diaknowzit

Diagnosis Prediction
System for Open Medical
Record System

FINAL PROJECT REPORT
MASTERS IN INFORMATION MANAGEMENT AND SYSTEMS
SCHOOL OF INFORMATION, UC BERKELEY

Team:
David Greis, Rohan Salantry and Sayantan Mukhopadhyay

Advisor:
Prof. Marti Hearst

CONTENTS

1. Introduction and Motivation:

Over the past few years, “big data” has become a Silicon Valley buzz term whose slippery meaning only feeds its ever-growing marketing hype. While “big data” may be ill-defined, what is clear is that not everyone is invited to partake in this apparent technological sea change. In particular, non-profit firms seem to be left behind, as Silicon Valley gobbles up talent with salaries that few in the non-profit sector can match. The result is a non-profit sector without sufficient awareness of the potential benefits “big data” can bring, nor the know-how to realize them[1].

The problem isn’t universal; non-profits like Khan Academy leverage “big data” on a daily basis to help students learn around the world. In healthcare, non-profit academic research in the field of biomedical informatics has made huge strides in developing clinical decision support systems--meant to put the power of “big data” at the fingertips of a clinician. Such efforts have been going on so long it begs the question of just how new of a phenomenon “big data” really is[2].

While “big data” may not be new (at least in healthcare), a significant gap remains between the advances in academic biomedical informatics research and their implementation in everyday medical practice. The difficulties that create this gap are many--affecting both the for and non-profit sectors. However, exploring these root causes is not our focus here. Instead, our project deals first-hand with one of them. Namely--the integration of “big data” style analytics tools with existing electronic medical record (EMR) systems.

Insulated to a degree from the eventual economic pressures of post-graduate life, our final MIMS project gives us a unique opportunity to explore problems that may not be personally feasible after leaving the academy. As such, we decided to take the opportunity presented by our final project to address the “big data” labor shortage. We do not know the exact magnitude of this labor supply gap, but it does seem there is no shortage of problems to solve. The only limitation is the availability of data, which can be a major issue in the non-profit sector more generally--as outcomes are more difficult to measure. Fortunately, healthcare does not suffer as much from this problem since outcomes are relatively objective and measurable.

While healthcare data is theoretically available, institutional barriers remain to would-be developers. Healthcare providers closely guard their data for a variety of reasons, including patient privacy. In addition, data is generally locked in enterprise-built EMR systems whose codebases are not public. Two key factors gave our final project group hope that these barriers could be overcome. First, one of our final project team members, David Greis, used to work for a major non-profit healthcare provider, Partners In Health (PIH). Second, PIH’s EMR system is based on the open source platform Open-MRS--a software platform in whose development PIH played a founding role.

While these key factors convinced us that the project was a feasible undertaking, we still faced many challenges in our implementation. We go into depth about these challenges in fourth section of our report. However, we first begin with some background on PIH as an organization as well as the Open-MRS EMR platform. In our 'Problem Statement', we describe the specific problem our project means to solve and in our 'Solution' section, we describe how our tool, *Diagknowzit*, works. In our 'Challenges' section, we go into depth about the challenges--both organizational and technical--we encountered in developing Diagknowzit. Finally, we conclude with a section evaluating the performance of our tool and some final thoughts on future work for the project.

In the end, our project stands somewhat apart from other I School projects in that it is an add-on to a real-world software system--in use by many healthcare providers worldwide. While a project based on stand-alone software would have been easier for many reasons--including the freedom to develop in toolsets more familiar to us--we are motivated by the fact that our project helps further the cause of "big data" use in non-profit organizations. Our tool has the potential to be used by real-world organizations and aid them in their mission to help provide healthcare to the world's most underserved populations.

2. Background: Partners In Health & Open-MRS

PIH is world's one of the largest non-profit healthcare organizations. Based in Boston, Massachusetts, PIH was founded in 1987 by Dr. Paul Farmer, who believes that "the idea that some lives matter less is the root of all that's wrong with the world". Since its inception, PIH has expanded its operations to multiple countries--beginning in Haiti--and expanding to operations in Mexico, Rwanda, US, Russia, Kazakhstan, Lesotho and Malawi. PIH's mission is to "provide a preferential option for the poor in health care". By establishing long-term relationships with sister organizations in resource-poor settings, Partners In Health strives to achieve two overarching goals: to bring the benefits of modern medical science to those most in need of them and to serve as an antidote to despair.

While PIH is first and foremost a healthcare organization, it is also surprisingly a powerhouse of open source software development. For obvious reasons, healthcare organizations need effective information systems--whether they are for or non-profit. While for-profit or academic healthcare providers can afford expensive enterprise EMR solutions--like Epic--non-profit providers needed a more economical solution. To fill this void, several organizations--including

PIH, the World Health Organization, and Google--came together in the mid 2000s to create the Open Medical Record System (Open-MRS).

Conceived as a fully open source EMR system, Open-MRS is a flexible EMR system that can tailor itself to the specific needs of a healthcare provider. As noted by David Thomas of PIH:

“OpenMRS is incredibly adaptable – the 'concept dictionary' allows an implementer to configure the EMR's terminologies to reflect the terminologies used by the local health system, while mapping to international terminology standards at the same time. Additionally, the modular architecture of OpenMRS allows implementations to use the OpenMRS API to develop applications or extensions (or override virtually any part of the out-of-the-box OpenMRS install) as needed. For these reasons, I think of OpenMRS as an EMR platform, rather than as an EMR application.”[3]

Currently, Open-MRS is in use by a growing number of non-profit healthcare providers across the developing world. There are several mechanisms in place (including Github, and a module repository maintained by Google) that allow implementations to share Open-MRS functionality at both the application and the source code levels. The community around OpenMRS is extremely active and open to sharing ideas, technology, and even resources, around shared requirements. For the most part, these exchanges takes place in the Open-MRS wiki and forum resources online.

The core team of Open-MRS development is to a great extent led by PIH. Without their support, as well as the support of the overall PIH organization, our project would not have been possible.

3. Problem Statement

The problem we focus on (and its corresponding solution) can be broadly placed in the domain of clinical decision support (CDS). CDS encompasses a variety of tools to enhance decision-making in the clinical workflow. It provides clinicians, staff, patients, or other individuals with knowledge and person-specific information, intelligently filtered or presented at appropriate times, to enhance health and health care[4].

Diagnosis prediction is an example of CDS. Its utility to the practice of medicine is obvious—an accurate and timely diagnosis is the first step in addressing what ails the patient. There are many ways to approach the diagnosis prediction problem; it is an active area of research in the field of biomedical informatics. Some researchers have used a combination of patient attributes and clinical test results as inputs into machine learning algorithms [5]. Others have leveraged vast medical dictionaries, like the Unified Medical Language System (UMLS) to match keywords or phrases in unstructured text from electronic medical records to map them to formal diagnoses[6]. Other researchers have gone further, applying natural language based grammar rule extraction methods (such as sentence structure decomposition methods, tree methods) to match unstructured text to diagnoses based on their semantic structure[7].

Despite the potential benefits, diagnosis prediction is not without controversy. Some have worried that such an effort is a dangerous attempt to render doctors obsolete[8]. They maintain that making diagnoses is too important to be left to a computer algorithm. Our project attempts to sidestep these issues by tackling a more simple problem—the clerical integrity of an EMR record system. In fact, our system is not even meant to be used by doctors, but rather intake staff or other data entry technicians.

The specific problem we deal with is based on the clinical workflow at PIH. Within PIH's implementation of Open-MRS, there exist both coded diagnoses, which map to illnesses in the concept dictionary of the underlying system database, and uncoded diagnoses--entered as free text--which do not. An uncoded diagnoses may refer to a legitimate medical condition, but it is nonetheless problematic because the system does not recognize it as such. Still, uncoded diagnoses are necessary to make the system more accessible to anyone who might need to use it. More specifically, data entry technicians or intake staff--who lack exact knowledge of medical terminology--enter presumed diagnoses into the system on a daily basis. When these diagnoses enter the system as uncoded, they remain that way until someone with greater knowledge manually points the uncoded diagnosis to a coded diagnosis in the Open-MRS concept dictionary.

While this workflow enables greater accessibility to users generally, it still takes time for a skilled professional to go back and classify the uncoded diagnoses. Presumably, since this person has more sophisticated medical knowledge, he/she would likely have greater impact doing something else. Most notably, if the skilled professional happens to be a doctor, he/she could be spending that time providing life-saving care to patients.

4. Solution: Diagknowzit

4.1 Tool Description:

Diagknowzit operates as an API prediction service for coded diagnoses layered on top of an existing Open-MRS installation. Similar to Google's "Did you Mean…?" feature, Diagknowzit makes suggestions based on user input. Diagknowzit only offers a coded diagnosis suggestion when the user is entering an uncoded free text diagnosis into the system. Further, Diagknowzit thresholds its suggestions based on the probability that it is correct#[1]. If the model is not “confident” in its prediction it will not interfere with the user’s activity.

When Diagknowzit does prompt the user, it does so with a dialog box that asks whether they would like to accept Diagknowzit’s suggested coded diagnosis. The user can choose to either accept or discard this suggestion. In this way, the user is not forced to accept any suggestion that he or she does not want to.

4.2. The Prediction Engine:

Diagknowzit’s predictions are powered by a machine learning algorithm. The model that underpins the algorithm is based entirely on historically labeled data within the OpenMRS database. This stands in contrast to, for example, a model that draws upon information in an external medical dictionary to recognize word tokens associated with a particular medical condition. For its training data, our system only uses uncoded diagnoses that have already been manually paired with a coded diagnosis at some point in database’s history.

There are two obvious weaknesses to this approach. First, the model cannot make intelligent predictions based on user input that it has never seen before--or that has never been manually paired with a coded diagnosis. Second, our model presupposes that *some* data in the database has been manually classified. This is a shortcoming because we are essentially asking the user to do--at least initially--the exact work our tool is meant to save them from doing.

To the second point, we see the initial work of labeling data as an investment for the organization in the future efficiency of their clinical workflow. With respect to PIH, we verified that they did in fact have some data already labeled so that this wouldn’t be a potential snag in our development.

To the first point, we made the decision very early that our project would limit itself to historically labeled data. In approaching this problem, our goal was not to introduce external resources to answer the prediction problem, but rather make sure that the historical “wisdom” embedded within system could be put at the user’s fingertips. Given the challenges we anticipated in integrating even a basic machine learning system into the Open-MRS platform, we feared that choosing a more sophisticated approach would have made the project scope unfeasible. Further, a more sophisticated model would have required a longer gestation period to experiment with our training data. From the outset, it was always a question exactly

[1] Through experimentation, when class with maximized probability was less than 0.5, the model never predicted the class correctly (based on training the model on 100% of the data).

when we would receive training data from PIH. Thus, the ambition of our model development was necessarily circumscribed.

4.3. Integration with Open-MRS

To keep Open-MRS adaptive and extensible, most features are incorporated into Open-MRS as modules that can be loaded and unloaded at will--unless they are dependent on other modules. In keeping with this design pattern, Diagknowzit is built as an independent Open-MRS module.

Because training a machine learning algorithm is resource intensive, it did not make sense for this process to run while the Open-MRS system is carrying on with its usual workload. Therefore, like many recommendation engines in common use, our model trains “off-line”. That is, recommendations are delivered by running the user input through a fully parameterized model. The model is parameterized or trained via the administrative interface, so that a system administrator can choose an appropriate time to let the model run when the system is not busy.

The administrative interface is comprised of a wizard that guides the administrator through the process of initializing or updating the model. The process is divided into two steps. The first step pulls the requisite data from the Open-MRS database and structures it appropriately to be consumed by the machine learning algorithm. The second step runs the machine learning algorithm on the output of step one. The trained model is then serialized and stored in the Open-MRS core module. When the prediction API service is called, the model is deserialized, executed against the user input, and the resulting prediction is delivered back to the user.

Lastly, we wanted to enable the user to update the model to reflect the addition of new training instances that could improve engine performance. Updating the model works the same as initializing it for the first time. Whenever a model is run, it saves its creation date and time. Each time the service is called, the model checks its age; if the model is more than three months old, text appears in the diagnosis entry form that alerts the user to contact a system administrator to update the model.

4.4. User Interface

Because Open-MRS’s is so customizable, there is no single user interface that every organization will use to enter diagnoses into the system. Given that, PIH instructed us to build our tool as an API endpoint with client side javascript code that an organization could integrate wherever they enter diagnoses into their system.

Despite the design directive from PIH, it would still be necessary for us to demonstrate our tool. Thus, we implemented a form for diagnosis entry for purposes of illustration. Because it would never be used by a larger organization, we did not focus extensively on the interface. It was implemented using Open-MRS's basic form entry module, which placed severe restrictions on the kind of HTML elements available to us. This is why our demonstration UI appears rather basic. When viewing this interface, one must keep in mind the its true purpose is simply to demonstrate the functionality of the Diagknowzit tool. Were it actually implemented by an organization, it would be integrated into their specific diagnosis entry form.

5. Challenges

5.1. Access to Data and Cross-Organization Collaboration

5.1.1. Problem

Acquiring training data was a major dependency of our project and by far our greatest challenge. Like most healthcare organizations, PIH is very protective of its data. And just because one of our group members used to work there, that proved to be no guarantee that we would ultimately get any data at all.

Before committing himself to do the project with PIH, David tried to elicit assurances from the Open-MRS development staff at PIH that getting access to data would not pose a problem. At that point, in November 2013, the staff gave us no reason to worry. It was only in January--after we had committed to the project and were setting down to do serious work--that we encountered halting bureaucratic barriers. This was predictable, but nonetheless anxiety provoking.

The main issue was that our 'champion' on the Open-MRS development team was not in control of all the processes that governed the release of PIH's data. PIH gives the director of each country operation tremendous autonomy and authority to run the organization as they see fit within their country. In effect, the country director owns the data. This organizational structure reflects the ethos of PIH overall. When it comes to providing healthcare in the developing world, they believe "bottom-up" works better than "top-down". Furthermore, the organization is particularly sensitive to the legacy of colonialism in international development work. As an organization, they consciously avoid a dynamic where the Boston headquarters "calls all the shots."

The upshot of this all was a high level of uncertainty, which threw the imbalance of risk between our final project group and the PIH development team into sharp relief. Given that our graduation hinged on the delivery of the data, the stakes were much higher for us than for the PIH development team. This fact wasn't lost on our 'champion' at PIH; feelings of obligation towards his prior commitment to us in November kept him motivated to fight on our behalf. Still, matters were still largely outside of his control.

We were largely shielded from the exact nature of the bureaucratic barriers in our way. Our ‘champion’ kept them relatively opaque to us despite our efforts to troubleshoot--one of the strategies we employed to overcome the barriers. Thus, in the end, we can only hypothesize as to the exact organizational dynamics (or combination thereof) that made acquiring the data such a challenge.

We imagine one root cause of the problem was the indirectness of the benefit of our tool. While in theory our tool will save doctors time--enabling them to do more valuable work--we have no quantified sense of the size of this benefit. Just how much time will be saved? And will that saved time be translated into better medical care? By how much? These questions are difficult to answer a priori, and attempting to do so was beyond the scope of our project. Still, without concrete answers to these questions, the value proposition of Diagknowzit could not be made explicit to all stakeholders in a way that was easy to understand.

In addition, we know country directors have a very difficult job that keeps them incredibly busy from day to day. Because PIH is closely affiliated with Harvard University, country directors already field regular requests from academic researchers who need data for ongoing studies. Thus, the organization already finds itself needing to strike a balance between the competing demands of research and service. We imagine our request was viewed as ‘yet another’ request for data. The fact that this request was coming from outside the larger umbrella of the organization probably put the country director even more on guard. It is possible that the director did not understand what the tool would even do. Or even if they did understand, because the benefit of Diagknowzit is so diffuse, it was not enough to overcome the potential risks of sharing data with an outside group--real or imagined.

In the end, we must recognize the possibility that our zealousness to bring “big data” to the non-profit world may be somewhat misplaced. That is, it might not pass a comprehensive cost-benefit analysis. In anticipation of this, before the exact project was defined, our team did try--along with the development team at PIH--to identify which potential “big data” application would bring the greatest benefit to the organization. Since the PIH team wasn’t too familiar with “big data”, and since they had pressing demands of their own to deal with, it is possible this conversation could not take place under optimal conditions.

5.1.2. Mitigation Strategy

Ultimately, the main strategy that prevailed in gaining access to the data was simply patience and persistence. We were lucky that we had enough time to wait while our request eventually made its way through the different bureaucratic barriers within the organization. Bi-weekly update requests to our ‘champion’ kept us in his mind. A delicate yet clear communication style was required to balance our need against alienating our biggest advocate within PIH.

As a back-up plan, we tried to solicit support from other partners within the larger Open-MRS community. These efforts largely did not go very far--aside from a connection made to a Fulbright Scholar based at a rural health clinic in Ghana named Alex Ocampo. Alex provided us with an extract of his organization’s Open-MRS database. Unfortunately, for several reasons, this extract wouldn’t prove useful in the development of our tool. However, it did give us useful context to understand how an organization actually adapted the Open-MRS data schema to suit its specific needs.

From a technical standpoint, the unavailability of the data required serious adjustment of our project plan. Fortunately, one can install Open-MRS with a ‘dummy’ database that is pre-populated with data on roughly 5000 patients. The dummy database was key to develop certain basic elements of our system. For example, the mechanism to pull patient data from the database and structure it properly to be read by the machine learning algorithm could be developed using dummy data. As much as we could, we made sure “the pipes” of our larger system would be ready to handle real data once we received it.

At the same time, in the absence of PIH’s database, we had to make assumptions about the data that did not always turn out to be correct. The problem stems from the flexible structure of the Open-MRS database. Because the concept dictionary is so general, an organization has a lot of leeway around how to represent diagnoses in the system. It was impossible for us to anticipate the exact way that PIH would do so in their database without seeing the data itself. Thus, when we actually received the data in mid-April, some of our prior code had to be rewritten or discarded altogether. As a result, the final code base has elements specifically tailored to PIH. This is obviously undesirable, but given the compressed timeline resulting from our issues with data access, it was likely unavoidable.

The other project element that was severely curtailed as a result of the compressed timeline was the machine learning algorithm selection and development. Because we didn’t know which algorithm would work best a priori, it did not seem to be a good use of time to wait for an answer to this question and then implement the subsequent machine learning algorithm into our code from scratch. Thus, at the outset we decided to leverage the WEKA open source machine learning algorithm library. We chose WEKA because it is written in Java--like Open-MRS--and also because its algorithms implement the serialization interface. It was key that our algorithm run and its results be ‘saved’ so that the algorithm would not need to run each time our service was called WEKA allowed us to do this, and offered a wide variety of different machine learning algorithms that we thought would satisfy the “minimum viable product” version of our project. Structuring our data such that it could be read by the WEKA program required some time, but relative to implementing an algorithm ourselves in a short amount of time, it was the preferred option.

5.2. Working with a Legacy System

5.2.1. Problem

Since it was designed in the mid 2000s, Open-MRS’s core technologies--Java, Maven, and Spring/Hibernate--do not reflect the latest advances in web development technology, and they are certainly not emphasized at the I School. While our group was familiar with many of the concepts underpinning the Open-MRS architecture--APIs, the Model View Controller framework, etc--we were more familiar applying these concepts in modern Pythonic frameworks like Flask and SQLAlchemy. When dealing with Open-MRS’s technology stack, we were on a steeper learning curve, and we discovered exactly how more modern frameworks make our lives easier by abstracting certain development and deployment processes away from the attention of the user.

Additionally, Open-MRS is a system designed for localized web based applications to run on low performance commodity hardware. Therefore, getting the system to cooperate with our demands proved to be a significant challenge. Open-MRS dictates interactions developers can have with the database through its API service layer. Our program required a particular subset of data be pulled from the database after several complicated SQL join statements. Open-MRS’s APIs were too rigid for the task because they could not deliver what was needed with sufficient granularity. Instead, they delivered too much and overwhelmed the web server and Java virtual machine. Somehow, we needed to find a way to adapt the existing system to our needs.

5.2.2. Mitigation Strategy

To acclimate ourselves to the unfamiliar software ecosystem, we had to build time into our project plan to get acquainted with the Open-MRS technology stack. We began our development with small goals (i.e. create a ‘Hello world!’ Open-MRS module) and build up in simple steps from there. When we ran into roadblocks, we sometimes turned to our experienced development partners at PIH. But more often--to avoid bothering our collaborators at PIH further--we turned to the larger Open-MRS developers community or online documentation about Open-MRS. As is the case with many open source projects, we found the existing documentation sorely lacking. When we queried the larger community via the forum, most times we got helpful advice, but other times the our only path forward was painstaking trial and error.

To get around Open-MRS's service layer limitations, we implemented our own API services so that we could execute our raw SQL statements against the Open-MRS database. Learning how to do this took time, but it was the only way we could bend the system to our will. To balance our demands on the system, we employed batch processing, dumping the structured data into an output file bit by bit so it could be interpreted by the WEKA machine learning algorithm.

5.3. User Interface Limitations

5.3.1. Problem

As mentioned above, we were instructed to build our tool as an API service endpoint with client side javascript that an organization could integrate into their specific form entry module. In developing our diagnosis form for demonstration purposes, we would have liked to apply our personal aesthetic preferences and intuitions about good user interface design. However, we found ourselves severely limited by the the default Open-MRS form entry module,

Open-MRS's default form entry module is designed to enable users with no knowledge of HTML to create custom forms to suit the needs of their organization. The module incorporates the inputs needed to communicate with the Open-MRS database as custom tags. A parser engine then renders these custom tags as HTML that is served to the user. To our frustration, the parser engine would not accept most HTML tags. As a result, we could not bring our full knowledge/skills in HTML to create a form as we would have liked.

5.3.2. Mitigation Strategy

We used our knowledge of JQuery to create a modern user experience--as much as we could within the confines of the Open-MRS form entry model. For example, we would have liked to hook our API up to the 'Diagnosis' box in our form--as we felt this would have been more intuitive to the user. However, since free-text was not allowed in this field by the form entry parser engine, we had to hook our service up to the 'Clinical Note' field in our form instead. This felt problematic to us because nothing about a box labeled 'Clinical Note' would suggest that this is where a user should enter their diagnosis. In the end, we used a text instruction directing the user to select "Other non-coded" in the 'Diagnosis' box before a JQuery blur function fires that directs the user's focus to the 'Clinical Note' box and prompts with ghosted text, "What's wrong with the Patient?" Our hope was that this event flow would be sufficient to inform the user how to interact with the system.

Other features we were able to implement include autocomplete for the 'Diagnosis' field, dialog boxes to present the results of the prediction engine as well as system updates for the install/update wizard, tab-operated focus shift, and basic CSS styling for proper alignment. Within the scope of our technical limitations, our advisor, Professor Hearst was incredibly helpful in sharing her expertise in the field of Human Computer Interaction. In the end, while we know we have not achieved the "gold standard" in user experience design, we hope our form satisfies the purpose of our demonstration--to illustrate how our tool might be used within an organization's clinical workflow.

6. Engine Performance/Results & Discussion

As mentioned above, our time to develop the prediction engine was severely curtailed by the length of time we actually had access to the data. Prior to actually receiving the data, we also did not know what exact data we would receive from PIH. We hoped and prepared to receive detailed clinical information about patients to use as features in our model. Ultimately, PIH gave us data from one of their health centers at La Colline, Haiti. La Colline is one of PIH's smaller centers--a fifty-four bedded hospital with a maternity clinic that has a dedicated focus to tuberculosis. While the Open-MRS system at La Colline is used to document diagnoses, it is not used to capture any clinical information on patients. Therefore, the only features that we could ultimately include in our model were the tokens that entry technicians used to describe the patient diagnosis.

We turned these tokens into a simple "bag of words" model and mapped each set of tokens to the coded diagnosis with which they had been manually paired. In total, we had a dataset of about 2,500 instances to train upon. We experimented with three standard machine learning algorithms--naive bayes, multinomial logistic regression, and support vector machine--and reapplied the algorithms on increasing subsets of the data. At each subset size, we randomly separated the data into a training and test set along a 75/25 percent ratio. Ultimately, logistic regression seemed to perform best--with 71% accuracy on average. This was the algorithm we used in the final Diagknowzit prediction engine.

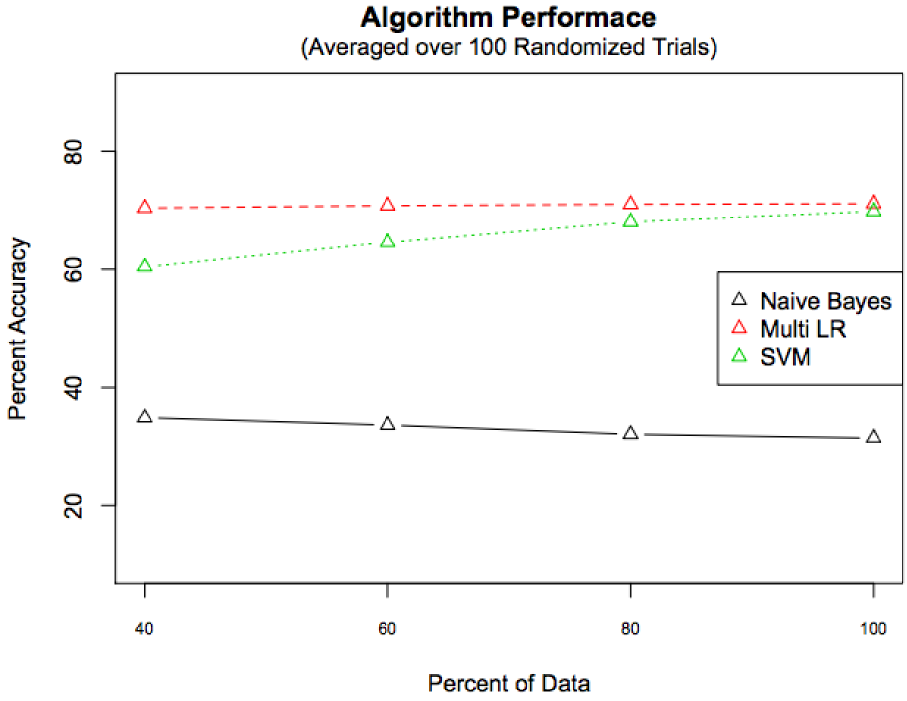


Figure1: Percent Accuracy vs Percent of Data line graph plot for Naive Bayes, Multi LR and SVM classification algorithms

Given the simplicity of our model, the decent performance of the algorithm was somewhat surprising. At the same time, it is within the ballpark of other results seen in the biomedical informatics literature. For example, Rios et al [9] achieved 61% and 88% accuracy on datasets from two institutions--the University of Kentucky Medical Center and the Computational Medicine Center, respectively. Using feature sets generated from external medical dictionaries, they accounted for the difference in performance across the two datasets by citing increased complexity in the Kentucky dataset.

Pestian et al took a text based classification approach for a corpus of 1800 medical records from an anonymous company and Cincinnati Children's Hospital Medical Center[10]. They collected more than 20,000 records over a year period of time and selected 1800 records into a 50 percent training and 50 percent test dataset. Like us, their featureset was based on a bag-of-words model. Classifying 45 radiology-related clinical situations, their methods achieved an average 85% accuracy for Micro-averaged F1 score and around 80% correct for Macro averaged F1 score.

We wish we would have had more time to fully explore our dataset and experiment with different learning algorithms. Given the compressed timeline in which we actually had the data, we are happy that our results sit in the same neighborhood as these other studies.

7. Conclusion/Future Plans

Our hope is that PIH and other organizations will use our module in their everyday clinical workflow. At the same time, we know there are still barriers to real-world utilization. For one thing, our module does not work “out of the box”; some additional set-up is required to integrate Diagknowzit with an organization’s diagnosis form entry mechanism. Also, to make the module useable to other organizations, we would likely need to see one more organization’s full database. In this way we would understand how to abstract away from the specific ways in which PIH stores their diagnosis information. At least some of our group members would be interested in continuing work on these efforts.

Moving forward, we hope that the promising result we achieved with the performance of our prediction engine might overcome some of the organizational “buy-in” barriers we described in our “Challenges” section. But more generally, we hope at the very least that our result will help build awareness of what “big data” tools can do for non-profit healthcare and inspire others to follow in our path.

Works Cited:

1. Bays J., Davis S. “Harnessing big data to address the world’s problems”. McKinsey&Society Voices. McKinsey. Web. 5/8/2014. <<http://voices.mckinseyonsociety.com/harnessing-big-data-to-address-the-worlds-problems/>>

2. Miller R. A. et.al. “The Art, Science, History and Future of Clinical Decision Support Systems”. Princeton University CS Department. Vanderbilt University Medical Center. 2004. Web. 5/8/2004. <<https://owl.english.purdue.edu/owl/resource/747/08/>>

3. Hannan T. "Expert Panel: OpenMRS Implementers: Experiences and Lessons Learned, May 14th to 18th 2012.” GHD Online. 2012.Web. 5/8/2014. <<http://www.ghdonline.org/tech/discussion/expert-panel-openmrs-implementers-experiences-and-/>>

4. Clinical Decision Support (CDS). HealthIT. Web. 5/8/2014. <<http://www.healthit.gov/policy-researchers-implementers/clinical-decision-support-cds>>

5. Kononenko, Igor. "Inductive and Bayesian learning in medical diagnosis."*Applied Artificial Intelligence an International Journal* 7.4 (1993): 317-337.

6. Aronson AR. Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. In: Bakken S, editor. 2001 AMIA Symposium. Hanley & Belfus; 2001. p. 17–21.

7. Friedman, Carol, Pauline Kra, and Andrey Rzhetsky. "Two biomedical sublanguages: a description based on the theories of Zellig Harris." *Journal of biomedical informatics* 35.4 (2002): 222-235.

8. Khosla V. “Technology will replace 80% of what doctors do”. CNN Money. 2012. Web. 5/8/2014. <<http://tech.fortune.cnn.com/2012/12/04/technology-doctors-khosla/>>

9. Rios, Anthony, and Ramakanth Kavuluru. "Supervised Extraction of Diagnosis Codes from EMRs: Role of Feature Selection, Data Selection, and Probabilistic Thresholding." Healthcare Informatics (ICHI), 2013 IEEE International Conference on. IEEE, 2013.

10. Pestian, John P., et al. "A shared task involving multi-label classification of clinical free text." Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing. Association for Computational Linguistics, 2007.

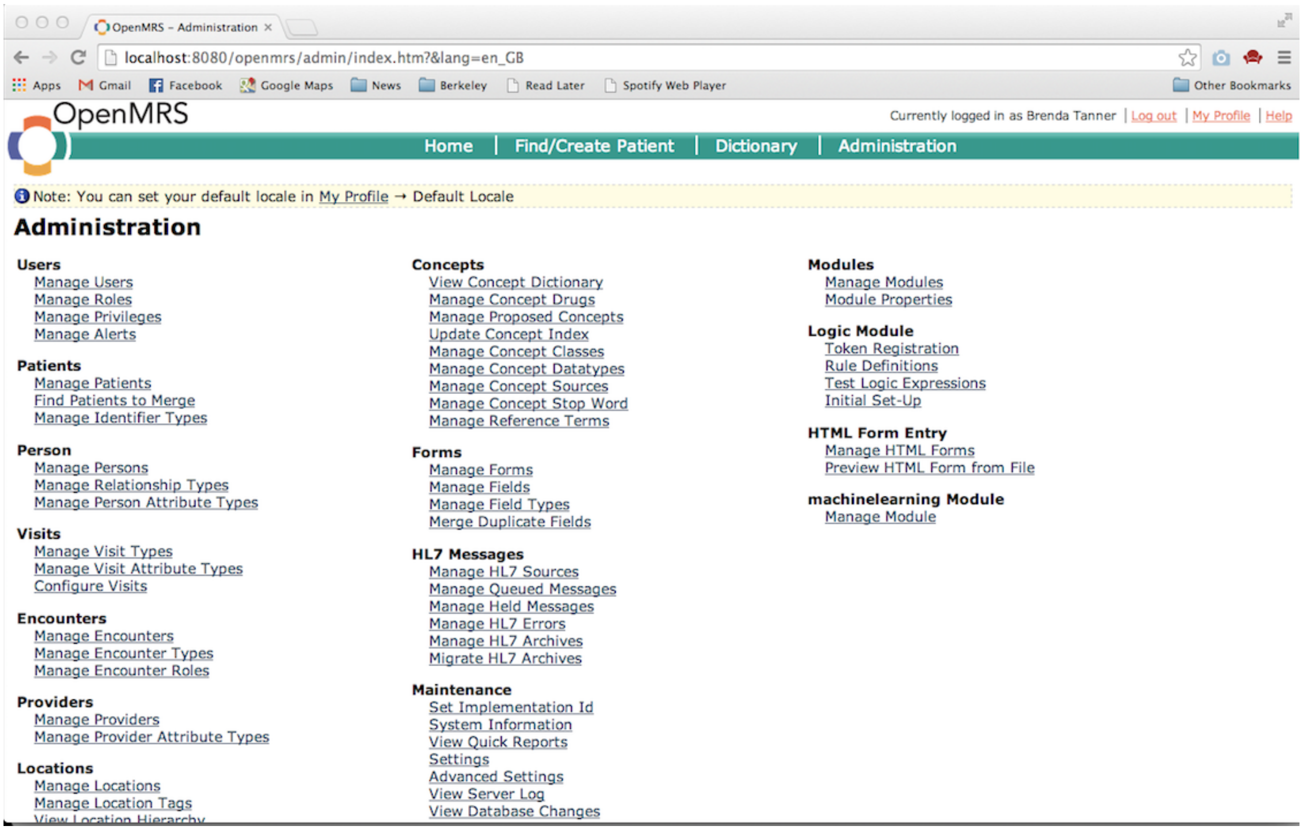
Acknowledgement

We sincerely acknowledge the support and guidance we received from our advisor Prof Marti Hearst. Her extensive experience and expertise in the areas of Natural Language Processing, User Interface Design and Software Systems guided us to refine our design and solution. We also sincerely acknowledge the support from our “champion” in PIH and other developers from PIH who helped us in multiple occasion. Though we have not been able to use but we sincerely thank Alex Ocampo for his support and the data. Also we are thankful to the developers from OMRS community who helped us to solve the riddles of the OMRS system. Finally we want to express our heartfelt gratitude to the ISchool stuff and fellow students for their support and inputs.

Appendix

Please refer to this section for the list of screenshots related to Diagknowzit module within Open MRS installation.

Open MRS Administration:



Note: You can set your default locale in My Profile → Default Locale

Administration

Users
- Manage Users
- Manage Roles
- Manage Privileges
- Manage Alerts

Patients
- Manage Patients
- Find Patients to Merge
- Manage Identifier Types

Person
- Manage Persons
- Manage Relationship Types
- Manage Person Attribute Types

Visits
- Manage Visit Types
- Manage Visit Attribute Types
- Configure Visits

Encounters
- Manage Encounters
- Manage Encounter Types
- Manage Encounter Roles

Providers
- Manage Providers
- Manage Provider Attribute Types

Locations
- Manage Locations
- Manage Location Tags
- View Location Hierarchy

Concepts
- View Concept Dictionary
- Manage Concept Drugs
- Manage Proposed Concepts
- Update Concept Index
- Manage Concept Classes
- Manage Concept Datatypes
- Manage Concept Sources
- Manage Concept Stop Word
- Manage Reference Terms

Forms
- Manage Forms
- Manage Fields
- Manage Field Types
- Merge Duplicate Fields

HL7 Messages
- Manage HL7 Sources
- Manage Queued Messages
- Manage Held Messages
- Manage HL7 Errors
- Manage HL7 Archives
- Migrate HL7 Archives

Maintenance
- Set Implementation Id
- System Information
- View Quick Reports
- Settings
- Advanced Settings
- View Server Log
- View Database Changes

Modules
- Manage Modules
- Module Properties

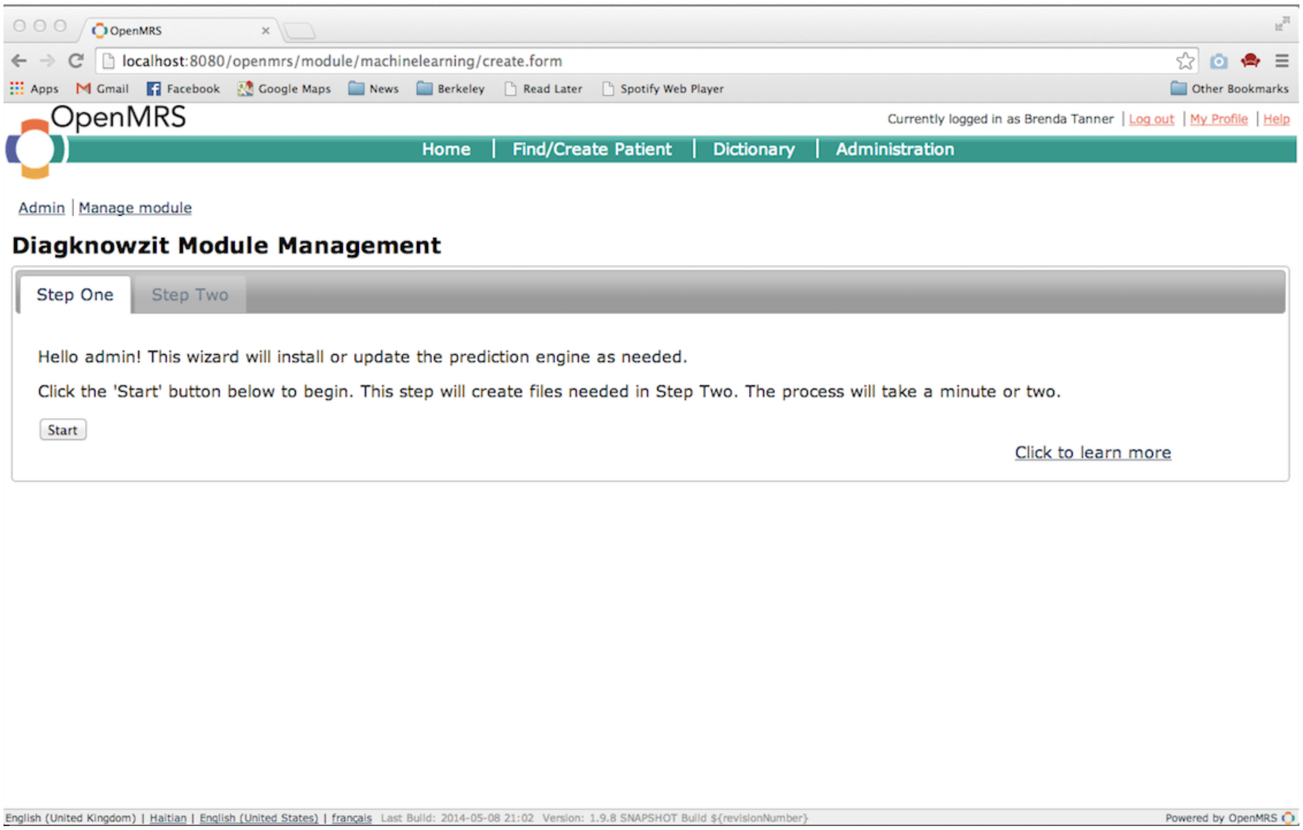
Logic Module
- Token Registration
- Rule Definitions
- Test Logic Expressions
- Initial Set-Up

HTML Form Entry
- Manage HTML Forms
- Preview HTML Form from File

machinelearning Module
- Manage Module

Screenshot 1:Open MRS Admin Page and Link to “machinelearning module”

Open MRS Diagknowzit Module Management:



OpenMRS
localhost:8080/openmrs/module/machinelearning/create.form
Apps
Gmail
Facebook
Google Maps
News
Berkeley
Read Later
Spotify Web Player
Other Bookmarks
OpenMRS
Currently logged in as Brenda Tanner | Log out | My Profile | Help
Home | Find/Create Patient | Dictionary | Administration
Admin | Manage module

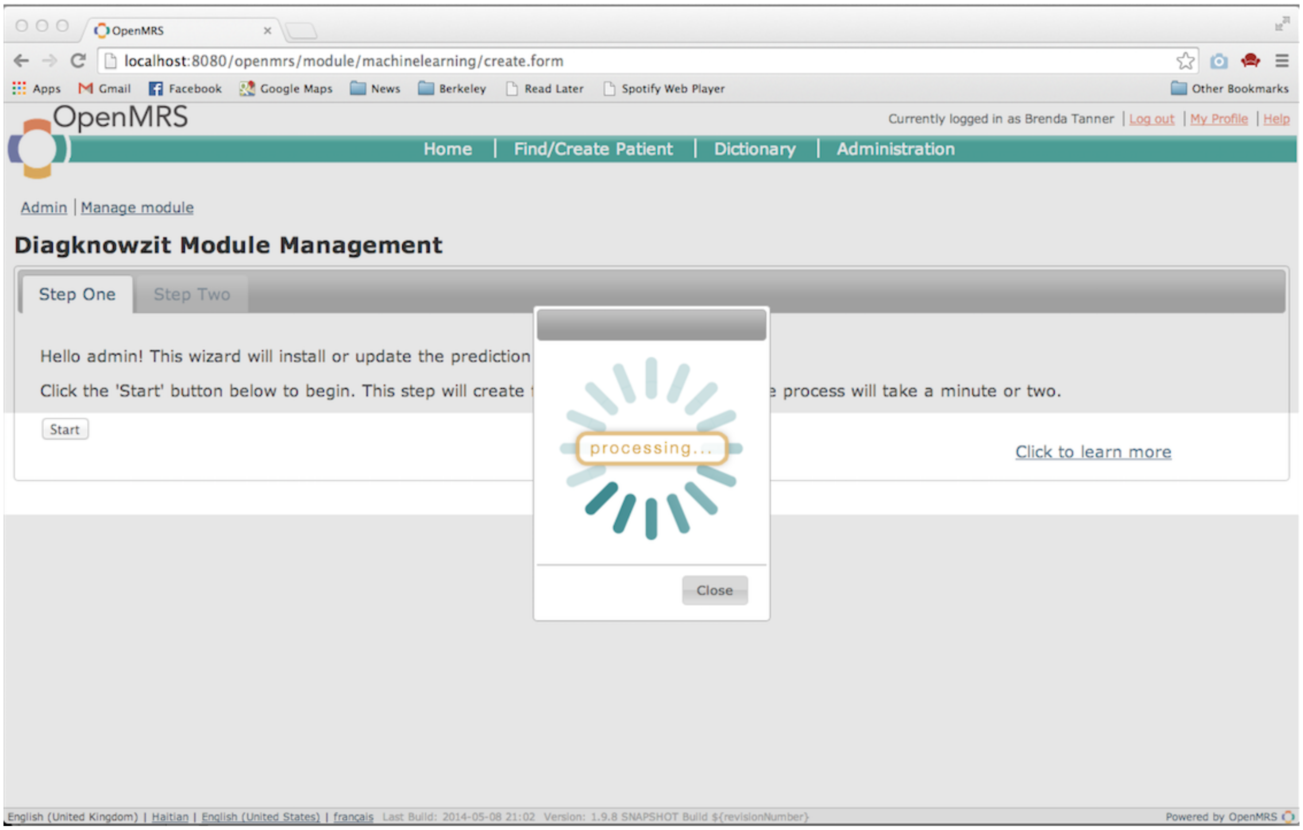
Diagknowzit Module Management

Hello admin! This wizard will install or update the prediction engine as needed.

Click the 'Start' button below to begin. This step will create files needed in Step Two. The process will take a minute or two.

Start
Click to learn more
English (United Kingdom) | Haitian | English (United States) | français Last Build: 2014-05-08 21:02 Version: 1.9.8 SNAPSHOT Build ${revisionNumber}
Powered by OpenMRS

Screenshot 2: Diagknowzit Module Management Step 1 for loading data in the OMRS system



OpenMRS
localhost:8080/openmrs/module/machinelearning/create.form
Apps
Gmail
Facebook
Google Maps
News
Berkeley
Read Later
Spotify Web Player
Other Bookmarks
OpenMRS
Currently logged in as Brenda Tanner | Log out | My Profile | Help
Home | Find/Create Patient | Dictionary | Administration
Admin | Manage module

Diagknowzit Module Management

Hello admin! This wizard will install or update the prediction

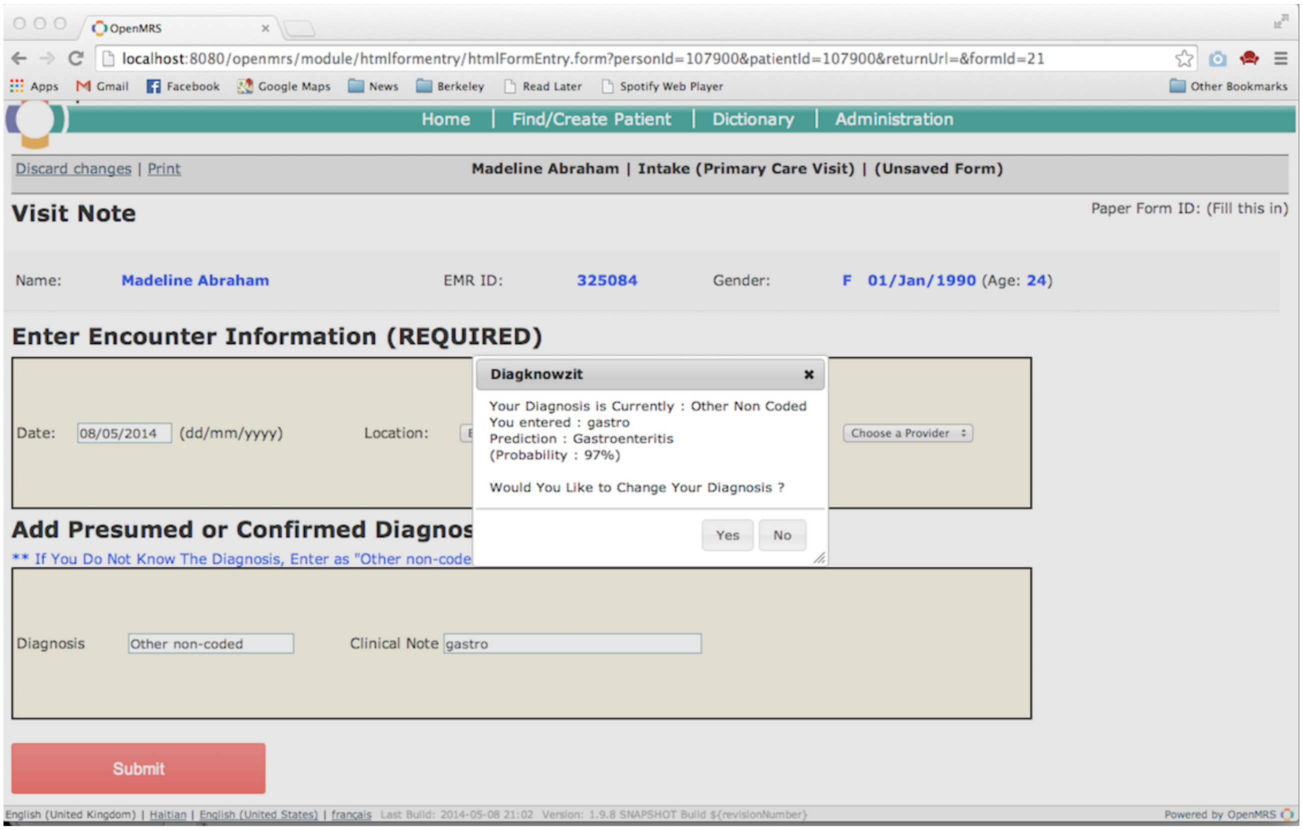
Click the 'Start' button below to begin. This step will create

e process will take a minute or two.

Start
processing...
Close
Click to learn more
English (United Kingdom) | Haitian | English (United States) | français Last Build: 2014-05-08 21:02 Version: 1.9.8 SNAPSHOT Build ${revisionNumber}
Powered by OpenMRS

Screenshot 3: Diagknowzit Module Management Step 1 In Progress

Open MRS Diagknowzit Module Prediction Engine In Action:


OpenMRS
localhost:8080/openmrs/module/htmlformentry/htmlFormEntry.form?personId=107900&patientId=107900&returnUrl=&formId=21
Apps
Gmail
Facebook
Google Maps
News
Berkeley
Read Later
Spotify Web Player
Other Bookmarks
Home | Find/Create Patient | Dictionary | Administration
Discard changes | Print
Choose a Provider
Yes
No
Submit
English (United Kingdom) | Haitian | English (United States) | français
Powered by OpenMRS

Madeline Abraham | Intake (Primary Care Visit) | (Unsaved Form)

Visit Note

Paper Form ID: (Fill this in)

Name: **Madeline Abraham** EMR ID: **325084** Gender: **F** **01/Jan/1990** (Age: **24**)

Enter Encounter Information (REQUIRED)

Date: 08/05/2014 (dd/mm/yyyy) Location:

Diagknowzit

Your Diagnosis is Currently : Other Non Coded
You entered : gastro
Prediction : Gastroenteritis
(Probability : 97%)

Would You Like to Change Your Diagnosis ?

Add Presumed or Confirmed Diagnos

** If You Do Not Know The Diagnosis, Enter as "Other non-code

Diagnosis: Other non-coded Clinical Note: gastro

Last Build: 2014-05-08 21:02 Version: 1.9.8 SNAPSHOT Build ${revisionNumber}

Screenshot 4: Diagknowzit Modeule Prediction Engine in action by Clinician.