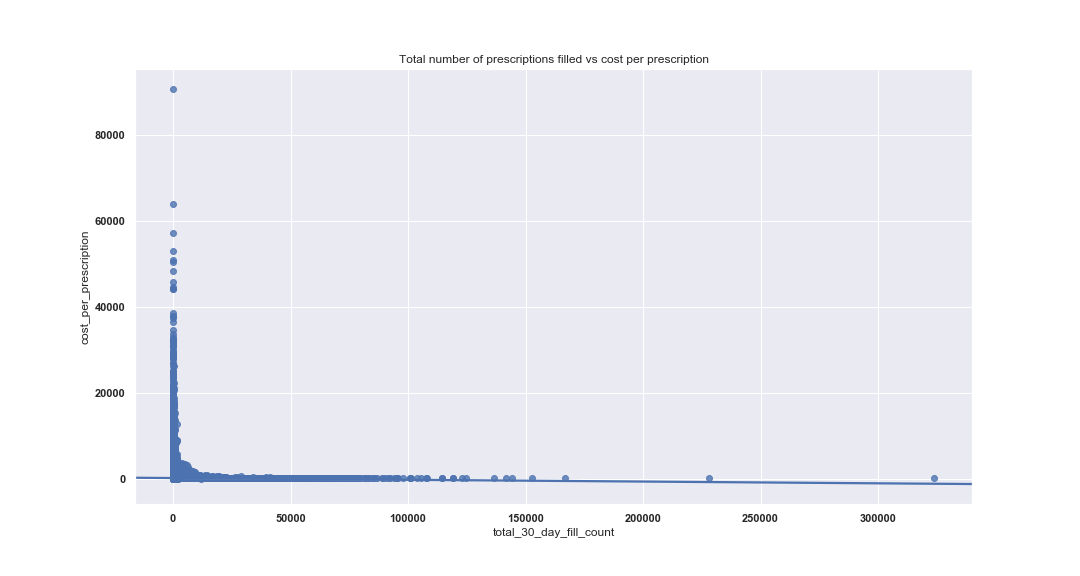




Supporting the theory above, one can see that the slope for cost per prescription is higher when comparing HCC risk score without filtering doctor types, while internal medicine (primary care physicians) had a much weaker slope.

I also found that to demonstrate the correlations of chronic illness to average cost per prescription, it was helpful to filter out doctors who barely prescribed any prescription. This could be a result of these doctors prescribing so little medications that just one or two expensive medications could dramatically increase the average costs of the medications they prescribe as a whole.

Additionally, a doctor prescribing so few medications is less important to the underlying problem of high drug costs because they prescribe so little and make up a small amount of overall drug costs. Regardless of the reason, it is clear that physicians who prescribed very little medications are more likely to have significantly higher average drug costs. Below is a plot demonstrating such. The plot below demonstrates the irregularity of physicians who do not prescribe many prescriptions in Medicare’s Part D drug plan.



*Results and In Depth Analysis*

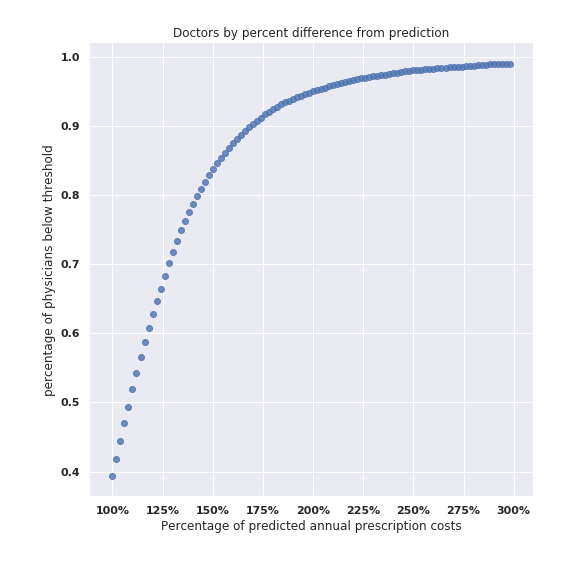
I made predictions with scikit learn’s RandomForestRegressor, and its’ GradientBoost. First I ran GridSearchCV to find the best hyperparameters for both algorithms. I then split the data into training and testing subsets. I did this using the code below:

|  |
| --- |
| X = all\_med.drop('total\_drug\_cost', axis = 1)  X = X\_initial.drop('National Provider Identifier', axis = 1)  y = all\_med.total\_drug\_cost  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = .20) |

Finally, I worked with both Random Forest and GradientBoostRegressor to make predictions for each doctor’s annual drug cost. I used the score() method to find out how accurate the models were. The score value for both of these regression models tell the Coefficient of Determination, which tells how much of the dependent variable’s variance the model is able to explain by the independent variables.

|  |
| --- |
| rf **=** RandomForestRegressor**(** n\_estimators **=** **200,** max\_features **=** **30,** max\_depth **=** **25)**  rf**.**fit**(**X\_train**,** y\_train**)**  rf\_y\_pred **=** rf**.**predict**(**X\_test**)**  rf\_SCORE **=** rf**.**score**(**X\_test**,** y\_test**)**  **print(**rf\_SCORE**)** |

After running both models the higher performing model was Gradient Boosting. It gave a Coefficient of Determination of .67, this is a reasonably accurate score. Some of the variability not represented in the Coefficient of Determination is likely caused by certain physicians tendency to prescribe expensive drugs at the recommendation of pharmaceutical representatives. In making the predictions, the Gradient Boosting model found the most important features were total number of monthly prescriptions filled (42.6%), the combined importance of all chronic diseases (15.1%), combined importance of features relating to specialty (12.9%) , the combined importance of factors relating to patient age (6.3%) and gender (3.3%) and the percentage of patients on Medicare alone or both Medicare and Medicaid (8.3%) The following plot shows the difference in predictions against the actual annual drug costs:



|  |  |
| --- | --- |
| **Percentage above predicted annual prescription costs** | **percentage of physicians below threshold** |
| 20% | 63.1% |
| 40% | 79.5% |
| 60% | 88.2% |
| 80% | 92.8% |
| 100% | 95.6% |

*Conclusion*

This project provided meaningful information to recognize physicians who are outliers in terms of their prescription habits. Industries that could apply the project’s conclusions:

1. Medicare advantage plans who can choose the doctors that do and do not participate in their networks in order to drive down drug costs.
2. Pharmaceutical companies releasing new brand name drugs to recognize the best doctors to target upon approval of their medications.

These results can obviously not be accepted as completely conclusive. There are variables that are not included in this dataset. For instance some physicians specialize in very uncommon diseases, diseases that are not measured by Medicare when looking at individual physician’s patient population. Often the least common diseases have the most expensive medications. Additionally certain specialties have sub specialties that may have a wide array of sub specialties. For instance, interventional, invasive and non-invasive are three cardiology subspecialties which causes cardiologist when isolated to have a lower coefficient of determination. However interested parties could flesh that out going forward. Regardless, this data is useful to recognize the outliers whose prescribing habits could be further fleshed out.