Binary Diagnostic Classification of Prevalent Cardiovascular Diseases using Support Vector Machines

Jose Javier Gonzalez Ortiz[*]
University of Michigan

Melissa Wu[†]
University of Michigan

Yin Yu Lam[‡]
University of Michigan

April 15, 2016

Abstract

The nuances of modern healthcare diseases have limited the capabilities of performing machine learning and information retrieval on cardiovascular diseases (CVD). In this research paper we explore a straightforward approach in building multiple binary classifiers using Support Vector Machines based on labeled clinical notes, in order to discriminate between the four most prevalent CVD from the MIMIC II dataset. To convey interpretability and actionability, the model is capable of ranking similar documents to the provided unlabeled clinical notes. The results were successful with high precision and recall scores on the classifier and satisfactory relevancy scores from manual evaluation of the documents. Future improvements look towards using different datasets and expansion towards multi-classification.

Keywords: *Information Retrieval, Machine Learning, Cardiovascular Disease, Clinical Notes, MIMICII, Support Vector Machines, Binary Classification, Cosine Similarity*

1 Introduction

The complexity of healthcare in the United States has become increasingly apparent in the last quarter of this century with rising healthcare costs and the growing recognition that common medical problems have shifted from straightforward infectious diseases towards multifactorial chronic diseases that require care from several parts of the healthcare industry. These complex diagnoses have also impacted the difficulty for providers to determine a specific diagnosis for the patient, often having to conduct individual research or reach out to other providers who had similar cases to make this decision. However, the healthcare industry has recently turned towards the support of technology in aiding clinical decisions. Many hospitals now rely on storing records electronically and utilizing these reports to perform analytical tasks such as combining records from many areas of the hospital system in the creation of a clinical decision support tool. Providers can rely on this tool to quickly assess the patient's condition and make an efficient clinical diagnosis decision based on the algorithm of the support tool. Despite the advancement of this technology, the frontier of using machine learning to carry the burden of clinical decisions are impeded by not

[*] jjgo@umich.edu

[†] meliwu@umich.edu

[‡] yylam@umich.edu

only the exponentially increasing types of similar diseases (this is especially prevalent in diseases with high percentage of mutations such as cancer) but by the basic understanding that a disease in one patient does not necessarily reflect in the same way in another patient. Providers need to factor in many other considerations such as age, demographics, past medical history.

Thus, this paper focuses on building a system that looks into clinical reports and uses them to discriminate between two relatively similar affections that belong to different ICD-9 [1] categorizations. Though these reports contain important information for providers to quickly summarize and remember the patients by, they do not usually contain a clearly identifiable diagnosis of the patient.

The proposed system combines both supervised and unsupervised learning. Since the multiclassification task of the most prevalent CVD can be a hard starting point, the algorithm treats it as a series of pairwise binary classification problems. For each of them we train a support vector machine classifier due to its efficiency, interpretability and good performance in the data set. Nevertheless, in the healthcare domain getting interpretable and actionable results is crucial since this will help physicians in making better decisions. Thus, once the algorithm gets a prediction for the ICD-9 code, it returns the most similar documents labeled with that particular code and the most prominent and salient words among these documents. This is an inherent information retrieval task that not only focuses in outputting the documents that have the higher similarity to the one provided but also computes the terms with a higher frequency among the returned documents.

1.1 Previous Work

Multiple approaches have shown the validity of using linear SVMs for text classification as we can see in the works of [Joa98] and [TK02]. Moreover, in the literature we can see that the usefulness of employing feature engineering techniques such as the vector space model and TF-IDF [Joa97].

On the other hand, when looking at the state-of-the-art in information retrieval applied to clinical notes we can find several examples. Among these there is the work of Pakhomov et al. [PHBS08] where they study foot examination findings based in the unstructured text contained in clinical reports. They employed a support vector machine as the classifier of choice and achieved a significant 87% accuracy which strongly suggests the viability of performing text classification tasks in clinical notes corpus. Similarly, the work of Zheng et al. [ZRW$^+$14] uses Natural Language Processing techniques to identify Gout Flares from a corpus of electronic clinical notes. Again in their approach, they used an SVM to compute predictions, but they also employed several NLP techniques to get a better feature extraction from the gold standard dataset they were working with. Consequently, both these works seem to confirm the hypothesis of the SVM being a good classifier to use for clinical note classification. To our knowledge, no attempts have tried to use the SVM parameters to convey an understandable interpretation of the model.

ML and SVMs are not restricted to clinical note analysis in the healthcare domain. The research efforts of Guyon et al. [GWBV02] show the validity of using SVMs and recursive feature elimination to gene selection for cancer classification. In a similar vein, the work of Salem et al. [SRED09] shows that SVM are successful for classifying ECG for various cardiovascular diseases.

[1] International classification of diseases system. Standard of coding for diagnoses in the U.S. until October 2015

2 Background

2.1 Support Vector Machines

In machine learning, support vector machines or SVMs are a type of supervised learning model that has associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. A support vector machine constructs a hyperplane or set of hyperplanes in a high-dimensional space, which can be used for classification. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.[CV95]

3 Dataset and Feature Extraction

3.1 MIMIC II

For this project we have used the deidentified clinical dataset MIMIC II (Multiparameter Intelligent Monitoring in Intensive Care), which consists of roughly a million records with about 30,000 patients. We have focused on the clinical notes and reports fields recorded by physicians since they contain many pertinent details such as basic patient demographics, medical conditions, past medical history discharge summary, nursing progress notes, cardiac catheterization, ECG, etc. All of this information will help in acquiring a robust and reliable classifier. To get the labels for these records we are employing the ICD-9 code field to distinguish between the classes.

Nevertheless, the raw structure and contents of the MIMIC II dataset are not suitable to directly apply machine learning techniques. MIMIC II records are organized by patient ID. However a single patient can have multiple visits to the hospital that will correspond to different Hospital Admission IDs. Since the notes are identified by their Hospital Admission ID, we concatenated the clinical notes of each visit and added a record to our intermediate dataset. The intermediate datasets only contain notes for two specific codes in order to accelerate the data manipulation required for training the classifier. Moreover, patients that have both ICD-9 codes in their listing were not considered since we cannot meaningfully assign a label to those records and thus the example would not contribute well to the model for the classifier.

Furthermore, a main problem that arises when dealing with clinical notes is the sparsity that they usually present. Going through the dataset we encountered some notes that are almost empty, and at the same time, some of them contained extremely long records and exhaustively detailed information. We can get a statistical sense of the length of the clinical notes contained in the MIMIC II dataset in Figure 3.1. The distribution shown in the figure closely resembles a Poisson distribution which makes sense given the inherent nature of the documents.

We performed a filtering of the notes according to their length in characters. Documents that are too short may not contain much useful information, and likewise documents that are too long may contain too much information which can render uninformative. We did some analysis on the documents in the corpus, and have decided to cut around the mean of the distribution considering documents that have between 10,000 and 150,000 characters.

Applying all these criteria we created an intermediate dataset for each pair of codes. Each intermediate dataset contains positive and negative labels (arbitrarily chosen from the two selected

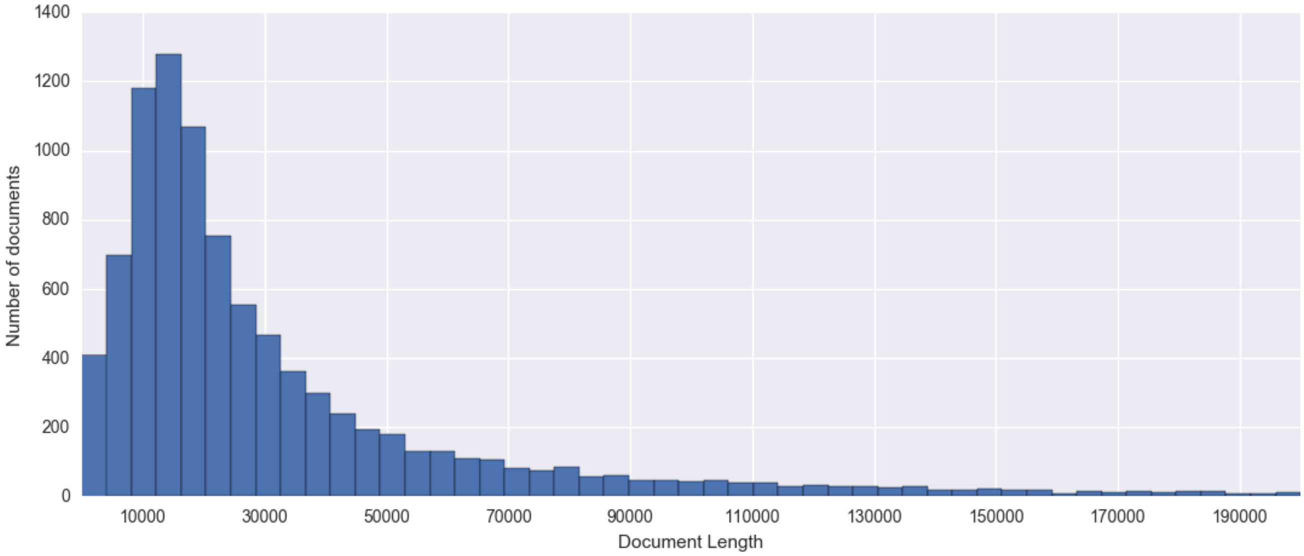

Figure 3.1: *Histogram of the clinical notes lengths (in characters) of the MIMIC II dataset*

codes) and the concatenated notes for each unique Admission ID.

4 Methodology

This paper presents a methodology that involves both supervised and unsupervised techniques for text classification. Figure 4.1 presents the general structure of the system with the main building blocks. We start from one of the multiple intermediate corpus detailed in the last section and compute a 80%-20% Training-Test split to properly evaluate the performance of the classifiers. We apply multiple preprocessing techniques that transform the documents in a vector representation model that we can directly feed the Linear SVM. Once we have trained the classifier, we can provide it with an unseen test document and get the predicted ICD-9 Code. With the decision of the predicted class, we use cosine similarity to retrieve documents that are in the same ICD-9 code class, and rank them according to their similarities compared to the provided document.

4.1 Selection of ICD-9 Codes

Since one of the main caveats of building a reliable text classification scheme is obtaining a large enough corpus so that the classifier is able to learn a meaningful model, we performed a preliminary analysis of the dataset using several queries in order to rank ICD-9 codes by their absolute frequency. The top four results are as follows 401.9, 414.01, 428, 427.31. All of these correspond to CVD and therefore in general they can have similar symptoms and manifestations.

It was the intention to have several sets to see how the performance is for different combinations of ICD-9 codes in order to boost the robustness of our project design and to avoid overfitting for one specific combination of codes. Table 4.1 shows the total number of records in the master file for each code and each code combination. Most of the combinations have around 9,000 records which seems like a large enough sample of documents to properly train a text classifier. Note however that, as we commented previously, records containing both codes will be ignored since it is not easy to

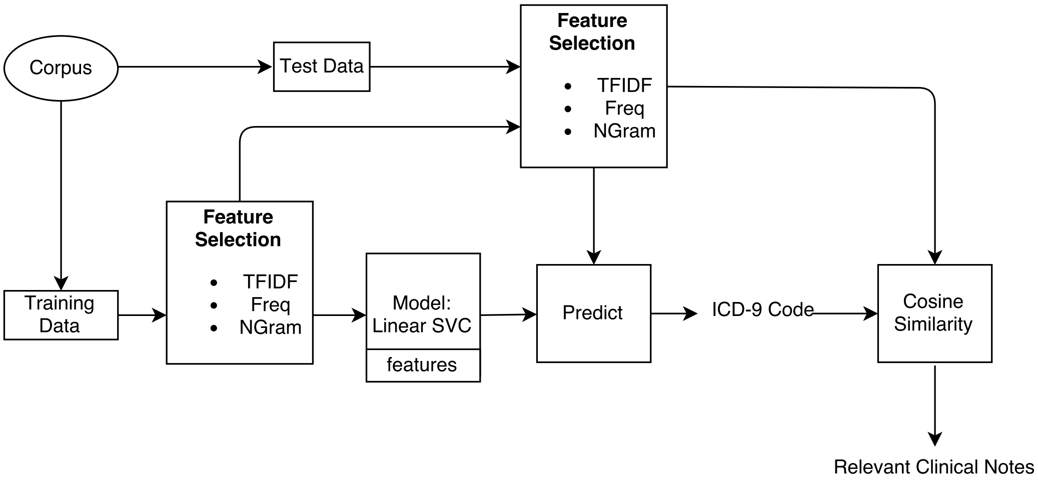

Figure 4.1: *General diagram of the system*

properly label them.

4.2 Preprocessing

As we can see in Figure 4.1, Feature Selection is performed for both the Training and the Test sets. However, specific TF-IDF parameters are learned when fitting for the Training Data and are later employed when transforming the Test data. As a rule of thumb we want to transform the test documents independently but also generate an analogous feature representation of that achieved for the training data.

Since Linear SVMs deal with arbitrarily dimensioned euclidean vectors, we need to apply several preprocessing techniques to achieve a vector representation of the raw data. Thus the following transformations are performed.

1. **Removal of all non-alphanumerical characters**. Only numbers, lowercase and uppercase letters and the percent sign % are allowed, everything else is replaced with a space. We want to remove these special characters since they are usually employed for formatting schemes in the notes but do not convey any useful information for the text classification task.
2. **Space squeezing** is performed where multiple instance of a space are replaced with just a single one.
3. **Stopword Removal** is performed for the English language stopwords, since they do not convey meaningful information despite their high density.
4. **N-grams generation**. Due to medical terms usually spanning multiple words we have used a model that considers combinations of terms including unigrams, bigrams and trigrams.
5. **Relative frequency thresholding** Moreover, in order to remove outliers and reduce the final dimensionality of the model we introduce thresholds for the document frequency of a term. Terms that appear in less than 2.5% of the documents are ignored since they fail to convey the characteristics of any class. At the same time, terms that appear in more than 90% of the documents are also ignored since they constitute as a stopword category for this corpus in particular.
6. **Vector Space Model** Due to the robust re-

ICD-9	Freq.	Description
401.9	11292	Unspecified Essential Hypertension
414.01	7148	Coronary Atherosclerosis of Native Coronary Artery
428	7114	Congestive Heart Failure Unspecified
427.31	6726	Atrial Fibrillation

(a) *Absolute frequencies of the most common ICD-9 Codes*

	401.9	**414.01**	**428.0**
427.31	11849	9177	7936
428.0	13104	9382	
414.01	10023		

(b) *Records per ICD-9 Code combination*

Table 4.1

sults that has been proved to achieve in the past, we use the simple Vector Space Model representation of the N-grams

7. **Term Frequency Inverse Document Freqeuncy** Due its good results with linear SVM in the literature [Joa97], we employ TF-IDF weighting to smooth the absolute frequency counts and and get the most relevant terms within the corpus as a whole.

4.3 Binary Classification

As we have stated previously Linear Support Vector Machines[2] were the classifier of choice for the final model of our system since they were show to have a relatively quick training runtime, good performance scores and a good way to interpret which words the model is using to classify the documents into one class or the other. Since we are going to produce vector representations with several thousand dimensions to get both a model that is the most interpretable and that is able to generalize better, we will want for the weights of the LSVM to be as sparse as possible. To enforce this sparsity we simply employed the LASSO cost function by introducing L1 regularization. To confirm this choice, results were also computed for L2 regularization and as we had expected, it produced a larger generalization error.

Nevertheless, several other classifiers such as Naive Bayes, Decision Trees or Random Forests were employed but none of them achieved better results in all of the previously mentioned areas to compete with the choice of LSVM. Random Forests achieved strikingly similar results to LSVM and have also a way to get interpretability from the trained model; however, when comparing the runtimes it took to train the classifier, LSVM outperformed Random Forest significantly.

To select model parameters for both the preprocessing stages and the classifier itself, exhaustive cross validation was performed. Multiple possible realistic values were chosen as candidates for each parameter and a combinatorial grid search was executed, and in order to reduce possible biases in the data, a five fold cross validation was performed in this stage. Cross validation was only performed for the 414.01 - 428.0 combination. The parameters learned from said cross validation were used for all the code combination on the assumption that clinical notes among these codes were fairly similar. As we shall see in Section 5, the performances were consistent so we applied those parameters directly to our model instead of repeating the cross-validation process for time optimization purposes.

In the medical domain, it is extremely relevant for machine learning systems to be able to

[2] We employed the LinearSVC available through the publicly available sklearn Python package. Preprocessing, crossvalidation, metrics and evaluation were also implemented using standard methods from the sklearn library

explain how the decisions are made and which factors were the most relevant for the produced output. In order to convey this interpretability, we can exploit the fact that SVMs with a linear kernel compute a vector of parameters that directly correspond to the features in the vector space model. In general, the larger the absolute value of the parameter, the more relevant said feature is for classification purposes; and the sign of the value will tell us for which of the two classes is said parameter contributing to in the dot product. Therefore, once we have trained our classifier, we produce a sorted list of the parameters and their associated terms, reporting the most positive and most negative ones since they will make larger contribution during the classification.

4.4 Clustering

The interpretability technique outlined in the previous section is independent of the test document at hand; namely, it will return the same terms no matter which test document we provide. Thus, it would make sense to return some kind of information which would be of special interest given that we know the contents of said test document.

In order to provide more interpretability and actionability of the model, we get all the documents from the predicted ICD-9 code class, and compute the cosine similarity of the TF-IDF weighted vectors of the test document and every document within the predicted code class. Next, we sort them and return only the ones with this cosine similarity closest to one. For the purpose of this analysis we chose to arbitrarily return 15 documents. It would be interesting, though, if we could extract the terms that are specifically common amongst these documents but not throughout the whole class.

A way to extract said terms is to compute the direct sum of the TF-IDF vectors of these most relevant documents and sort the terms by their score. This way, only terms that have a high TF-IDF score in the majority of this documents will be returned; and by the implicit definition of TF-IDF, we will avoid returning terms that are common among the whole class. Some other approaches were explored, including a backtracking analysis of the cosine similarity computation to determine which features make two documents to have a high similarity score; or a nearest neighbor model. In the end, the proposed model was chosen due to its straightforward implementation and analysis.

5 Classification Results

In this section we present the performance scores of all the pairwise classifiers for the 4 mentioned ICD-9 Codes. This section will only deal with the classification results.

ICD-9 Codes		Accuracy
414.01	428.0	0.489
414.01	401.9	0.68
414.01	427.31	0.52
428.0	401.9	0.66
428.0	427.31	0.522
401.9	427.31	0.697

Table 5.1: *Baseline Classifier Accuracy*

5.1 Baseline Classifier

In order to get a sense of both class imbalance and how hard the classification problem is, we built a simple naive baseline that computes the majority label class in the corpus and labels everything as that class. Results are shown in Table 5.1. As it was expected from the statistics shown in 4.1, class imbalance is not a problem since for most combinations there is close to an equal amount of

ICD-9 Codes		Precision	Recall	F1 score	AUROC
414.01	428.0	0.928	0.925	0.925	0.979
414.01	401.9	0.925	0.925	0.923	0.978
414.01	427.31	0.957	0.957	0.957	0.991
428.0	401.9	0.913	0.913	0.911	0.970
428.0	427.31	0.927	0.926	0.926	0.977
401.9	427.31	0.937	0.936	0.935	0.985

Table 5.2: *Performances for the Training Set*

ICD-9 Codes		Precision	Recall	F1 score	AUROC
414.01	428.0	0.892	0.891	0.891	0.952
414.01	401.9	0.897	0.897	0.893	0.948
414.01	427.31	0.913	0.913	0.913	0.974
428.0	401.9	0.845	0.847	0.844	0.915
428.0	427.31	0.888	0.887	0.886	0.945
401.9	427.31	0.907	0.906	0.904	0.962

Table 5.3: *Performances for the held out Test Set*

records pertaining to each code.

5.2 Linear SVM

Classifiers were trained for each of the $\binom{4}{2} = 6$ possible combinations. For the parameter selection, a combinatorial grid search cross validation was carried out for the codes $(428.0, 401.9)$ and then used for all the classifiers. For each combination, multiple statistics were drawn since we wanted to see if there were any weaknesses in the presented algorithm. Results for the training set scores are shown in Table 5.2 whereas results for the test are presented in Table 5.3. It is important to notice that since this classification task does not contain inherently positive or negative class, average values for each score were computed for the scores of each class in order to convey the overall behavior of the system.

As we can see in the tables, the performance for the different estimators is quite similar in both training and test sets (with the exception of AUROC for the test set), so we will focus on F1-Scores for the purposes of the analysis since it has been shown to be a good metric in text classification problems. Therefore, a closer look into the training set scores reveals that the results were extremely satisfactory, with all the F1-Scores above a 90%. Moreover, AUROC results are even higher with all of them being closer to 98%, which suggests that a careful choice of the classifier bias could significantly improve its performance. Nevertheless, results are strikingly good when compared to baseline classifier.

Continuing this analysis to the test set in order to get a sense of the generalization error of the classifiers, we can see that F1-Scores are still close 90%. This indicates that the model is not overfitting to the data since both sets of scores are fairly close, and that the assumption of extrapolating the parameters from the cross validation to all of the classifiers held. In a similar way as it happened in the training set, AUROC scores are

5% higher which hints that our linear SVM model can potentially achieve those performances with the right bias.

As a general trend, we can see that no two ICD-9 codes are especially difficult to be differentiated from each other since most of the scores are similar for all of the combinations. In addition, we can see that there is no correlation between the class imbalance of 70%-30% shown in Table 5.1 and the results shown here, which implies that the model is robust to a reasonable class imbalance.

6 Relevance Evaluation

As we explained, once trained the model is able to generate a relevant features table for each of the codes used to trained the classifier by using the absolute value and the signs of the weights learned by the linear SVM. We can see an example in Table 6.1 for codes 414.01 & 427.31, where only the top 15 terms for each class are shown. Similar tables were computed for the different each of the combinations, however all of them display a similar structure to the one shown for codes 414.01 & 427.31. The table contains a series of terms which range from unigrams to trigrams, but as we see the most relevant terms tend to be bigrams. This result seems to indicate that bigrams can be specially informative when processing information contained in clinical notes. A closer look into these terms reveals their significance. Except a few of them, most contain medical terms closely related to the diseases at hand which confirm the fact that the preprocessing techniques have been accurate enough to retrieve just the relevant terms for the classification purposes.

Note that for the amount of similar documents being retrieved is arbitrary and was set to 6 for evaluation purposes. In a real world scenario, it would be easy for the physician to get any amount of documents.

In order to get a proper evaluation these rankings of relevant documents being retrieved, evaluators were asked to score out of 6 how many of the top 6 retrieved documents chosen were relevant to the test document. In addition, they were requested to comment on the features the classifier chose (as displayed in Table 6.2) and how applicable these terms were to the actual diagnosis. The evaluators deem an average of the feature terms (70%) as features that they recognize in condition. For example, 414.01 (Coronary Atherosclerosis of Native Coronary Artery), the term "mechanical" is a word often seen as a type of complication in conjunction to individuals who have the disease or in such other cases such individuals who had Atrial Fibrillation and often associated with the term "chronic microvascular" angina in their diagnosis.

ICD-9 Codes		Avg. Rel. Score
401.9	427.31	0.50
414.01	401.9	0.50
414.01	427.31	0.66
414.01	428.0	0.50
428.0	401.9	0.66
428.0	427.31	0.66

Table 6.2: *Average Relevance Scores for the top 6 most similar documents returned for each test document*

From these results we can conclude that whereas not perfect, at least half of the top 6 returned documents were relevant which seem to suggest that the provided model is a good first approach at retrieving similar cases in medical scenarios. However, a larger evaluation should be carried out to corroborate the statistical significance of the results presented here.

θ	414.01	θ	427.31
14.187	transitioned	-21.026	chronic microvascular
10.281	Continue mechanical	-20.788	improved compared
5.937	Cardiac Surgery	-16.135	hemorrhage There
5.356	pneumothorax detected	-16.020	urinary output
5.302	internal carotid	-14.606	increase
4.594	5MG	-13.007	afebrile
4.381	AS DIRECTED	-11.697	looks
3.853	3am	-8.619	specific
3.659	angiographically apparent	-7.482	3MM
3.518	25 12	-6.802	VIDEO OROPHARYNGEAL SWALLOW
3.342	Social No	-5.199	bundle
3.326	held	-3.646	iv given
3.271	ATTEMPTS	-3.459	TO PAIN
3.082	Pt remained	-3.423	basilar
2.842	Abdomen	-3.223	Amiodarone 400

Table 6.1: *Most relevant features for the code combination 414.01 & 427.31*

7 Conclusion

Overall the presented model exceeded our expectations, with promising results that could potentially lend itself towards future endeavors. We like to highlight the strength of our project when dealing with unstructured data and the exceptional performances achieved with general preprocessing techniques such as document frequency and TF-IDF.

The model employed linear support vector machines in order to differentiate between similar albeit not identical types of cardiovascular diseases. The choice of LSVM for interpretability was satisfactory with most of the learned parameters extracted from the trained classifier being relevant features when differentiating diseases. This is a significant result that if applied correctly, would allow physicians to look for unknown underlying patterns when trying to discriminate similar types of diseases if given a large enough corpus of clinical notes.

It is important to mention the fact that the MIMIC II is considered to be a "gold standard" dataset since it has been carefully collected from a single tertiary teaching hospital, and the dataset has been pre-processed to serve research purposes. Whereas the results shown here look promising we should take into account the fact that we are dealing with a dataset that it is not strictly the same as the ones found in real world environments. Nonetheless, there have been multiple research efforts to tackle this challenge in what is known as multitask machine learning [Car97].

7.1 Future Work

The work presented here can be extended and improved in numerous ways. The first and most interesting approach would be building a multiclass classifier based on the results of applying the pairwise classifiers and compute some kind of weighted majority vote based on some prior to determine which is the most likely code to have generated such note. For the four analyzed CVDs we would only need six classifiers. However, as we increase the number of codes the amount of classifiers would grow quadratically so another approach would be needed to directly build a robust multiclass discriminative model.

Another possible extension would be to generalize the performance with other kinds of diseases beyond heart conditions, such as broader complex chronic diseases (e.g. diabetes) and respiratory diseases. For these conditions there will be less data available (since we used the conditions that had the largest frequency), so a careful analysis of the performance could be carried out to determine if with less data or with different conditions, the performances still hold.

When dealing with the closest or most similar documents with the provided note, an arbitrary number of documents were retrieved regardless of their similarity. An unexplored approach would be filtering the retrieved documents with a given cosine similarity threshold. Nevertheless, it is not easy choosing a single cosine similarity threshold due to its dependency on the dimensionality of the vectors. Thus, it would be interesting to consider applying Principal Component Analysis to initially reduce the dimensionality of the vectors and then using cosine similarity and comparing the results with the ones outlined in this paper.

Finally, as we stated, the evaluation of retrieved relevant notes was not exhaustive enough and did not contain working health professionals. Carrying out health professionals evaluations of the model using some sort of agreement metric such as the kappa estimator, would be beneficial for determining the validity of the performed evaluations.

References

[Car97] Rich Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, July 1997.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[GWBV02] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[Joa97] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 143–151, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[Joa98] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142, London, UK, UK, 1998. Springer-Verlag.

[Joa01] Thorsten Joachims. A statistical learning learning model of text classification for support vector machines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 128–136, New York, NY, USA, 2001. ACM.

[LLGE10] Ying Li, Sharon Lipsky Gorman, and Noémie Elhadad. Section classification in clinical notes using supervised hidden markov model. In *Proceedings of the 1st ACM International Health Informatics Symposium*, IHI '10, pages 744–750, New York, NY, USA, 2010. ACM.

[PBM$^+$07] John P. Pestian, Christopher Brew, Pawel Matykiewicz, D. J. Hovermale, Neil Johnson, K. Bretonnel Cohen, and Wlodzislaw Duch. A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, BioNLP '07, pages 97–104, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[PHBS08] Serguei VS Pakhomov, Penny L Hanson, Susan S Bjornsen, and Steven A Smith. Automatic classification of foot examination findings using clinical notes and machine learning. *Journal of the American Medical Informatics Association*, 15(2):198–202, 2008.

[SRED09] Abdel Badeeh M Salem, Kenneth Revett, and El-Sayed Ahmed El-Dahshan. Machine learning in electrocardiogram diagnosis. In *Computer Science and Information Technology, 2009. IMCSIT'09. International Multiconference on*, pages 429–433. IEEE, 2009.

[TK02] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, March 2002.

[WWSEC15] Alyssa Weakley, Jennifer A Williams, Maureen Schmitter-Edgecombe, and Diane J Cook. Neuropsychological test selection for cognitive impairment classification: A machine learning approach. *Journal of clinical and experimental neuropsychology*, 37(9):899–916, 2015.

[ZRW$^+$14] Chengyi Zheng, Nazia Rashid, Yi-Lin Wu, River Koblick, Antony T Lin, Gerald D Levy, and T Craig Cheetham. Using natural language processing and machine learning to identify gout flares from electronic clinical notes. *Arthritis care & research*, 66(11):1740–1748, 2014.