Diabetic Data Analysis

Upasana Yadav

uyadav01@nyit.edu

*Abstract*— In this project we are doing an analysis on a dataset of people affected by diabetes. The primary goal of the project was to analyze the data in order to learn how the various steps involved in making the data available are useful. This was done by studying the data to construct hidden patterns from the data such that we can find out how the attributes linked to others. Based on the data, we initially constructed multiple hypotheses based on how each attribute should be linked to the others. However not all of these turned out valid.

Given a set of past data and that a given item belongs to one of the given classes, the classification problem is the process of determining which class a new item belongs to [1]. Clustering is the process of finding relations between the attributes to find out which of them are closely related. For our analysis we focused on the classification problem. We constructed three hypotheses based on the data. The three hypotheses included determining the approximate duration of a patients stay in the hospital, the possible drugs to be administered and whether the patient will be readmitted to the hospital. After proposing the initial hypotheses, we went ahead to prove/disprove our hypotheses based on the data.

# Introduction

Diabetes has become a serious matter of concern for people all over the world. One in eleven adults has diabetes [2]. The number of people affected with diabetes is growing at a very high rate. The data set of people affected by diabetes is made available by <http://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008>. It is collected from 130 hospitals in the US over a duration of 10 years from 1999 to 2008. This data initially consisted of a total of 55 attributes and about 100000 instances. Every instance in the database gives information about the patient’s visit to the hospital, the condition under which he was admitted, his period of stay during the current visit, the diagnosis and tests performed, the drugs given, whether the patient was readmitted or not within 30 days.

We were particularly interested in this dataset as it contains a variety of parameters of the diabetic patient while they were admitted in the hospital. These parameters can be used to build up certain models to predict possible conditions of patients with similar conditions.

# Database Design

The diabetes data set that we have selected consists of 55 attributes and about one hundred thousand instances. We studied what each attribute represents and grouped them on a logical level. Since all the data existed in a single relation (except the mappings of ID’s), we decided to split the data into smaller relations to improve the readability of the database. The entire data was split into seven relations namely *treatment\_details*, *Patient, Admission\_time\_details, Medicine\_reference, Admission\_source, Admission\_type, Discharge\_disposition.*

Next we thought of verifying if the attributes containing ids had any duplicate values to determine candidate keys. The primary keys were then decided if the id’s consisted all unique values and did not contain null values. In the *treatment\_details* relation, *encounter\_id* was the primary key since it recorded every instance of a patient’s visit to the hospital as a new one. *Patient\_nbr* (patient number), which is a foreign key in the *treatment\_details* relation is dependent on the patients record in the *Patient* relation. The *patient\_nbr* is unique to a patient and does not change once it is assigned to a patient.

The *Patient* relation contains general demographic information about all the patients. Each patient is assigned a unique *patient\_nbr*. Information like the patient’s *race, gender, age, weight* was stored in this relation. The *medicine\_reference* relation consists of the various drugs that were administered to each patient on each encounter(visit). In this relation, the *encounter\_id* acts as a primary key since it is used to uniquely identify the medicines given to every patient each time he was admitted.

*Admission\_time\_details* relation contains the data representing the patient’s condition when he was admitted, discharged and whether the patient was readmitted or not. *Encounter\_id* is used as the primary key in order to uniquely identify the patient’s condition during each of his visits. The conditions while admitting and discharging are represented in form of numbers, description of which can be found in their individual relations. The information of what each *discharge\_disposition\_id* represents can be located in *Discharge\_disposition* table, that of *admission\_source\_id* in *admission\_source* table and the information of *admission\_type\_id* in *Admission\_type* table. The attributes of *patient\_nbr*, *admission\_type\_id*, *discharge\_disposition\_id*, *admission\_source\_id* act as foreign key in the *Admission\_time\_details* table.

Once the original data was divided into smaller relations, we moved on to looking for values in the instances that did not makes any sense or were not available. We observed that the data which was unavailable was represented by “?”. The missing values can be a critical issue before analysis. If the number of missing values is too high, then they cannot be filled in. For example, the weight attribute had 97% of the missing values, so we decided to remove this attribute since it would not contribute much to the analysis. An important restriction to the filling of missing values is that we should be able to identify a pattern where the missing values can be filled in without any harm to the overall analysis. In our analysis, Payer Code and medical specialty had about 53% of the missing data and we decided to remove these instances since, we had very less knowledge about how to fill in the missing values. Also if the percentage of missing values is too less we can review their effect on the data and if any significant impact is not involved by ignoring them, we can replace them by *null*. The attribute *diag\_3* (third diagnosis) which was useful in predicting the hidden pattern had 1% of the missing data, and by eyeballing we observed that it would not affect the analysis much if these values were treated as unavailable, so we replaced them by *null* values.

After the data cleaning we were left with a total of 52 attributes and there was no effect on the number of instances since we only replaced the missing values with null or chose to ignore them completely.

# Analysis

We studied the various available attributes and what each of those attributes represented in the data. We came up to the decision that we will be going forward with the classification problem. As mentioned before, classification is the process of categorizing which class does a given instance belongs to. We decided to proceed with classification since the attributes that were particularly of our interest involved values such that any given instance could be assigned to a specific class. Based on the attribute description we went ahead with developing two possible hypotheses.

For our first hypotheses we tried to predict the approximate number of days a patient will have to stay in the hospital based on his age, sex, how he was he admitted to the hospital (through emergency room, by a physician referral etc.), how many times was he admitted to the hospital before the current admission, how many times he had consulted a doctor before current admission. The idea behind the hypotheses was that as soon as a patient is admitted to the hospital based on his past history we can predict an approximate number of days for which he will be admitted to the hospital. Here we realized the need to bin the time spent in hospital by the patients since there would be a lot of unique values. We created bins based on two criteria’s. Firstly, we observed the distribution of the various values in the entire data by creating graph of how many times each value is occurring. For example, since around 14,000 people stayed in the hospital for 1 day, 17,000 patients stayed for 2 days and only about 7,500 patients stayed in the hospital for 6 days, if we group the people who stayed for 1 and 2 days together the results would highly be inclined towards this bin. So we decided to keep the patients with 1 day spent in the hospital in a separate bin but grouped the patients who stayed for 5-6 days in the same bin, to make sure that the results do not get biased. Secondly, we also considered that how would the characteristics of the people be similar based on their duration. For example, anyone who has stayed for more than 10 days has a high probability to have the similar characteristics with someone who has stayed in the hospital for about 14 days, so we created a bin of 10-14 days. Based on the above two rules, bins were created to categorize the time spent in hospital. We decided to observe how this hypothesis holds up against the J48 decision tree and the Naïve Bayes tree.

Next, we thought about predicting whether a given patient will be readmitted to the hospital or not. Also, if he will be readmitted, then within 30 days or not. We created a hypothesis that the attribute *readmitted* will be dependent upon the patient’s behavior during his stay at the hospital, this consisted of the patients age, the result of his primary, secondary and third diagnosis, how was he admitted to the hospital, what drugs were administered and his test results (Glucose serum test, A1c test results).

We considered a random cut of 60% from the data based on the above attributes and built the J48 model on it, to observe how is it performing. However, the results of the J48 model were not very impressive. While studying the result of the J48 model, we realized that the J48 Tree was spreading to a large number of branches. After eyeballing we realized that one of the reason for this was the attributes *diag\_1, diag\_2, diag\_3* (primary, secondary and third diagnosis) had a set of numeric ICD9 codes which could vary from 1-999 and some V and E Codes. We realized the need for binning here. After research we realized that these codes actually had a pattern among them, for ex: any codes between 390-459 represent that there is a disease to the circulation system, similarly any codes between 1-139 represented Infectious and Parasitic Diseases [3] and so on. We went ahead and created bins of data inside all the three diagnosis and then checked if the model was able to perform any better.

# Results

Now, we used re-sampling technique from WEKA to split the data in 60 % training & 40% testing set. The results from training set differed from testing set to a great extent. This led us to rethink on our splitting strategy as data might not properly divided. This indicated the presence of inconsistencies among the two sets. It would have been possible that large portion of inconsistencies might be present in either one of the set. So to properly divide the inconsistencies, we decided to split the data in 70% training, 20% testing & 10% validating set. So now, training set won’t differ by much from testing set.

When we built the J48 Model and the Naïve Bayes model based on our selected set of attributes we observed that they were not performing significantly impressive. The J48 Model as well had an accuracy of about 39% while the Naïve Bayes model had an accuracy of only 19%. We identified that there could be two reasons behind such a poor performance either the attributes that we were considering were not correct, or the data was not sufficient to be able to predict the *time\_in\_hospital* correctly. Now, if we were considering incorrect attributes, we needed to add more attributes for improving the result. However, except the attributes that we stated in our hypotheses all the other attributes would be available to us only after the patient was admitted. If we consider any of these attributes it would ruin our purpose of predicting the *time\_in\_hospital* in the first place. Since the entire reason of predicting the *time\_in\_hospital* attribute was to enable the doctors to guess that if supplied with the patients past medical characteristics and general demographic information we would be able to predict the approximate number of days a patient would spend in the hospital. An accuracy as low as 40% was not sufficient so we dropped this hypothesis.

Then we built the J48 and Naïve Bayes model on the training set(which consisted of the attributes based on the second hypotheses) and studied the results. The results of J48 and Naïve Bayes model were still not impressive. This made us realize that what if readmitted attribute is dependent on some other attributes which we haven’t considered yet i.e what if there were some other hidden patterns within the data. The original data set had 50 attributes, so it would be difficult to identify on what attributes readmitted is dependent. So, we used InfoGainAttributeEval technique of Weka that ranks the various attributes that the readmitted attribute is dependent upon. After studying the results of this technique we decided to add attributes in our hypotheses based on ranking. We added attributes *insulin, time\_in\_hospital, num\_lab\_procedures* and some others. Now we built the J48 and Naïve Bayes model on this above refined set of attributes. The results were comparatively impressive with 67% and 57% accuracy respectively. Here by analyzing the confusion matrix we observed that the Naïve Bayes model was biased towards a specific value of the readmitted attribute (No). However, J48 model was not biased towards any value of readmitted attribute. We also learnt that J48 works best when we have a Boolean classifier, here in our case there were only 3 possible values of readmitted (<30, >30, No). Based on these factors we decided to move forward with J48 for further work of finding a way to improvise the results.

Earlier we did not consider the *patient\_nbr* and *race* attribute even though they were significantly high ranked in InfoGainAttributeEval technique as *patient\_nbr* had many distinct values and we thought that the race of a patient would not have a link with him being readmitted. However, after observing the performance of J48 model including the attributes *patient\_nbr* and *race* attribute we realized that they do actually have an impact on readmitted attribute. So we decided to add those attributes to our list of attributes considered for predicting whether a patient will be readmitted. Though the attribute *gender* was not ranked high by Infogain, we thought of considering it as a part of our analysis that it might be possible that women are more prone to get diabetes than men (our assumption). When we tried adding these attributes (*race, gender, patient\_nbr*) to our data set and then created the J48 model around it, our results improved more with a 72% accuracy. This made us realize that these attributes also played a significant role in determining the value of the *readmitted* attribute correctly.

# Lessons Learned

When we have a huge data set, the first step is to learn the significance of each attribute so we can make sense of the data. When dealing with the attributes, it is necessary to figure out how much percent of instances contain invalid or unknown values or missing values. If the percentage is very large then the entire attribute can be ignored. If the percentage is very small in that case the set of values which are invalid or unknown or missing needs to be neglected or needs to be replaced. Whenever the attribute contains the continuous value and if it is possible to group them based on some logic, binning of the values should be preferred. The dataset needs to be divided into training set, testing set and validation set such that inconsistencies of the data are distributed evenly among all the three sets. Whenever we need to determine what role an attribute plays in predicting the result, we can make use of InfoGainAttributeEval technique of the Weka tool.

The final list of attributes considered includes *patient\_nbr, race, gender, age, admission\_type\_id, discharge\_disposition\_id, admission\_source\_id, time\_in\_hospital, num\_lab\_procedures, num\_procedures, num\_medications, number\_outpatient, number\_emergency, number\_inpatient, diag\_1, diag\_2, diag\_3, number\_diagnoses, max\_glu\_serum, A1Cresult, metformin, insulin, change, diabetesMed,* to successfully predict whether a patient was *readmitted,* and if yes within 30 days or after 30 days.

##### References

[1] Silberschatz, Korth, and Sudarshan. Database System Concepts. 6th edition.

[2] http://www.diabetesatlas.org/

[3] Thomson Reuters - ICD-9 Code List http://www.tdrdata.com/ipd/ipd\_SearchForICD9CodesAndDescriptions.aspx