Final Report

Project: Understanding the Determinants of Health

Team members: Gagan Kaushal, Chengyuan E, Tiam Moradi

1. Introduction

BU School of Public Health has access to clinician notes collected during routine medical visits at the Boston Medical Center. These notes are written free-text reports of visits capturing the clinician's findings, observations, and diagnoses. Our objective is to predict a diagnosis of substance abuse, alcohol use and tobacco smoking from the data in the notes by using techniques such as NLP and machine learning methods.

2. Methods

2.1 Machine Learning Method 1 (Medical Annotation Tool)

Technologies used: scispaCy, MedCAT, python

2.1.1 Data Preprocessing

For preprocessing the text data, we did the following: Lemminization, Tokenization, Removing stop words, and utilizing regular expressions. Lemminization: The process of grouping together the inflected forms of a word so they can be analyzed as a single item. For example, 'smoke and 'smoking' have the same root word 'smoke'. Thus, they can be analyzed as a single item. Tokenization: Breaking up the given text into small units called tokens. We are using HuggingFace's "ByteLevelBPETokenizer" (Byte level Byte Pair Encoding Tokenizer) Removing Stop words: This step is extremely important since stop words are the words that are commonly found (e.g. a, the etc.). These words occur frequently in most of the notes and thus don't contribute much in identifying and extracting the three phenotypes under consideration. Finally, we utilized regular expressions in order to remove certain symbols from notes e.g. punctuation characters - comma (,).

2.1.2 Model

The first approach that we explored involved MedCAT (5), which stands for Medical Concept Annotating Tool. This tool uses SpaCy, a natural language processing package for its name entity recognition. We leveraged this tool due to its pre-trained models from SciSpaCy (7).

In our project, leveraging transfer learning and using a pre-trained model is significantly important since we don't have a very large dataset.The model used for annotation is pre-trained as per the following specification: 34, 632 concepts, 96,529 names, 30,501 concepts that received training, 624,019 seen training examples in total, and 20.5 average training examples per concept. This tool also had two parameters required to generate the annotations: medcat.cdb, and medcat.utils.vocab. The first parameter is the concept database, which contains all the concepts of interest for a specific case of medical applications. Leverages large databases like the UMLS (Unified Medical Language System) (6). The second parameter is the vocabulary, which is used for spell checking and word embeddings.

In the below screenshot (fig. 1), is an example of the tool successfully extracted meaningful words from the patient note ***"Says no ETOH use, and says occasional alcohol use"***. From the note, the following meaningful words extracted were: ETOH, occasional, alcohol use. This model also demonstrates its capability of identifying slang for medical / chemical formulas: Ethanol i.e. ETOH with CUID C0001962. Further, our goal is to deal with negation cases in our notes and appropriately identify the phenotype label as yes or no. For example, "Denies drug use". Just because the word 'drug' is present in the note doesn't mean that we can predict the phenotype 'Substance abuse' as yes. The preceding word 'Denies' needs to be considered as well.

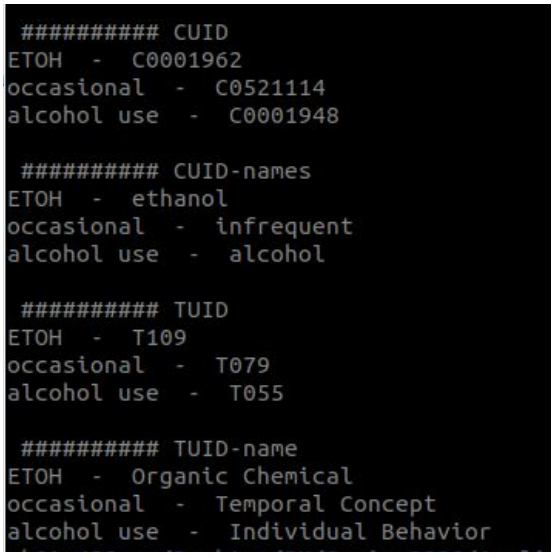


Fig. 1: Screenshot of MedCAT analysing the patient note

We further describe issues that we face with this method in the Discussion section.

2.2 Machine Learning Method 2 (Three standard ML models - Spacy)

Technologies used: scispaCy, python, sklearn, matplotlib, pandas, Linear SVM, KNN, Naive Bayes

2.2.1 Data Preprocessing

Preprocessing is a crucial part for this task. The original datasets of epic notes and logician notes have multiple columns which are not useful for the training purpose. So only the ID, notes, alcohol, drug use, smoking columns are extracted and stored in a new dataframe for later use. After splitting the new dataframe to 3 parts (alcohol, drug, smoking), delete NAN rows respectively for each part. Then Change 'Y' to 1 and 'N' to 0 in the target column for convenience of training and analysis.

2.2.1 Data Analysis

The histogram of review length is shown below, with the correlation graph between review length and alcohol, drug use, and smoking.

For the epic notes file:

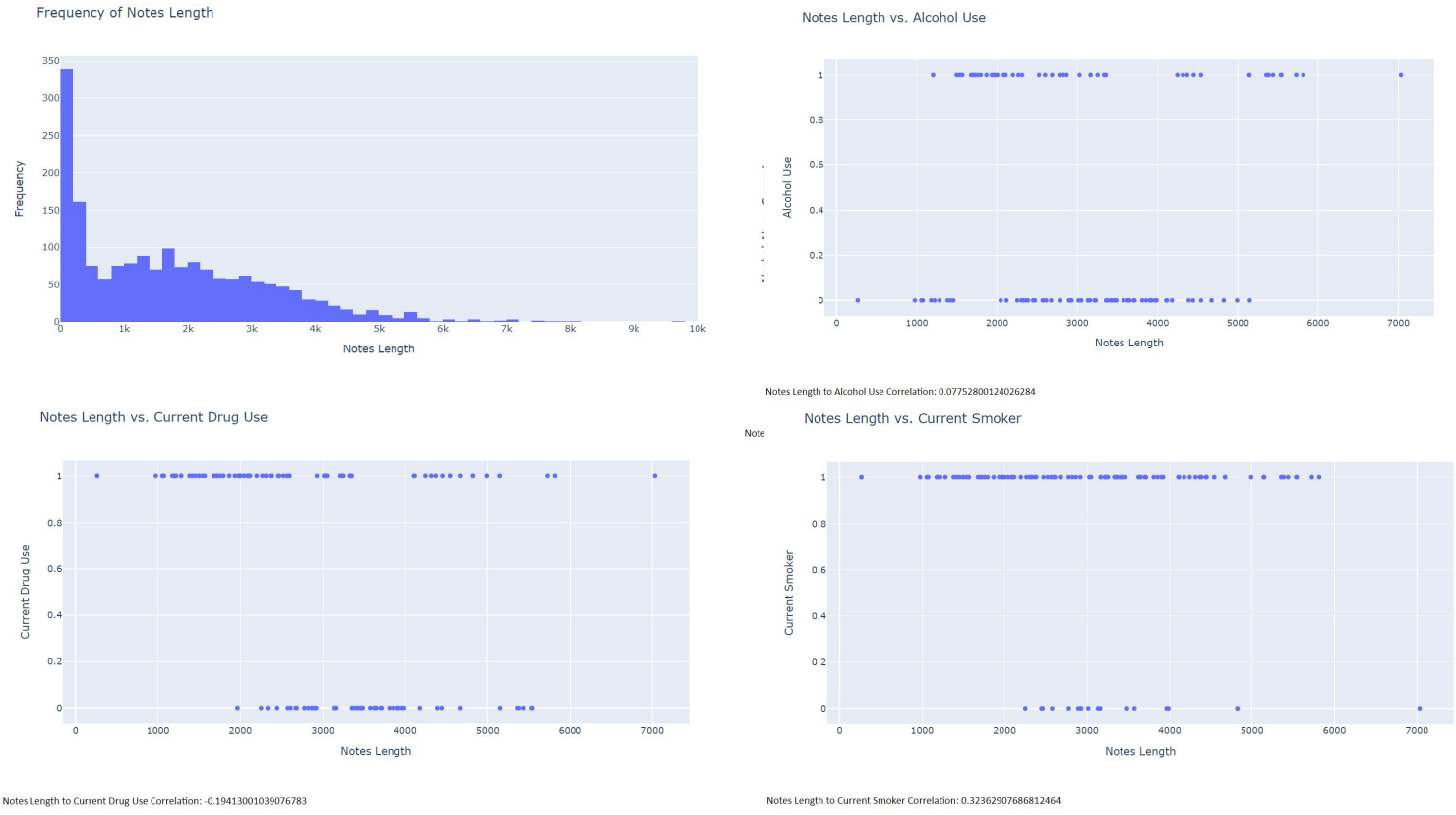
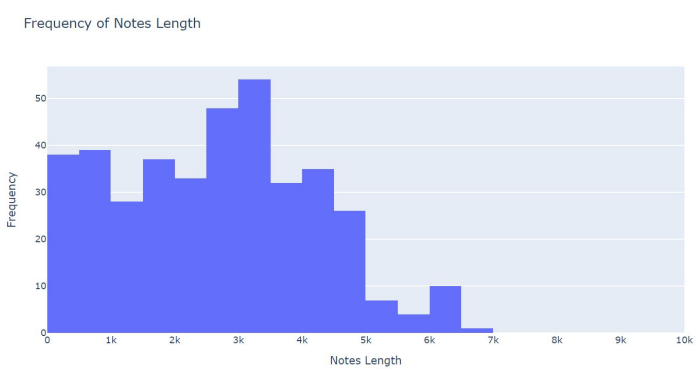
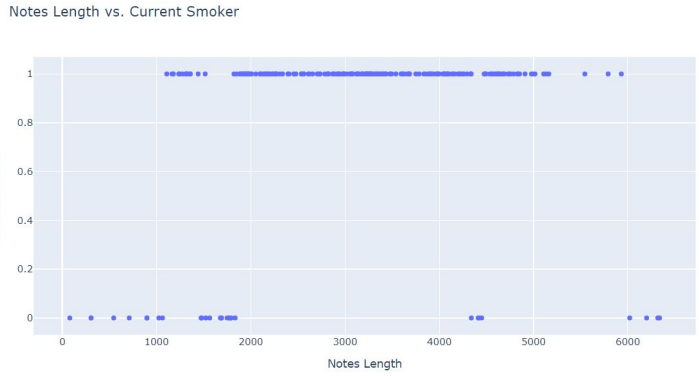


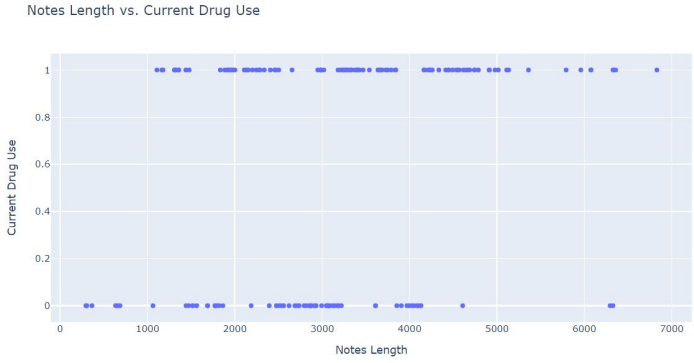
Fig. 2: Histogram of review length with the correlation graph between review length and alcohol, drug use, and smoking for epic notes

For Logician Notes file:

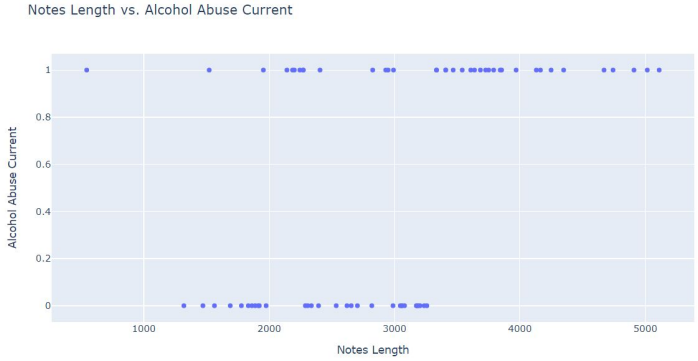




Notes Length to Current Smoker Correlation: 0.09145155190863861



Notes Length to Current Drug Use Correlation: -0.44486791020227934



Notes Length to Alcohol Use Correlation: -0.45119473852164943

Fig. 3: Histogram of review length with the correlation graph between review length and alcohol, drug use, and smoking for logician notes

The above figures showcase, mainly, following information:

1. The two histograms of the notes length provide a general view for the "notes" column condition. For the epic notes, the notes with shortest length have the highest frequency, and with the length increase, the frequency drops. For the logician notes, there's no such pattern, and the notes length around 3k has the highest frequency.

2. The other six correlation figures show the Notes Length to "target" Correlation, where "target" are alcohol, drug use, and smoking in each case. Because the dataset is not big, the correlation can be used as a tool for a larger dataset but not on this one since it's not credible enough to show a correlation.

2.2.1 Models

The main steps of the models are as follows:

1. Before training models, Spacy is used to extract the medical-related terms for more streamlined and useful text training data.(3) The spacy model used is "en_core_sci_sm". This model is used for locating and classifying entities (substance abuse, alcohol, and smoking) in an unstructured text (Logician and Epic notes) to extract the medical texts from the original notes.
2. Replace the original column of notes by these extracted texts and create a new dataframe.
3. Input these medical texts to the tf-idf of the unigrams and bigrams of the notes.(4)
4. Use the result from step 3 to be the input for each of the 3 classification algorithms: 1. Linear SVM (support vector machine) 2. KNN (k = 2) 3. Naive Bayes functions to predict alcohol, drug use, and smoking labels. (10)(11)

We trained the models with a training set size of 75 percent of the dataset, and had a test set size of 25 percent of the dataset.

The general comparison between the 3 models are as follows:
When comparing KNN with SVM, SVM takes care of outliers better than KNN. However there are not significant outliers in this task, so the difference is not clearly shown. SVM outperforms KNN when there are large features and lesser training data. In this task the situation happens to be so, all the cases except alcohol use prediction in epic notes, the linear SVM method has a similar or higher accuracy than KNN methods as shown in the result part.
When comparing KNN with naive bayes, Naive bayes is much faster than KNN due to KNN's real-time execution.(9) Our training data size is not huge, so the time factor could be ignored. However if our approaches are potentially used for much larger dataset, the efficiency factor should be taken into concern.

3. Results

a. The bar graph shows the frequency of different labels / unique features

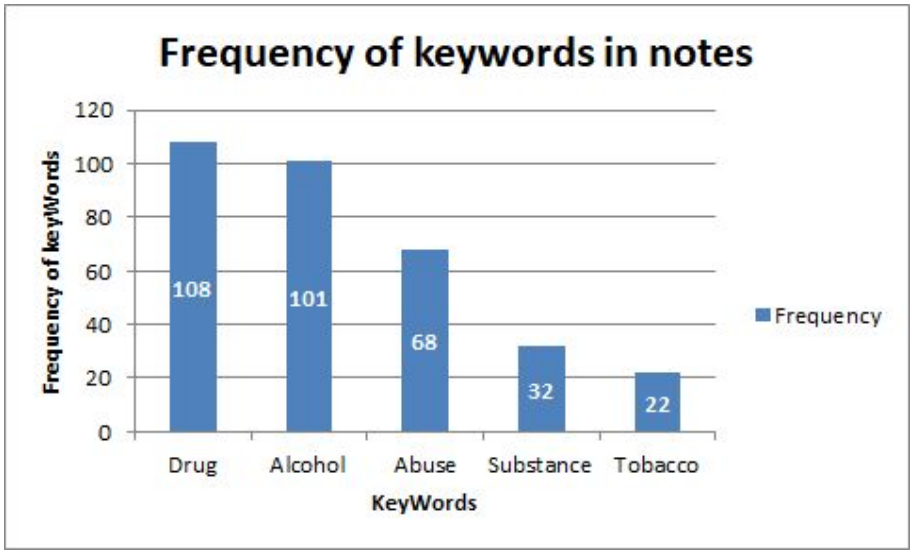


Fig. 4: Bar Graph showcasing frequency of keywords in notes

The **'drug'** keyword occurs the most in the notes. So, as per preliminary investigation, we suspect that there are more patients affected by '**drug**' as compared to **'Tobacco'**

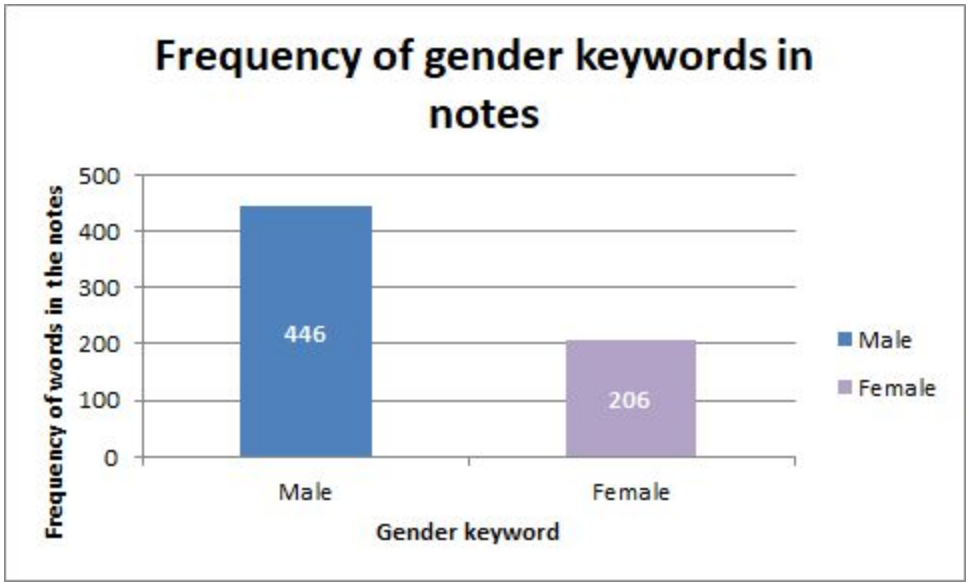
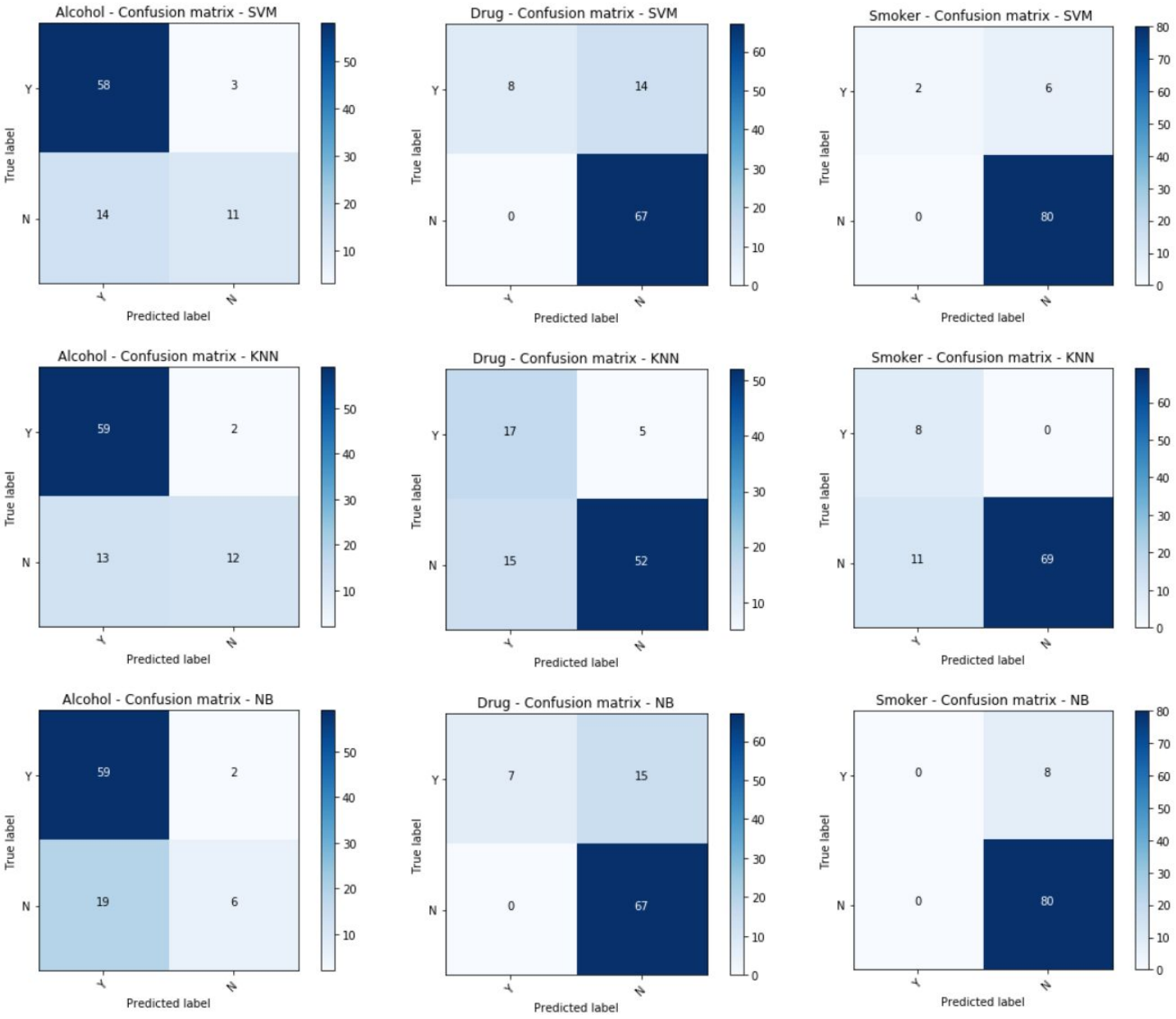


Fig. 5: Bar Graph showcasing frequency of gender keywords in notes

b. **The accuracies are as follows, which are reasonably high and satisfiable.**

For the Epic notes file:



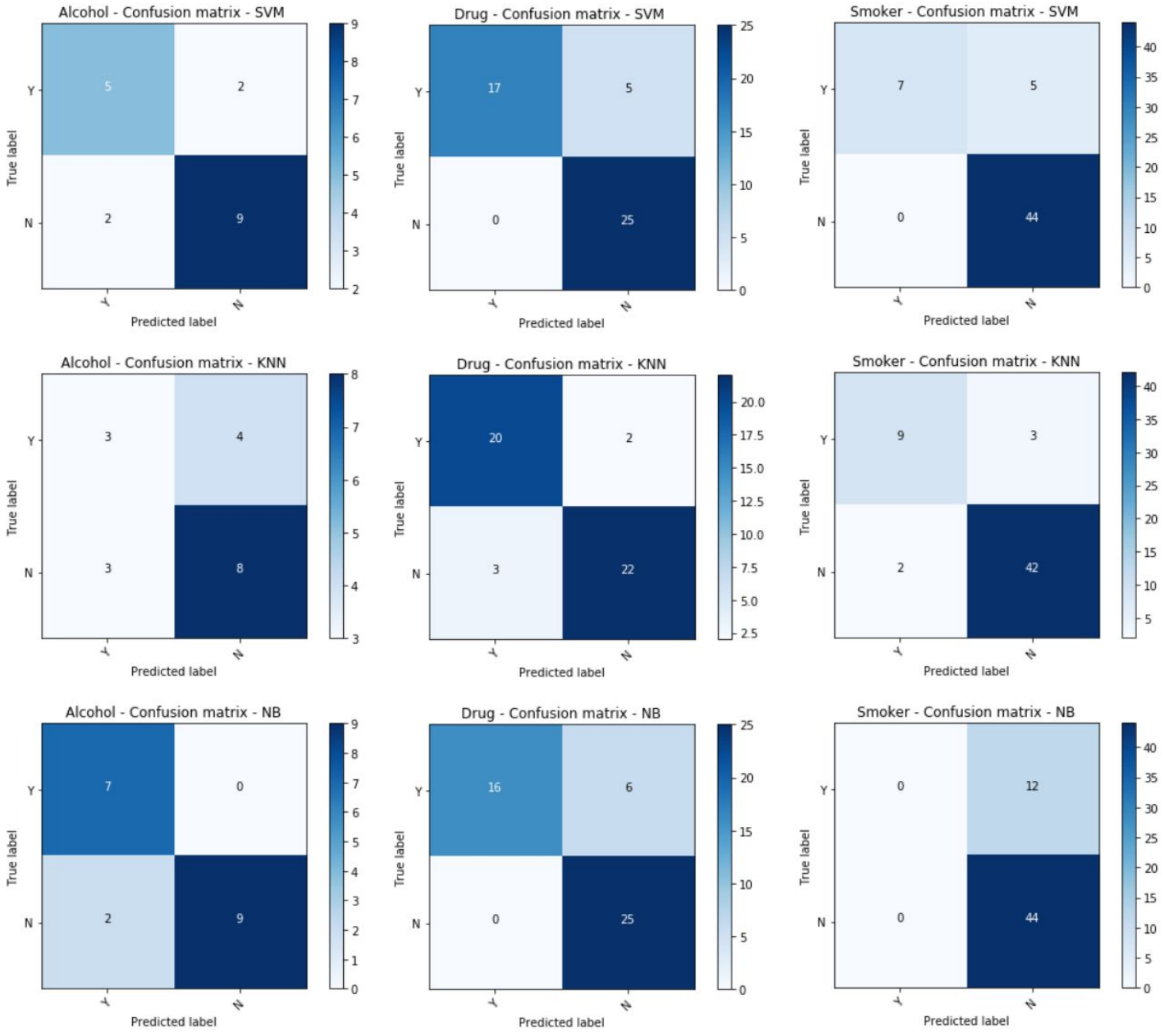
Linear SVC: 0.8023255813953488
kNN: 0.8255813953488372
Naive Bayes: 0.7558139534883721

Linear SVC: 0.8426966292134831
kNN: 0.7752808988764045
Naive Bayes: 0.8314606741573034

Linear SVC: 0.9318181818181818
kNN: 0.875
Naive Bayes: 0.9090909090909091

Fig 6. Confusion Matrices showcasing results for epic notes

For the Logician notes file:



Linear SVC: 0.7777777777777778
kNN: 0.6111111111111112
Naive Bayes: 0.8888888888888888

Linear SVC: 0.8936170212765957
kNN: 0.8936170212765957
Naive Bayes: 0.8723404255319149

Linear SVC: 0.9107142857142857
kNN: 0.9107142857142857
Naive Bayes: 0.7857142857142857

Fig 7. Confusion Matrices showcasing results for Logician notes

4. Discussion

As we are working on this particular project, we have experienced several challenges along the way. Initially, getting access to the data was a challenge. Upon completing the courses required to get access to the data, we were only given access during the middle of March. This gave us less time to understand our data well. After we were given access, we realized that we do not have a lot of data to work with. This meant that we were not able to train a model from scratch but instead utilize a pre-trained model and perform transfer learning. Within the limited time that we had, we also noticed that the majority of the Logistican Notes were sparse: out of the 392 notes present in the dataset, only 92 instances had notes we were able to use for either of our methods.

Another issue that we initially ran into was the structure of the labels of the data. One of the methods we believe would perform significantly well was Name Entity Recognition. This would require that our data would be labeled for every word in each note, rather than the whole label denoting the note itself. Initially, we believed that using the MedCAT tool would mitigate this issue, however, new issues came to fruition because of it ; specifically when it came to utilizing deep learning methodologies. We initially tried to use an algorithm called BlueBert (2) that was pretrained on clinical text from PubMed corpus. Since it was written with legacy code, we spent the majority of our time debugging the model in order to have it running. After that, we realized that our data was not of the same format as the training data. Also the output of the BlueBert model was not necessarily the same as our objective desired. When wanting to train a SpaCy model from scratch adding a new entity, or even adjusting a pre-trained model, our model was not of the proper format either. The entities that we extracted from the MedCAT tool did not contain the indices of the entity that's being extracted. This means that we would have to manually annotate a few thousand notes of all the entities that were present in order for the SpaCy models to have been successful; we unfortunately did not have the time for that. Finally, there is no golden standard to compare our results to. As this is a field of research, we do not have a baseline for the performance of our work.

As for next steps with this project, one suggestion would be to reformat the entities that were extracted into the format that SpaCy requires and then train a model to find entities in new clinical text. There were also other Clinical Text tools that were considered, but not looked into in-depth due to the lack of time, but one can try using the CliNER model(8) for clinical text entity recognition tools, however the same issue arises with needing the indexes of each entity. Another method that one might be a rule based algorithm. One reason being that some of the NER models are not able to capture the negation of text, an example being: "no pain experienced ". We also saved the entities for these next steps to be more efficient and easier for the next individuals who work on this project.

References

1. Tarcar, Amogh Kamat, et al. "Healthcare NER Models Using Language Model Pretraining." *HSDM 2020 Workshop on Health Search and Data Mining*. Vol. 1. 2020.(https://arxiv.org/ftp/arxiv/papers/1910/1910.11241.pdf)

2. https://github.com/ncbi-nlp/bluebert

3. spaCy 101: Everything you need to know https://spacy.io/usage/spacy-101

4. Document Classification Part 2: Text Processing (N-Gram Model & TF-IDF Model)

5. Zeljko Kraljevic, Daniel Bean, Aurelie Mascio, Lukasz Roguski, Amos Folarin, Angus Roberts, Rebecca Bendayan, Richard Dobson. *MedCAT -- Medical Concept Annotation Tool,* arXiv:1912.10166 (https://arxiv.org/pdf/1912.10166.pdf)

6. Unified Medical Language System https://www.nlm.nih.gov/research/umls/sourcereleasedocs/index.html

7. scispaCy https://spacy.io/universe/project/scispacy

8. CliNER model http://text-machine.cs.uml.edu/cliner/

9. Comparison of Classification Methods Based on the Type of Attributes and Sample Size, Reza Entezari-Maleki, Arash Rezaei, and Behrouz Minaei-Bidgoli https://pdfs.semanticscholar.org/2ce0/664cfcb32461b900dd9e889cbbb2259c503e.pdf

10. Machine Learning Basics with the K-Nearest Neighbors Algorithm https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

11. SVM using Scikit-Learn in Python https://www.learnopencv.com/svm-using-scikit-learn-in-python/