Data Science Final Report

crz19 and njl88

May 8, 2018

1 Introduction

In this project we explore what information can be gained from publicly available medical data sets. We primarily used the HCUP databases, though Medicare submissions by provider were also explored for feasibility. For Predictions we both implement our own K Nearest Neighbor algorithm and utilize sci-kit-learn's algorithm for K Nearest Neighbor regression to have two different ways of predicting Diagnoses.

1.1 The HCUP Database

HCUP is a federally funded database meant to further public Healthcare research. They offer a few free services and databases which we made use of but most of their data must be bought. Some of the databases they include are The National Inpatient Sample, The Kids' Inpatient Database and the State Inpatient Sample. Their free service is called HCUPnet and is an online query system.

1.2 Previous Work Done using HCUP

Previous work done by researchers using the HCUP databases has involved making predictions such as length of stay or mortality rate for individual patients. Researchers were able to do this because they had access to the full databases where each enounter is logged for individual patients. For public use, HCUP data is averaged over multiple patients (at least 10) for privacy reasons. Because of this, the work we will be doing will be very different in nature from previous work. This is because we will focus on what kind of information one can extract from a limited health care data set, instead of making actual relevant medical informatics predictions.

2 Data Collection

2.1 Interface

Our main source of data collection was through the HCUP databases. The user interface offered by HCUPnet for data access was simply a website. The user had to enter the location, year, database, type of data, specific diagnoses and more information, each on a separate page. This information was handled as a series of HTTP GET and POST requests. Since they offer data about each procedure/ diagnoses in each county it required a script to handle all possible permutations to download all possible data.

2.2 Format

We wrote a script in python which sends a series of HTTP Get and Post requests to https://hcupnet-archive.ahrq.gov/HCUPnet.jsp using the imported python Requests library. These requests terminate in an HTML Table for each county and diagnosis. The python library BeautifulSoup was used to transform each HTML table into a python object and parsed into a csv. The format of each row is: County, State, Diagnosis, Payer, number of discharges, rate of discharges, mean length of stay, aggregate number of days, number of inpatient days, mean cost per stay, aggregate costs for all stays, costs for inpatient stays.

2.3 Problems

One problem was that the HCUP server required the user to accept their terms of agreement. However, whenever this was tried through a script the server rejected it. Once a user accepted this once though, they could access data as many times as they wanted if the same Id was used. To get around this, the terms were accepted in a browser and the accepted Id was passed to the script for future use. As long as the session remained open, this was allowed by the server and they placed no limit on how many requests they could receive from one ID. However, the server did tend to timeout every few hours so the completed states had to be marked down and the script restarted. We downloaded both Diagnosis and Procedure data from every state and each state took at least an hour to run so in total the script ran for more than 2 days.

3 Data Processing

3.1 HCUP

We also had to deal with many irregularities within the data while transforming it from HTML to CSV. One of the problems was dealing with commas within the data itself. Since this was not to be interpreted as multiple columns, the commas were replaced with underscores using the python string.replace() function. Another irregularity that had to be removed was that many of the text

columns had a series of non-breaking spaces (nbsp) in the beginning used in HTML formatting. So for every column, we had to test if the beginning began with this character or an irrelevant number and erase these characters.

3.2 Medicare Provider Billing Codes

Though we did not have time to run testing on the Medicare CMS data set, we did create functions to map the different coding systems used to each other. We wanted to go from the more specific billing codes present in the Medicare CMS data to the more general procedure codes used in the HCUP data sets.

3.3 CCS Procedure Codes

CCS Procedure codes are a dictionary of codes from 1 to 244 which map to generalized procedures. For feature vectors that are based on each county's procedure distribution, we do a 1:1 mapping where the procedure number is the index of the vector, and the element itself is equal to the number of patients that had that procedure done in that county in one year. The CPT codes used in the Medicare data are grouped by similar procedures, which allows us to bundle up to 100 codes together to map to a single CCS code.

4 Visualization

To visualize the HCUP dataset, we used an online resource called OpenHeatMap which takes a csv as input and overlays the data onto a map of the United States. OpenHeatMap breaks the data values into three ranges and color codes each range. OpenHeatMap also allows hovering over each county to display the value for that county. FIPS codes were used as the unique location identifier for each county, which required us to convert HCUP's county name to its FIPS code. This was more difficult than origininally thought since it required us to convert each state name to its unique postal abbreviation before converting to fips code. Please see the attached vis.py file which was used to make these conversions. 5 OpenHeatMap maps were created by combining and averaging each HCUP row by county. One map for each of the following categories was created: Aggregate Costs For All Hospital Stays, Aggregate Number of Days in the Hospital, Costs For Inpatient Stays Per Capita, Number Of Inpatient Days Per 100000 Population, Rate Of Discharges Per 100000 Population. OpenHeatMap allows each map to be exported as an HTML iFrame tag so each map was put into one HTML file. This HTML file provides a javascript dropdown menu to select which category's map to view. Please see the attached map.html file to view these maps.

5 Interesting Results

To look for interesting patterns in the HCUP data we first sorted the frequency of each diagnosis and procedure to pick out the top 5 most common for each state. This turned out to not be very interesting because every state had roughly the same top 5. So we then filtered these to return Diagnoses and Procedures that were in the top five of only one state. This yielded much more interesting results. For example Hawaii is the only state to have 'Debridement of wound infection or burn' in its top 5 most common procedures. Also Illinois and California are the only states to have Schizophrenia as one of its top 5 most common medical diagnoses. We also looked at the differences in diagnoses and procedures depending on type of insurance used. These also yielded interesting and in a few cases very unexpected results. For instance the top 5 diagnoses for Medicare are Septicemia, Congestive heart failure, Osteoarthritis, Pneumonia and Cardiac dysrhythmias. Since the age of eligibility for Medicare is 65, we expected arthritis and Pneumonia to top the list so Septicemia being at the top came as a surprise. To see all results please see the attached interestingResults.txt file.

6 Predicting Top Diagnoses Based On County Information

6.1 Goals

In addition to the HCUP and Medicare data, we also downloaded US census data that provides population and income data about each county. Each row of this data has the population for a given county, age group, race, sex and origin. This was used to perform supervised learning on population data and common diagnoses per county. The goal was to input a feature for a new county that contains information about its population and return 3 diagnoses or procedures that should be extremely common in that county. This has many possible practical applications. For example, imagine a new hospital is being created in a county. This hospital may need to buy specialized equipment for certain procedures but it cannot afford every procedure's special equipment. Being able to predict procedures they will have to perform often based on population data can lead to prioritization of equipment. This can save the hospital money and also allow them to provide better and more specific care.

6.2 Implementation

To implement this we used the K Nearest Neighbors algorithm on feature vectors that we created. To create these feature vectors we mapped each possibility of county, age, race and sex to one feature with a value equal to the population of that subset. There were 2 sexes, 31 races and 18 age groups which led to a 1116 dimension feature vector for each county. To predict probable diagnoses and procedures for an input county's feature vector, the euclidean distance

between it and every other feature vector in the training set was calculated using numpy's linear algebra library. Then this list of distances was sorted and the top 5 nearest counties were chosen. Then we used the HCUP data to get the top 3 diagnoses and procedures for each of those counties. Then the top 3 most frequent were chosen out of this list of 15 diagnoses and procedures. These top 3 are the results and show 3 diagnoses or procedures that should be common to that county. See the attached featvecsknn.py for the exact code.

6.3 Validation

To test the accuracy of this method we divided our county feature vectors into two groups. For each vector in the validation set, we computed the 3 probable diagnoses and procedures and then compared these to that counties actual most frequent diagnoses and procedures. Accuracy was calculated by dividing correct number of predictions by the total number. We also tested how different subdivisions of the total data into training and test sets affects accuracy. Using various breakdowns from 50% of data to 90% the results show that the amount of training data is not an important factor for accuracy. We can conclude that the model requires very little training data to become accurate. See figure 2.

7 Determining Feasibility of Using HCUP Data for Prediction

In order to gain some information about the nature of the feature vectors we created from the distribution of procedures per county, we performed PCA dimensionality reduction to visualize the vectors. PCA reduction was performed using the sk-learn library, with feature vectors first being normalized. From Figures 3 and 4 we can see clustering based on payment type, which shows there to be some systemic differences in the feature vectors based on the population covered. Because of these differences, we decided to first implement our prediction models with the data partitioned by payment type.

8 Predicting Diagnoses Based on Procedure Information

When the KNN regression model from sk-learn was first run, we kept the data separated by payment populations of Medicare, Medicaid and Uninsured, however this gave a low accuracy rate. In order to increase the accuracy of prediction, the three payment populations were combined without differentiation to increase the size of the training set. This is shown in Figure 5. We can see the effect this had on increasing accuracy, even though the PCA clustering had shown there to be significant differences between the feature vectors for each payment type. When the payment populations were combined, the average accuracy rate more than doubled from around 30% to over 60%.

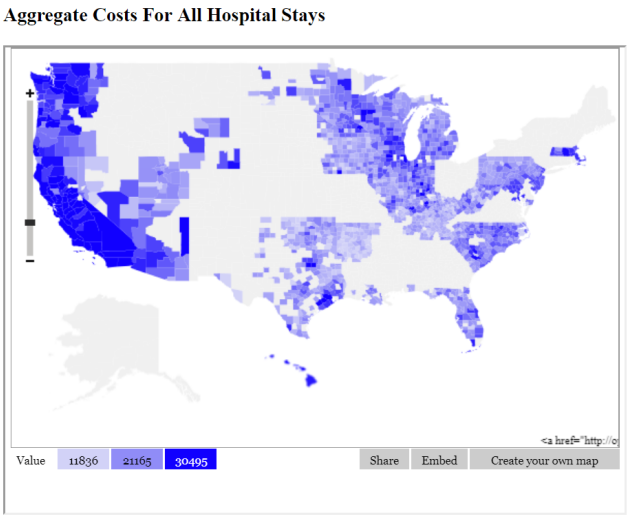

Figure 1: Example Graph

9 Conclusions

If we had had more time, we would have liked to finish creating the feature vectors for the Medicare PUF data in order to see how the distribution of procedures for the same county is different due to how the data was collected. In addition, we would have liked to use the Medicare data to make prediction on the HCUP data, since the Medicare dataset is so much larger and more specific.

10 Acknowledgements

Visualization: www.openheatmap.com
Python Libraries: sci-kit and numpy

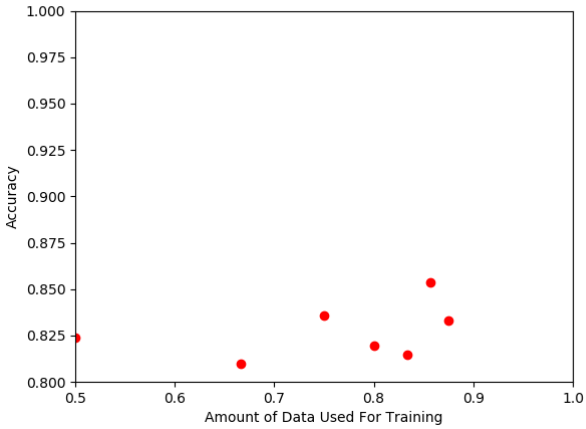

Figure 2: Accuracy Graph

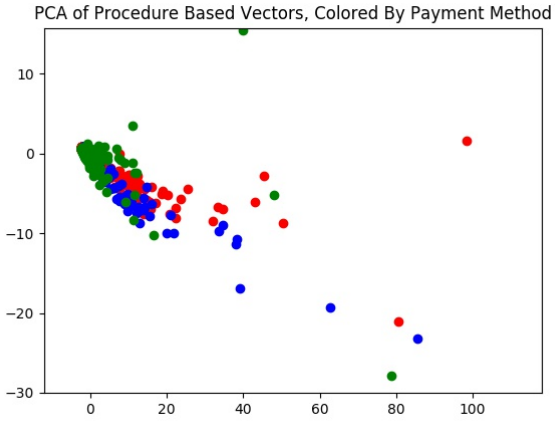

Figure 3: PCA on Procedure Feature Vectors, Medicare=Red, Medicaid=Blue, Uninsured=Green

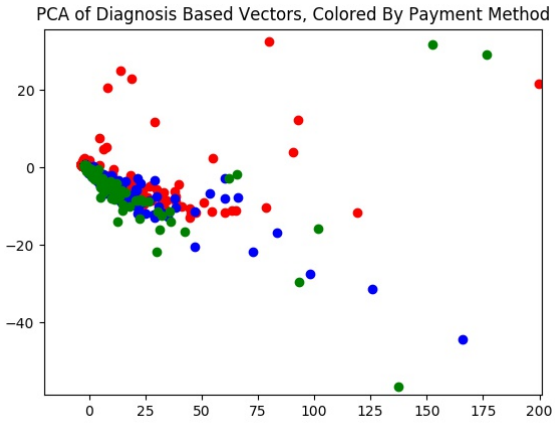

Figure 4: PCA on Diagnosis Feature Vectors, Medicare=Red, Medicaid=Blue, Uninsured=Green

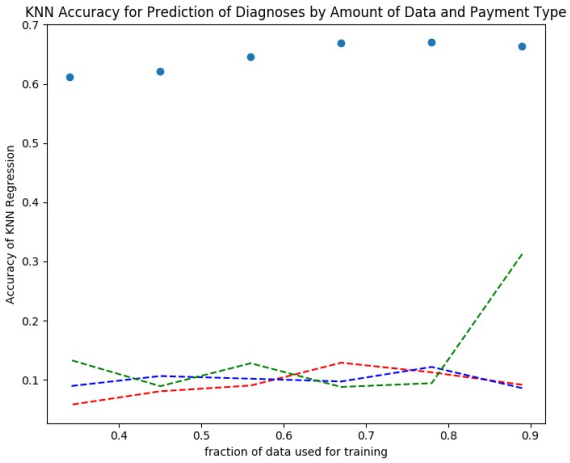

Figure 5: KNN Accuracy, Combined Payment=Dots,Medicare=Red, Medicaid=Blue, Uninsured=Green

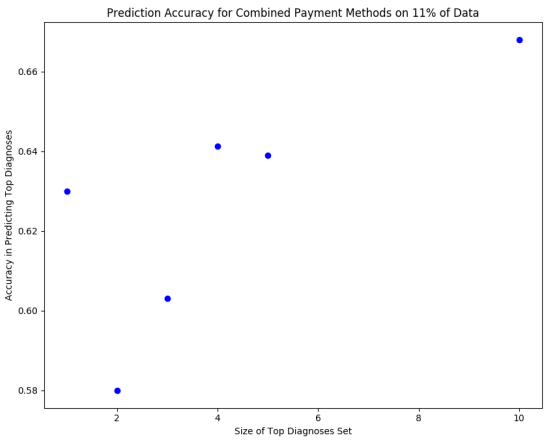

Figure 6: Accuracy By Size of Predicted Diagnoses Set