EDM Forum
EDM Forum Community

eGEMs (Generating Evidence & Methods to improve patient outcomes)

Publish

8-10-2016

Using Machine Learning and Natural Language Processing Algorithms to Automate the Evaluation of Clinical Decision Support in Electronic Medical Record Systems

Donald A. Szlosek
University of Southern Maine, donald.szlosek@maine.edu

Jonathan M. Ferretti
Champlain College, jon@jonathanferretti.com

Recommended Citation
Szlosek, Donald A. and Ferretti, Jonathan M. (2016) "Using Machine Learning and Natural Language Processing Algorithms to Automate the Evaluation of Clinical Decision Support in Electronic Medical Record Systems," *eGEMs (Generating Evidence & Methods to improve patient outcomes)*: Vol. 4: Iss. 3, Article 5.
DOI: http://dx.doi.org/10.13063/2327-9214.1222
Available at: http://repository.edm-forum.org/egems/vol4/iss3/5

The Electronic Data Methods (EDM) Forum is supported by the Agency for Healthcare Research and Quality (AHRQ), Grant 1U18HS022789-01. eGEMs publications do not reflect the official views of AHRQ or the United States Department of Health and Human Services.

Using Machine Learning and Natural Language Processing Algorithms to Automate the Evaluation of Clinical Decision Support in Electronic Medical Record Systems

Abstract

Introduction: As the number of clinical decision support systems incorporated into electronic medical records increases, so does the need to evaluate their effectiveness. The use of medical record review and similar manual methods for evaluating decision rules is laborious and inefficient. Here we use machine learning and natural language processing (NLP) algorithms to accurately evaluate a clinical decision support rule through an electronic medical record system and compare it against manual evaluation.

Methods: Modeled after the electronic medical record system EPIC at Maine Medical Center, we developed a dummy dataset containing physician notes in free text for 3621 artificial patients records undergoing a head computed tomography scan for mild traumatic brain injury after the incorporation of an electronic best practice approach. We validated the accuracy of the BPA using three machine learning algorithms (SVC, DecisionTreeClassifier; KNeighborsClassifier) by comparing their accuracy for adjudicating the occurrence of a mild traumatic brain injury against manual review. We then used the best of the three algorithms to evaluate the effectiveness of the BPA and compared the algorithm's evaluation of the BPA to that of manual review.

Results: The electronic best practice approach was found to have a sensitivity of 98.8% (96.83-100.0), specificity of 10.3%, PPV = 7.3%, and NPV = 99.2% when reviewed manually by abstractors. Though all the machine learning algorithms were observed to have a high level of prediction, the SVC displayed the highest with a sensitivity 93.33% (92.49-98.84) , specificity of 97.62% (96.53-98.38), PPV = 50.00, NPV = 99.83. The SVC algorithm was observed to have a sensitivity of 97.9% (94.7-99.86), specificity 10.30%, PPV 7.25%, and NPV 99.2% for evaluating the best practice approach, after accounting for 17 cases (0.66%) where the patient records had to be reviewed manually due to the NPL systems inability to capture the proper diagnosis.

Discussion: Evaluation of clinical decision support systems incorporated into electronic medical records can be achieved in an automatic fashion by using natural language processing and machine learning techniques.

Acknowledgements

We would like to thank Craig Lessard, Joseph Mansour, the reviewers, and the editor for their extensive critique, recommendations, and review.

Keywords

machine learning, data use and quality, health information technology, natural language processing, classification, electronic health records, data collection, quality improvement

Disciplines

Biostatistics | Clinical Epidemiology | Databases and Information Systems | Epidemiology | Health Information Technology | Theory and Algorithms

eGEMs
Generating Evidence & Methods
to improve patient outcomes

Using Machine Learning and Natural Language Processing Algorithms to Automate the Evaluation of Clinical Decision Support in Electronic Medical Record Systems

Donald A Szlosek, MPH;[i] Jonathan Ferrett[ii]

ABSTRACT

Introduction: As the number of clinical decision support systems (CDSSs) incorporated into electronic medical records (EMRs) increases, so does the need to evaluate their effectiveness. The use of medical record review and similar manual methods for evaluating decision rules is laborious and inefficient. The authors use machine learning and Natural Language Processing (NLP) algorithms to accurately evaluate a clinical decision support rule through an EMR system, and they compare it against manual evaluation.

Methods: Modeled after the EMR system EPIC at Maine Medical Center, we developed a dummy data set containing physician notes in free text for 3,621 artificial patients records undergoing a head computed tomography (CT) scan for mild traumatic brain injury after the incorporation of an electronic best practice approach. We validated the accuracy of the Best Practice Advisories (BPA) using three machine learning algorithms—C-Support Vector Classification (SVC), Decision Tree Classifier (DecisionTreeClassifier), k-nearest neighbors classifier (KNeighborsClassifier)—by comparing their accuracy for adjudicating the occurrence of a mild traumatic brain injury against manual review. We then used the best of the three algorithms to evaluate the effectiveness of the BPA, and we compared the algorithm's evaluation of the BPA to that of manual review.

Results: The electronic best practice approach was found to have a sensitivity of 98.8 percent (96.83-100.0), specificity of 10.3 percent, PPV = 7.3 percent, and NPV = 99.2 percent when reviewed manually by abstractors. Though all the machine learning algorithms were observed to have a high level of prediction, the SVC displayed the highest with a sensitivity 93.33 percent (92.49-98.84), specificity

[i]Muskie School of Public Service, University of Southern Maine, [ii]Champlain College

CONTINUED

of 97.62 percent (96.53-98.38), PPV = 50.00, NPV = 99.83. The SVC algorithm was observed to have a sensitivity of 97.9 percent (94.7-99.86), specificity 10.30 percent, PPV 7.25 percent, and NPV 99.2 percent for evaluating the best practice approach, after accounting for 17 cases (0.66 percent) where the patient records had to be reviewed manually due to the NPL systems inability to capture the proper diagnosis.

Discussion: CDSSs incorporated into EMRs can be evaluated in an automatic fashion by using NLP and machine learning techniques.

Introduction

The utilization of electronic medical records (EMRs) in order to improve health outcomes through quality improvement efforts has been on the radar for over a decade.[1] The collection of clinician-entered data through EMR systems provides an easy way for health care providers to record their information for viewing by other providers in real time. This information has been used to develop clinical decision support systems (CDSSs) to better aid clinicians in proper diagnosis. The process of evaluating these CDSSs currently requires manually reviewing physician order notes to adjudicate the medical diagnosis. Some studies have provided a workaround for this issue by having manual check boxes for specific symptoms. Incorporating such a task into a clinician's already fast-paced environment can lead to an overburdening of the provider.[2-4] Most EMRs have incorporated interactive CDSSs. The Epic System calls its clinical decision support (CDS) alerts "Best Practice Advisories" (BPAs). For this reason, we will be using the terms "BPA" and "CDS" interchangeably. BPAs notify the clinicians when they need to tend to important tasks such as reviewing orders, completing charts, reviewing the patient's medical history, and medical diagnosing (Epic; Verona, WI). These customized, practice-specific alerts can be programmed by the institution's clinical leadership to activate according to predetermined triggers—either individually or in combination; using inclusionary or exclusionary logic; and ranging from chief complaints entered by nursing staff, vital signs, or diagnoses entered by providers. We decided to validate a BPA for head CT scan utilization in mild traumatic brain injury patients (MTBI) that was already incorporated into the EPIC system. The Canadian Head CT Scan Rule (CHCTR) for decreasing the utilization of head CT scans in MTBI patients has been externally validated over eight times and thus was the obvious choice for validation through our machine learning algorithms.[2-9] The BPA is triggered whenever a clinician orders a head CT scan (Figure 1). The CDS requires the ordering clinician to input data that reflected the evidence-based criteria needed to provide justification for the imaging, as prescribed by the CHCTR[10] consisting of six risk factors: (1) failure to reach a Glasgow Coma Scale of 15 within two hours, (2) any sign of basal skull fracture, (3) vomiting greater than two times after the event, (4) being over the age of 65,

(5) amnesia of events more than 30 minutes before impact, and (6) any high-risk mechanism such as a motor vehicle collision (MVC) or a fall of greater than three feet or five stairs. Thus far, the CHCTR and other clinical decision rules have been validated mostly through extensive manual review of EMRs and data collected from CDSS. This process takes a trained medical abstractor and an extensive amount of time.

With the growing use of EMRs, automated outcome validation may be possible using Natural Language Processing (NLP)—in which a computer processes free text to create structured variables—and machine learning, where a computer distills a data model from input and uses that model to make inferences about future input. NLP and machine learning have already been shown to be useful tools to tease out important clinical information from large numbers of physicians' notes.[11] Such tools could be used with a defined set of clinical criteria, such as the six-level criteria used in the CHCTR, in order to automate the collection of medical diagnosis, mechanism of injury, and other criteria that would be too cumbersome for the provider to select as an option through the EMR.

The primary goal of our study was to compare several machine learning algorithms against manual abstractors in reviewing clinical information to evaluate the efficacy of a CDSS. Maine Medical Center (MMC) has already implemented a CDS in head CT scan utilization for mild traumatic injury patients using the CHCTR. We decided to generate dummy data through the CDS to reflect the structure of real patient information. This data was generated by a programmer guided by a clinician and does not reflect actual patient information. We used these reports to train several machine learning systems and assess their accuracy against manual review of the clinical reports.

Methods

Data Sources

We developed an artificial dummy data set that is structured similarly to that of the EMR system EPIC. The free text in this data set was developed by programmers guided by clinicians in order to produce accurate medical conditions of patients with possible MTBI. All information in this data set does not correspond to actual patient-level clinical data. This data set contains a known population of normal and abnormal head CT scans and was developed in Oracle and SQL to model exported patient level information from the EPIC. This data set was modeled after MMC's Emergency Department on the clinical research scale. MMC is an integrated health care delivery system in the Northeastern United States with extensive electronic health data. The racial and ethnic composition of the artificial data set was modeled similar to that of the surrounding Portland, Maine region including 84 percent Caucasian, 8 percent African American, 4 percent Asian, 3 percent Hispanic, 1 percent other race.[12] The data set contained free text, modeled to be comparable to data collected by clinicians during the time of diagnosis through the EMR. Our dummy data was initially reviewed manually by trained abstractors to determine if the radiologist reported a clinically important brain injury from the head CT scan.

The data set also includes the use of EPIC's CDS system BPA for mild traumatic brain injury patients based on the CHCTR. The BPA launches whenever a clinician orders a head CT, and requires the ordering ED clinician to input data that reflects the evidence-based criteria needed to justify the imaging, as prescribed by the Canadian Head CT Scan Rule (See introduction, Figure 1). If the patient met one of the six criteria outlined by the CHCTR for clinically important traumatic brain injury (ciMTBI), then the provider would select "Criteria Met" and would be able to finish ordering the CT scan. If the patient did not meet the criteria then the provider would click "Does Not Meet Criteria" and the provider would not be able to order a scan. If the providers clicked "N/A," they would not be able to order a scan. If the patient was in a nontrauma incident, intoxicated, or on anticoagulants, then the provider would click "Other" and select one of those three options to order a scan.

NLP and Machine Learning Tools

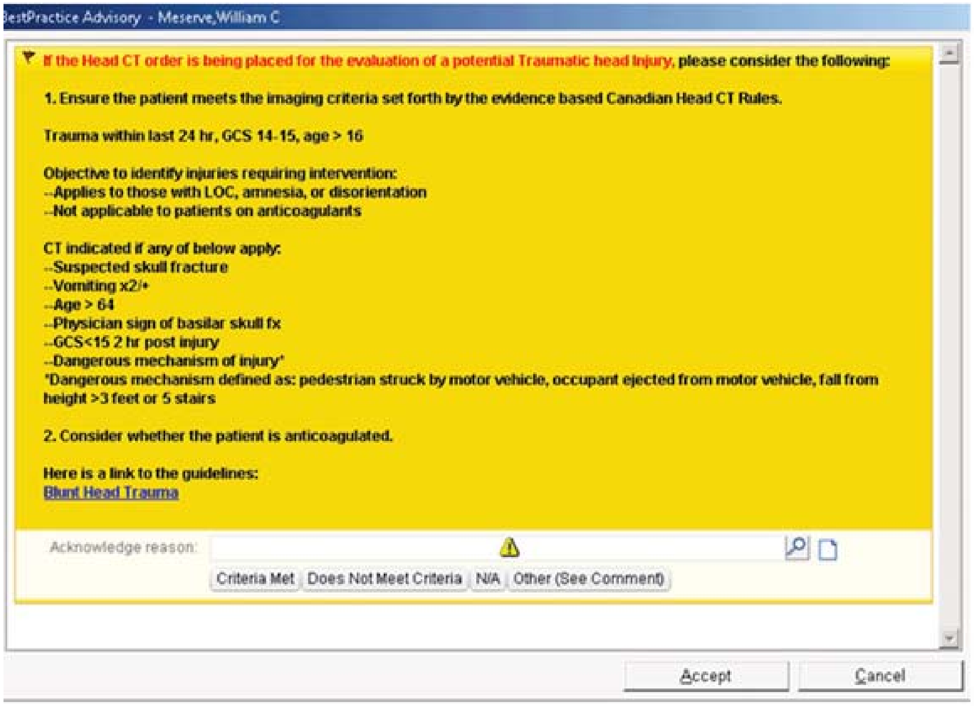
The machine learning and NLP system created for this study was written in the Python programming language version 3.4.3. The library selected for the NLP aspect of this program was spaCy, due to its high speed syntactic parsing capabilities.[13] The machine learning library used was scikit-learn.[14] Scikit-learn was chosen due to its speed, extensive documentation, extensive algorithm, metrics, and the peer review process required to contribute code.

For simple manipulation and distribution, the code was drafted using Juypter Notebook (Jupyter Team, 2015) running the IPython kernel. The environment was created in Linux Containers (LXC) using Docker LXC (Docker, Inc., 2015) to allow for portability, easier repeatability, and disaster recovery.

Machine Learning Process

Our machine learning program runs through a three-stage processing step prior to looking through the data for key words associated with an event. The first stage uses NLP to clean the data by removing common or superfluous words and punctuation. The second stage is another form of data cleaning, called "vectorization and tokenization," where pieces of texts are turned into tokens. If two words are the same, then the *token value* for that word is two. If a token value is high enough to be considered redundant, then the redundant words are removed. The last stage involves training and testing the machine learning programs.

Figure 1. Screenshot of BPA Caption from EPIC EMR System



If the Head CT order is being placed for the evaluation of a potential Traumatic head Injury, please consider the following:

1. Ensure the patient meets the imaging criteria set forth by the evidence based Canadian Head CT Rules.

Trauma within last 24 hr, GCS 14-15, age > 16

Objective to identify injuries requiring intervention:
--Applies to those with LOC, amnesia, or disorientation
--Not applicable to patients on anticoagulants

CT indicated if any of below apply:
--Suspected skull fracture
--Vomiting x2/+
--Age > 64
--Physician sign of basilar skull fx
--GCS<15 2 hr post injury
--Dangerous mechanism of injury*
*Dangerous mechanism defined as: pedestrian struck by motor vehicle, occupant ejected from motor vehicle, fall from height >3 feet or 5 stairs

2. Consider whether the patient is anticoagulated.

Here is a link to the guidelines:
Blunt Head Trauma

In order to train the machine learning algorithms, we split our data set into training and testing data. Of our 3,621 data points, 2,414 (two-thirds) were used for training and 1,207 (one-third) were used for testing. The next phase was to start creating components of a scikit-learn pipeline. A scikit-learn pipeline is a convenience tool for creating a self-contained workflow for a machine learning process; each point of input data must pass through the pipeline. For full details on how we defined our natural language and machine learning processing see the appendix.

We tested three scikit-learn classifiers to assess their different predictions accuracies: k nearest neighbors classifier (KNeighborsClassifier), C-Support Vector Classification (SVC), and a Decision Tree Classifier (DecisionTreeClassifier). The two restrictions we had for classifier selection were its computational resource requirements and its ability to determine the probability of its classifications. KNeighborsClassifier was chosen due to its simplicity and ability to perform multiclass classification, and it was run with its default parameters. SVC was chosen for its ability to handle high dimensional input, i.e., text documents, efficiently and accurately, and was run with prediction enabled and a linear kernel to reduce computational cost. DecisionTreeClassifier was selected for its acceptable computational cost and ability to perform multiclass classification.

Having constructed the pipeline, we fit the training data to each model. Once the training was complete we had each classifier make predictions on our testing data, and recorded the results for later analysis.

Statistical Analysis

For each classifier, we calculated the proportion of reports that the machine learning system classified as requiring manual review, then estimated the sensitivity and specificity for the remaining reports. Sensitivity is the proportion of true ciTBI CT scans that the machine learning system correctly identified as having a ciTBI. Specificity is the proportion of normal CT scans (those scans without a ciTBI) that the system identified as a normal head CT. In addition, we conducted a sensitivity and specificity analysis for the trained abstractors who manually reviewed the radiologists' notes for ciTBI. We then compared the sensitivities and specificities of both the machine learning system and the trained abstractors. In addition, we ran sensitivity analysis for both manual abstractors and the machine learning algorithm against the Head CT BPA to assess whether the algorithm produced similar results. Analyses were performed using SAS Software, version 9.4 (SAS Institute, Inc., Cary, NC).

Further analysis on classifier performance was conducted using the python scikit-learn classification_report function for accuracy, precision, recall, f1-score, and confusion matrices.

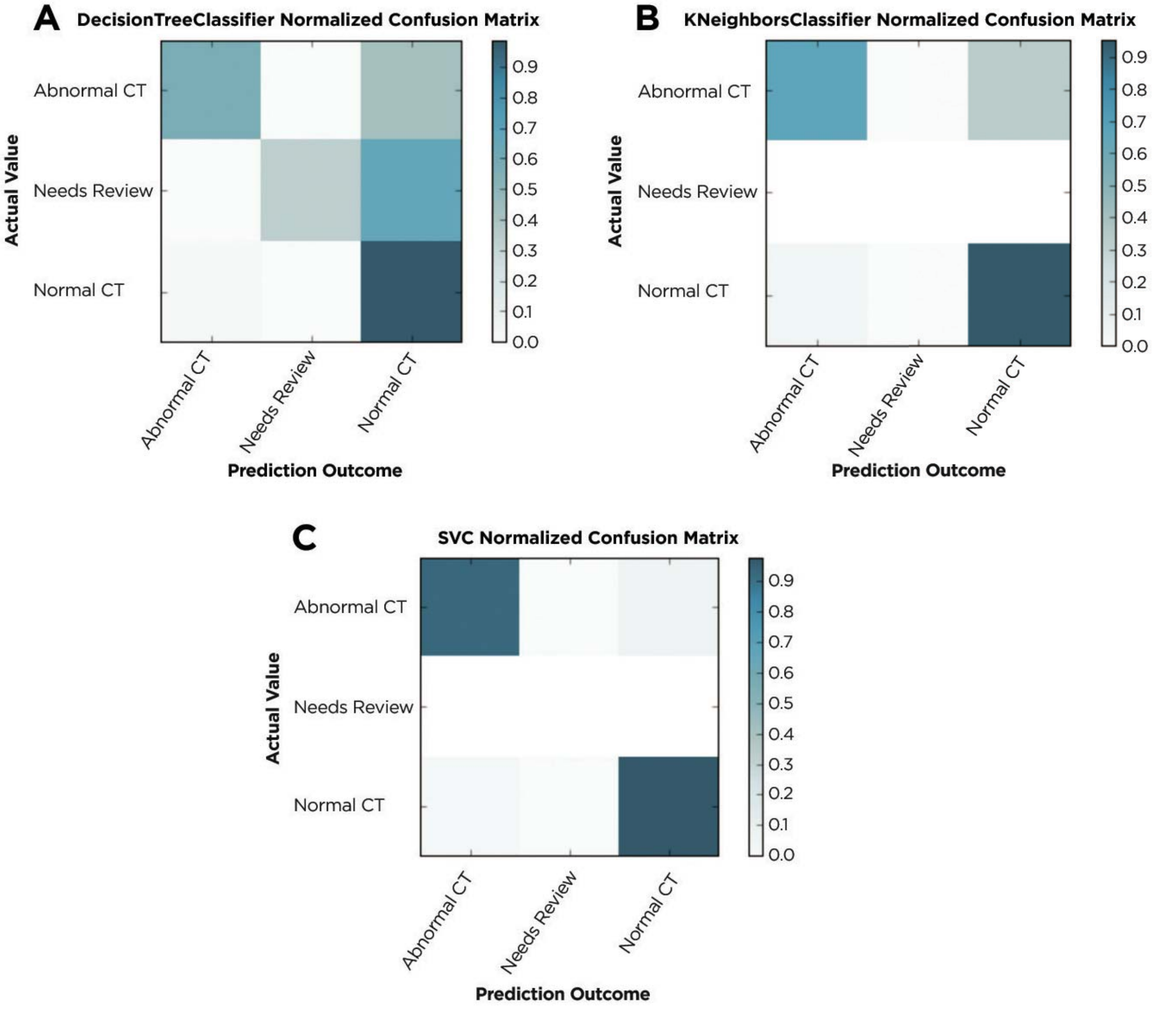
Results

Classifier Performance

After training our machine learning algorithms, we recorded their performances on the testing data set to see which one was the most accurate. By recording the prediction accuracy we were able to gather metrics on individual classifier performance. While all results were within acceptable ranges, some performed better than others. The KNeighborsClassifier was the least accurate, with a precision average of 94 percent, recall average of 95 percent, and an f1-score average of 93 percent. The DecisionTreeClassifier resulted in the following averages: precision 97 percent, recall 96 percent, and f1-score 96 percent. SVC held a 97 percent average across precision, recall, and f1-score. When evaluating the classifiers' accuracy by percentage using the scikit-learn accuracy_score function, the KNeighborsClassifier reported 95.36 percent, the

DecisionTreeClassifier reported 96.11 percent, and the SVC reported 97.43 percent. We also generated confusion matrices (Fig. 2) to better visualize the accuracy of the classifiers. These findings, in addition to the fact that the SVC ran significantly faster than the DecisionTreeClassifier, determined that the SVC was the best classifier for our data. Sensitivity analysis exploring the three classifiers revealed that the SVC classifier had the highest sensitivity compared to manual review at 93.33 percent (Table 1). While the DecisionTreeClassifier had a sensitivity of 57.75 percent and KNeighborsClassifier had a sensitivity of 66.67 percent, the specificity of all three algorithms was relatively the same.

Figure 2. Normalized Confusion Matrix for A: DecisionTreeClassifier, B: KNeighborsClassifier, and C: SVC Classifier



Evaluation of BPA

Due to the high performance of the SVC machine learning algorithm, this algorithm was chosen to evaluate the validity of the BPA. In addition, we tested the sensitivity of the BPA against the manual abstractors and compared the results against that of the machine learning algorithm. We found that the BPA had a sensitivity of 97.9 percent against manual abstractors whereas the machine learning algorithm had a sensitivity of 98.8 percent against the manual abstractors.

Discussion

We have tested three machine-learning-based algorithms for the classification of free text head CT scan data in an artificial health care data. The algorithms classify the physician notes into normal CT, abnormal CT, or needs manual review. These algorithms were developed using an artificial data set with a known population of normal and abnormal CT scans. The most efficient machine learning algorithm was the SVC classifier with a prediction value of 0.97. The classifier reported 1 case 1/1207 (0.08 percent) as "needing manual review." Thus our program could eliminate almost 99.92 percent of manual medical record review with a high level of accuracy while maintaining a low false positive rate and moderate false negative rate.

When the SVC algorithm was being used to evaluate the BPA we found sensitivity results comparable to that of manual review (97.9 percent versus 98.8 percent). Though the sensitivity was 0.09 percent lower in the algorithm, we are able to mark the 3 false negative cases for manual review. Thus, 4 cases 4/1,207 (0.33 percent) of the tested population needing manual review. The SVC algorithm, which was found to be our most efficient at predicting the correct number of abnormal CT scans significantly, decreased the amount of manual review from 1,207 cases without the use of the algorithm to 4. Although we saw a high amount of sensitivity, both the manual reviewers and the SVC algorithm showed low amounts of specificity (10.30 percent versus 9.92 percent) and PPV (7.34 percent versus 7.25 percent), though no different than values found in the literature.[2-8,15] With low specificity, the CHCTR requires further evaluation to prove whether or not it is effective in lowering head CT scans in MTBI

Table 1. Sensitivity Analysis of Three Machine Learning Algorithms Against Manual Review

ALGORITHM	SENSITIVITY	SPECIFICITY	PPV	NPV
SVC	93.33 (92.49-98.84)	97.62 (96.53-98.38)	50.00 (36.50-63.50)	99.83 (99.30-99.97)
KNeighbors	66.67 (12.53-98.23)	95.51 (94.14-96.58)	3.57 (0.62-13.38)	99.91 (99.44-100.0)
DecisionTree	57.75 (45.46-69.19)	98.68 (97.68-99.23)	73.21 (59.46-83.77)	97.39 (96.24-98.20)

Table 2. Sensitivity Analysis on the BPA Against SVC Algorithm and Manual Review

ALGORITHM	SENSITIVITY	SPECIFICITY	PPV	NPV
SVC vs. BPA	97.90 (94.47-99.86)	9.92 (6.76-11.18)	7.25 (6.23-12.50)	99.20 (97.15-99.96)
Manual vs. BPA	98.84 (96.83-100.0)	10.30 (8.13-15.78)	7.34 (0.62-13.38)	99.92 (99.06-100.0)

patients. The use of a computer algorithm that can evaluate such CDSSs faster than, and just as accurately as, manual review can help pinpoint issues in these rules.

We attribute strong test results, in part, to the use of artificial data that has been modeled after real medical data. In addition, the algorithms were not tested on multiple types of data sets (such as a different CDS, EMR system, or region of the country), which has been shown in the past to drastically change the success of classification programs.[16] To further evaluate the use of the SVC algorithm, our next goal is to test it against actual EMR data from MMC's EPIC and compare this to the use of our modeled data set.

While the methods used in this study were sufficient to accomplish our goals, there are still many potential NLP and machine learning methods that could be used to enhance our prediction accuracy or to diversify our label set. Due to time constraints we were not able to fully explore the possibilities granted by NLP. Currently we only use NLP for cleaning our input data, but additional features could be obtained through NLP to assist classification such as sentiment analysis, or in reducing overall vocabulary by reducing sentences to subject, object, verb triplets. Reducing the overall vocabulary would allow for faster training and prediction, and possibly more accurate prediction if fewer extraneous words remain. Different aspects of the machine learning pipeline could also be adjusted to improve speed and accuracy. Further research on classifier parameter tuning and data preprocessing techniques might yield higher prediction accuracy and faster prediction. In addition, free note text is very data rich. Our use of machine learning binned physician order entries into only three categories (normal CT, abnormal CT, or needs manual review). We could increase the level of complexity of our classification scheme to include levels of ciTBI such as intracranial brain hemorrhage, subarachnoid brain hemorrhage, intraparenchymal hemorrhage, etc. The machine learning process could also be used to contribute back into the workflows that feed it. For example, classifier and vectorizer metadata could be used to analyze the input received and to help clinicians better anticipate the outcomes of their CT requests before submitting them.

Converting this program into a service would be a small undertaking, but was outside the scope of this study. The bulk of the code could be placed onto the server that is already hosting the incoming CDS data and could continually make predictions on it. This would allow continual feedback on the CDS. Thus, the CDS could be evaluated in real time from the EMR.

In conclusion, the availability and amount of clinical data in free text has significantly increased within EMRs. New advances in technology such as data mining, machine learning, and NLP have come into use for getting information out of free text in a less expensive and time-consuming manner. These new technologies can also be used evaluate the rise of CDSSs, such as our machine learning algorithm as an evaluation method for the Head CT BPA. Studies such as ours find new potential uses for powerful machine learning systems in evaluating quality improvement efforts to advance clinical care.

This study demonstrates that a machine learning algorithm can be used to identify abnormal head CT scans in free text, health care data with a high degree of accuracy based on internal validation. The goal of this study was not to develop an entirely new machine-learning algorithm for sifting through physician ordered entries, but to use machine learning algorithms in a novel way by evaluating the efficacy of CDSSs. External validation is needed to replicate these results and to explore their performance in other settings.

Acknowledgements

We would like to thank Craig Lessard, Joseph Mansour, the reviewers, and the editor for their extensive critique, recommendations, and review.

References

1. Treweek S. The potential of electronic medical record systems to support quality improvement work and research in Norwegian general practice. BMC Health Serv Res. 2003 Jun 6;3(1):10.
2. Stiell IG, Clement CM, Rowe BH, Schull MJ, Brison R, Cass D, et al. Comparison of the Canadian CT Head Rule and the New Orleans Criteria in patients with minor head injury. JAMA. 2005 Sep 28;294(12):1511–8.
3. Smits M, Dippel DWJ, de Haan GG, Dekker HM, Vos PE, Kool DR, et al. External validation of the Canadian CT Head Rule and the New Orleans Criteria for CT scanning in patients with minor head injury. JAMA. 2005 Sep 28;294(12):1519–25.
4. Schachar JL, Zampolin RL, Miller TS, Farinhas JM, Freeman K, Taragin BH. External validation of the New Orleans Criteria (NOC), the Canadian CT Head Rule (CCHR) and the National Emergency X-Radiography Utilization Study II (NEXUS II) for CT scanning in pediatric patients with minor head injury in a non-trauma center. Pediatr Radiol. 2011 Aug;41(8):971–9.
5. Qushmaq I, Cook DJ. The Canadian CT Head Rule was as sensitive as, but more specific than, the New Orleans Criteria for identifying minor head injury. ACP J Club. 2006 Mar;144(2):53.
6. Edmonds M. The Canadian CT Head Rule reduced the need for CT scans more than the New Orleans Criteria in minor head injury. Evid Based Med. 2006 Apr;11(2):61.
7. Papa L, Stiell IG, Clement CM, Pawlowicz A, Wolfram A, Braga C, et al. Performance of the Canadian CT Head Rule and the New Orleans Criteria for predicting any traumatic intracranial injury on computed tomography in a United States Level I trauma center. Acad Emerg Med. 2012 Jan;19(1):2–10.
8. Bouida W, Marghli S, Souissi S, Ksibi H, Methammem M, Haguiga H, et al. Prediction value of the Canadian CT head rule and the New Orleans criteria for positive head CT scan and acute neurosurgical procedures in minor head trauma: a multicenter external validation study. Ann Emerg Med. 2013 May;61(5):521–7.
9. Kavalci C, Aksel G, Salt O, Yilmaz MS, Demir A, Kavalci G, et al. Comparison of the Canadian CT head rule and the new orleans criteria in patients with minor head injury. World J Emerg Surg. 2014 Apr 17;9:31.
10. Stiell IG, Lesiuk H, Wells GA, Coyle D, McKnight RD, Brison R, et al. Canadian CT head rule study for patients with minor head injury: methodology for phase II (validation and economic analysis). Ann Emerg Med. 2001 Sep;38(3):317–22.
11. Nelson SD, Lu C-C, Teng C-C, Leng J, Cannon GW, He T, et al. The use of natural language processing of infusion notes to identify outpatient infusions. Pharmacoepidemiol Drug Saf [Internet]. 2014 Nov 17; Available from: http://dx.doi.org/10.1002/pds.3720
12. U.S. Census Bureau. 2009-2013 American Community Survey 5-Year Estimates: Portland, Maine. US Census Bureau.
13. Choi JD, Tetreault J, Stent A. It Depends: Dependency Parser Comparison Using A Web-based Evaluation Tool. Available from: http://www.aclweb.org/anthology/P/P15/P15-1038.pdf
14. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python [Internet]. arXiv [cs.LG]. 2012. Available from: http://arxiv.org/abs/1201.0490
15. Smits M, Dippel DWJ, de Haan GG, Dekker HM, Vos PE, Kool DR, et al. Minor head injury: guidelines for the use of CT--a multicenter validation study. Radiology. 2007 Dec;245(3):831–8.
16. Liao KP, Ananthakrishnan AN, Kumar V, Xia Z, Cagan A, Gainer VS, et al. Methods to Develop an Electronic Medical Record Phenotype Algorithm to Compare the Risk of Coronary Artery Disease across 3 Chronic Disease Cohorts. PLoS One. 2015 Aug 24;10(8):e0136651.
17. Rajaraman A, Ullman JD, Ullman JD, Ullman JD. Mining of massive datasets. Cambridge University Press Cambridge; 2012.

Appendix

Machine Learning Process

The first step in our process was a preprocessing stage to clean and verify the state of our testing and training data. The data was first transferred from its original Microsoft Excel sheet to a pandas (Lambda Foundry, Inc. et al., 2011–2012) data frame to retain its tabular organization while working with it programmatically. The columns relevant to this study, "NOTE_TEXT" (note_text) and "CT Result" (ct_result), were then extracted into a separate data frame—the note_text column contained the clinician's notes, and the ct_result column contained the ground truth label for the result of the scan. The data was sanitized by dropping any rows with a column containing invalid data, e.g., a Python None, NumPy "not a number" (NaN), an empty cell, or any other "missing" data that would absolutely cause a program error. The data is then checked against its previous presanitization state to identify any changes. The data will need to be further sanitized later, so we began constructing filters for future use. We created a list of stop words containing common or superfluous words or punctuation that should be disregarded in our test. "Stop words" are words that should be filtered from your data during NLP.[17] We then created a series of "regular expression" ("regex") objects to filter out more complicated data—such as varying dates, time stamps, and stray punctuation—and precompiled them to prevent the regex engine from needing to compile them at runtime for each use. We then split our data set into training and testing data. Of our 3,621 data points, 2,414 (two-thirds) were used for training and 1,207 (one-third) were used for testing.

The next phase was to start creating components of a scikit-learn pipeline. A scikit-learn pipeline is a convenience tool for creating a self-contained workflow for a machine learning process; each point of input data must pass through the pipeline. The pipeline typically consists of stages of data sanitization, vectorizing, and classifying. Our pipeline has the following stages: filter, tokenize, vectorize, filter, and classify (Table A1).

For the first stage we created a custom transformer (every component of a pipeline must be a scikit-learn transformer) designed to clean our input data as it enters the pipeline. Here we ran the regular expressions on the input data, as well as some additional minor filtering.

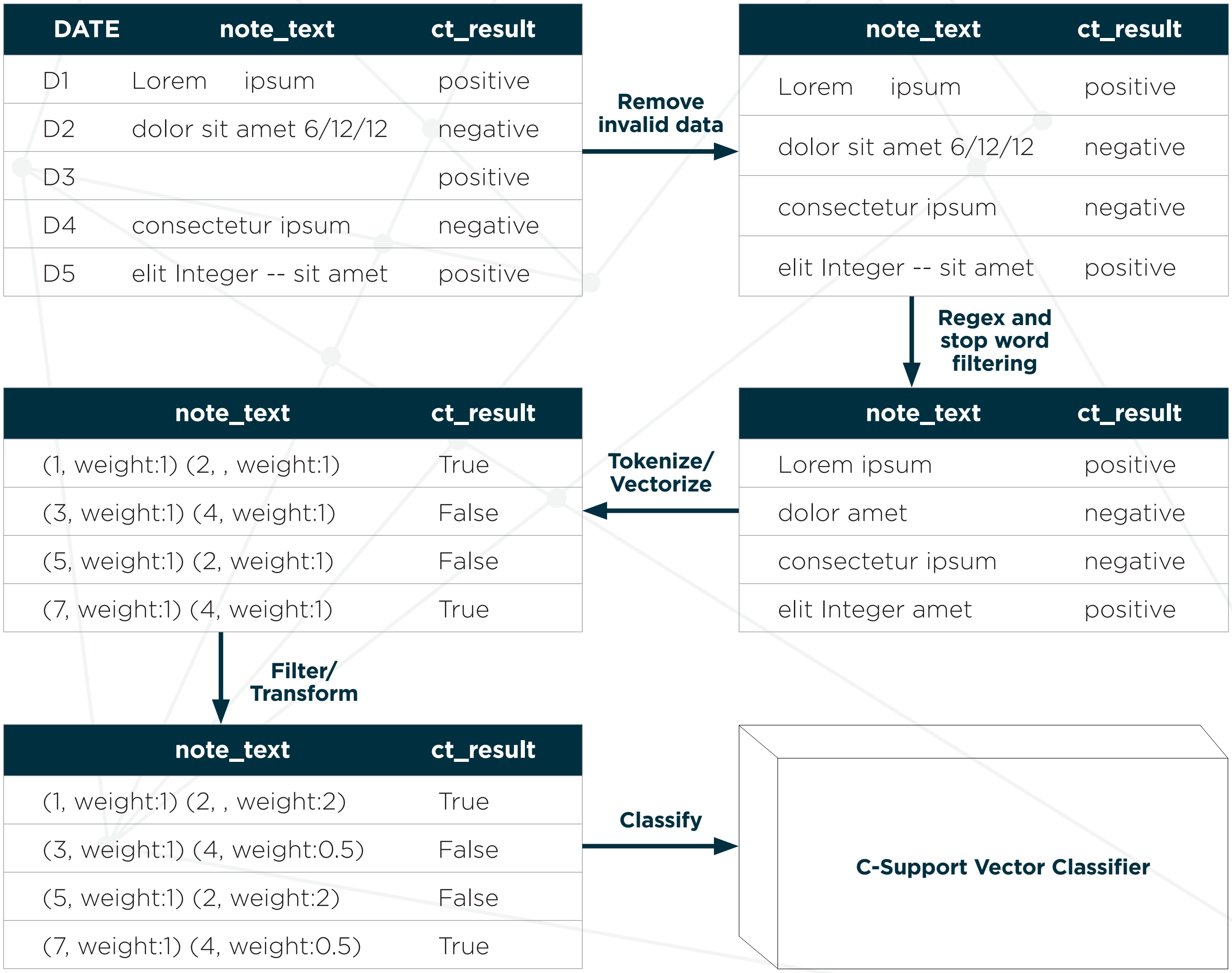
In the second stage we used scikit-learn's CountVectorizer with a custom tokenizer function. CountVectorizer will convert the input text into a matrix of token counts, also known as a bag of words. We set the CountVectorizer to break the text into n-grams of three to four tokens in size. We noted through testing that this gave us the highest prediction accuracy of the variations tested. The tokenizer function uses spaCy's parser for the initial tokenizing. We then iterated over the tokens and used spaCy to identify and remove pronouns and names of people, as well as to replace each word with its lemma (canonical form). This reduces the extraneous data that may confuse the algorithm, reduces the overall vocabulary, and makes it easier for the system to relate various words.

The third step used the scikit-learn or term-frequency (tf) inverse document-frequency (idf) transformer (TfidfTransformer). The TfidfTransformer is designed to reduce the weight of common words by attempting to normalize their frequency within the corpus. While it is optional, our Tfidf does use idf, as we noted better results with it when processing our data set.

The final step in the pipeline is the classifier. We tested three scikit-learn classifiers to assess their different predictions accuracies: KNeighborsClassifier, SVC, and a DecisionTreeClassifier. The two restrictions we had for classifier selection were its computational resource requirements and its ability to determine the probability of its classifications. KNeighborsClassifier was chosen due to its simplicity and its ability to perform multiclass classification, and was run with its default parameters. SVC was chosen for its ability to handle high-dimensional input—i.e., text documents—efficiently and accurately, and was run with prediction enabled and a linear kernel to reduce computational cost. DecisionTreeClassifier was selected for its acceptable computational cost and ability to perform multiclass classification.

Having constructed the pipeline, we fit the training data to each model. Once the training was complete we had each classifier make predictions on our testing data, and recorded the results for later analysis.

Table A1. Machine Learning Pipeline



DATE	note_text	ct_result
D1	Lorem ipsum	positive
D2	dolor sit amet 6/12/12	negative
D3		positive
D4	consectetur ipsum	negative
D5	elit Integer -- sit amet	positive

note_text	ct_result
Lorem ipsum	positive
dolor sit amet 6/12/12	negative
consectetur ipsum	negative
elit Integer -- sit amet	positive

note_text	ct_result
Lorem ipsum	positive
dolor amet	negative
consectetur ipsum	negative
elit Integer amet	positive

note_text	ct_result
(1, weight:1) (2, , weight:1)	True
(3, weight:1) (4, weight:1)	False
(5, weight:1) (2, weight:1)	False
(7, weight:1) (4, weight:1)	True

note_text	ct_result
(1, weight:1) (2, , weight:2)	True
(3, weight:1) (4, weight:0.5)	False
(5, weight:1) (2, weight:2)	False
(7, weight:1) (4, weight:0.5)	True