Springboard Data Science Capstone 2 Report

IDENTIFY AND ALERT OF POTENTIAL GOUT FROM A PATIENT'S 'CHIEF COMPLAINT'

MICHELLE IDE

SPRINGBOARD Data Science Capstone Project

Contents

INTRODUCTION

Business Problem

Determining proper diagnosis during a patient's emergency room visit is complicated by communication barriers, time constraints, and often a lack of history on the patient. Therefore, the patient's complaint is an important piece of information the physician uses in the process of diagnosis without the luxury of a previous relationship with the patient. When people are from different regions, different backgrounds, in a high-pressure environment, the meaning in their words can be misunderstood adding to the difficulty and stress involved in getting the correct diagnosis. Machine Learning has proven to provide highly accurate predictions in Natural Language Processing. This project will provide a tool that can alert physicians to statistically probable disease states based on descriptions provided by the patient as their chief complaint. In this particular case we will predict 'Gout'.

Solution

This project provides a communication bridge between physician and patient, improving diagnostic accuracy by providing an alert to statistically probable disease states based on a patient's 'chief comoplaint'.

Audience & Stakeholders

Specific to healthcare, this tool is used by physicians with additional stakeholders that include hospital administrators and emergency department leadership.. It should be noted this tool should only be used to alert physicians to potential diseases and not used as a diagnostic tool as it is not FDA approved for diagnostic purposes.

Description

This project uses Natural Language Processing (NLP) techniques to develop a supervised classification model to predict if a patient is experiencing symptoms of Gout based on their Chief Complaint as recorded by hospital staff. A similar solution from 2020 can be found at: https://physionet.org/content/emer-complaint-gout/1.0/. In this project we explore improvements through tuning and improved models since the time of the previous research.

DATA

Source

Data for this project was extracted from an MIT medical information mart for intensive care, specifically the MIMIC III database. This database contains deidentified health-related data from actual patients admitted to critical care units of Beth Israel Deaconess Medical Center and was collected during years 2019 and 2020. Access to this data requires PhysioNet Credentialing specific to regulations around use of patient data. Due to the nature of the data, the corpus details will not be shared or displayed publicly. Access to the data requires access permissions which can be requested from the PhysioNet site located at: https://physionet.org/content/emer-complaint-gout/1.0/.

Scope

The scope of this project is corpora from the Deep South. The demographics of the population from which they were derived are 54% female, and 46% male, 55% Black, 40% White, 2% Hispanic, and 1% Asian. Age distribution was 5% between ages 1-20 years, 35% between ages 21-40 years, 35% between ages 41-60 years, 20% between ages 61-80 years, and 5% between ages 81-100 years.

Collection

The patient's chief complaint was collected by hospital personnel at the time of admission into the ER and recorded using one of two software systems, CareVue and MetaVision from years 2019 and 2020. The resulting diagnosis was recorded by a panel of emergency room physicians in determination of Gout and recorded as a predicted outcome (Predict). If the patient was diagnosed with Gout they were referred to a Rheumatologist which further confirmed the diagnosis (Consensus).

Description

The data consists of two different methods of de identification, SYNTHETIC and REDACTED. The synthetic data was deidentified prior to extraction using the Bert and Albert NER algorithms to meet the HIPAA Safe Harbor specifications. Therefore the SYNTHETIC data files were selected for this project to eliminate the personal information contained in the REDACTED version. Two csv files were exported and included 2019 and 2020 data. Below is a description of the data extracted.

Cleaning

The 2 files extracted from the MIMIC III database represented 2 years, 2019 and 2020. These files were identical in format and contained the patient's chief complaint with the resulting diagnosis in two columns, Predict and Consensus. The Predict column was determined by a ER physician while the consensus was reviewed and confirmed by a Rheumatologist.

The two files were combined into 1 dataframe.

Data Description

- 2 csv files
  - 2019 : 300 records
  - 2020 : 8037 records
  - Identical layouts and formats: all text, 3 columns

- 3 Columns: ["Chief Complaint", "Predict", "Consensus"]
  - **Chief Complaint:**
    - text format
    - up to 282 Chars
    - nurse recorded patient complaint
  - **Predict:**
    - text format
    - single char ('-','U','Y','N')
    - prediction of Gout by the ER Physician
  - **Consensus:**
    - textformat
    - single char ('-','U','Y','N')
    - determination of Gout by the Rhuematologist

```
- : Null
U : Unknonw
Y : Yes
N : Gout
```

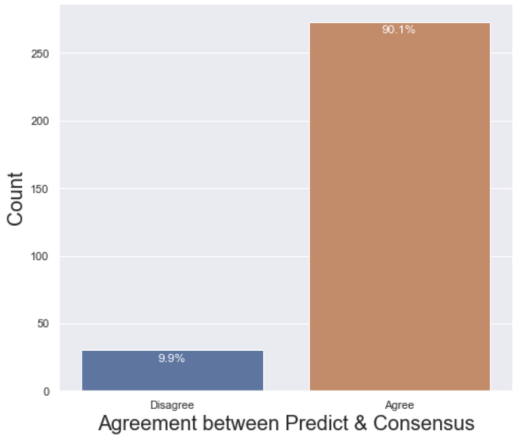
```
Shape of new file (8437, 3)

 Predict Column
 -       2
N    8168
Y     111
U     156
Name: Predict, dtype: int64


Consensus Column Values
 -    7976
N     350
Y      95
U      16
Name: Consensus, dtype: int64
```

The shape of the resulting dataframe and values in the proposed targets are to the left. The consensus column is preferred as it contains results from a Rheumatologist, clear confirmation of Gout. This column contains a total of 303 records with values Y/N.

	Chief Complaint	Predict	Consensus
0	"been feeling bad" last 2 weeks & switched BP ...	N	-
1	"can't walk", reports onset at 0830 am. orient...	Y	N
2	"dehydration" Chest hurts, hips hurt, cramps P...	Y	Y
3	"gout flare up" L arm swelling x 1 week, denie...	Y	Y

Feature Selection: This project required a single target, the Consensus column was selected.

Missing Values:



Before filling the null values in the consensus with the predict values, a comparison was performed to review agreement between Y/N values, if there were values where Predict was incorrect, how frequently were they in disagreement.

Comparing the Consensus and Predict values, 303 rows existed where both contained values of Y/N. We see about 10% disagreement which makes sense, if it were over 50% I would be concerned.

- Total Records: 303
- Agree: 273
- Disagree: 30

Approximately 10% of the time the ER predicted incorrectly according to the consensus. Of the 30 incorrect values were 22 False Positives vs 8 False Negatives.

- False Positive: 22
- False Negative: 8

This means, by using Predict to fill nulls in the Consensus column we may be introducing false values at a rate of 10% overall, with mostly false positives. For this project the 10% was acceptable.

The ratio of 30 misses by ER doctors out of 303 records gives us a comparison to target with the models. Our observations tell us the physicians were inaccurate 10% of the time. Another important point is that only 2.7% of the results were false negatives, and 7.3% false positives. This is out of a small portion of the data but gives us an approximate base to measure our results.

The missing values in consensus were filled with the Predict column.

	corpus	target
0	"been feeling bad" last 2 weeks & switched BP ...	N
1	"can't walk", reports onset at 0830 am. orient...	Y
2	"dehydration" Chest hurts, hips hurt, cramps P...	Y

Any remaining null values ('-' or 'U') were removed. The resulting data frame had a single 'Corpus' column containing the patient's chief complaint and a single column "Consensus" with the target value of '"Y" or "N" for the question "is the patient complaining of gout?".

Preprocessing

The following preprocessing steps were performed on the corpus in preparation for modeling:

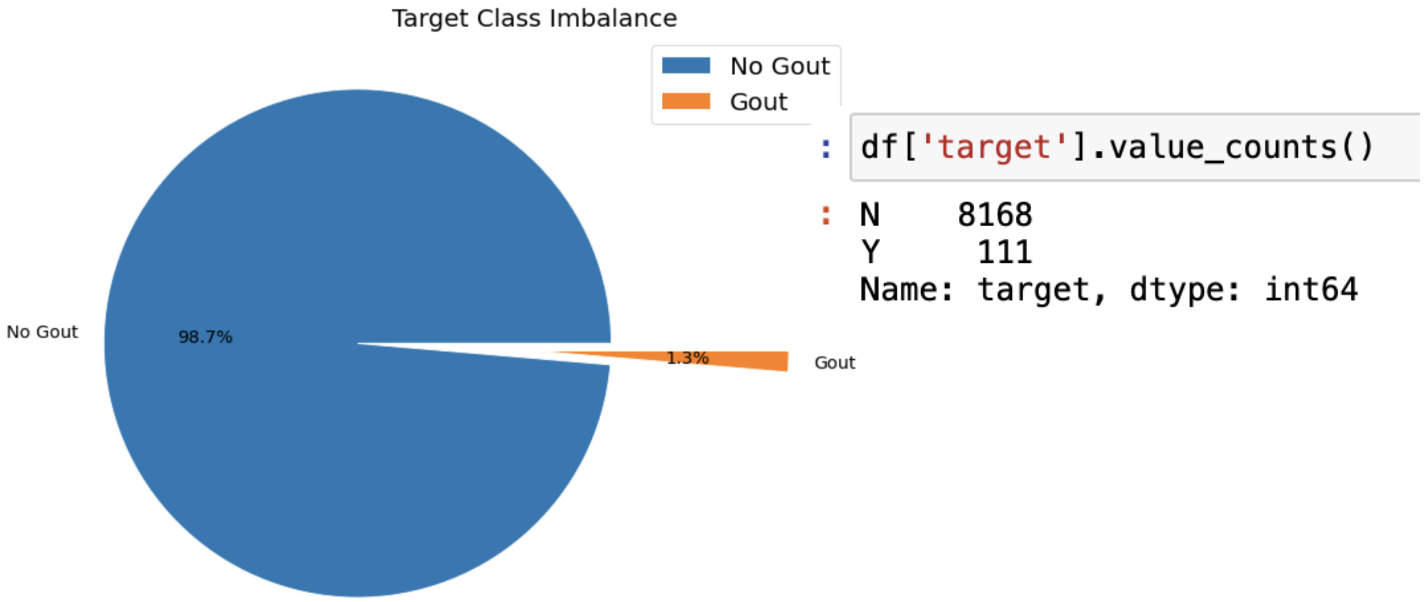
- ***Clean Text:*** remove brackets [ ], punctuation, embedded digits, quotes, and newlines.
- ***Remove Key Term:*** The word 'gout' was removed from the corpus to prevent cheating.
- ***Tokenize:*** Each patient corpus was tokenized by sentence.
- ***Stopwords:*** Stopwords were removed using nltk english library method
- ***Lemmatization:*** Lemming was performed using WordNetLemmatizer from nltk library
- ***Stemming:*** Stemming was also performed using PorterStemmer from nltk library

ANALYSIS

Class Balance

The following graphs demonstrate a significant class imbalance in the target which was addressed in the modeling phase and is described further in modeling.

Class Distribution is significantly imbalanced.

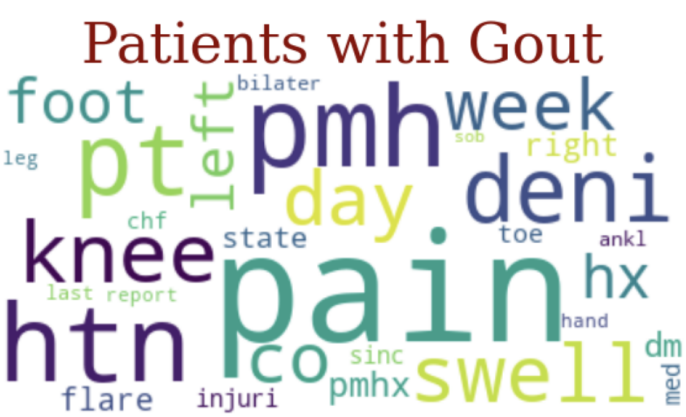



```
: df['target'].value_counts()

: N    8168
  Y     111
  Name: target, dtype: int64
```

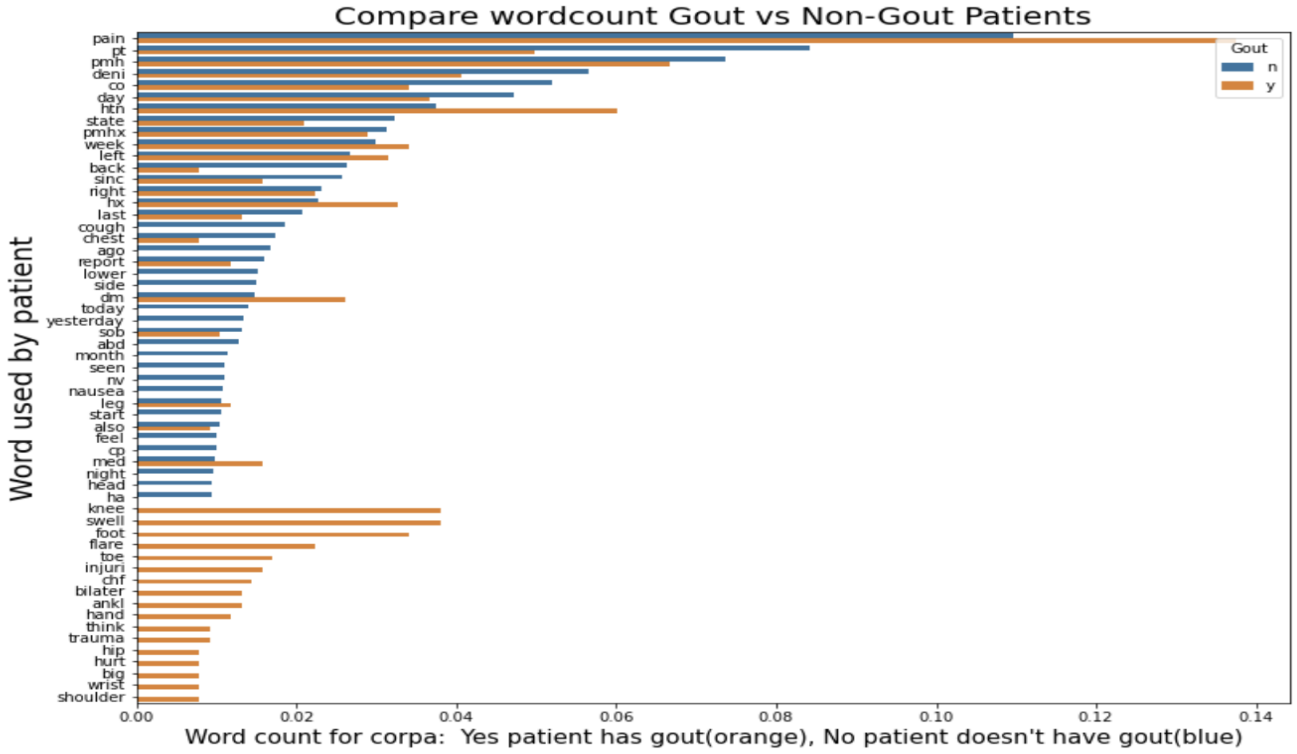
With **111** records in the Yes category for Gout out of **8279** total records, class imbalance is a concerning issue. In the method section this is addressed using oversampling.

Feature Correlation

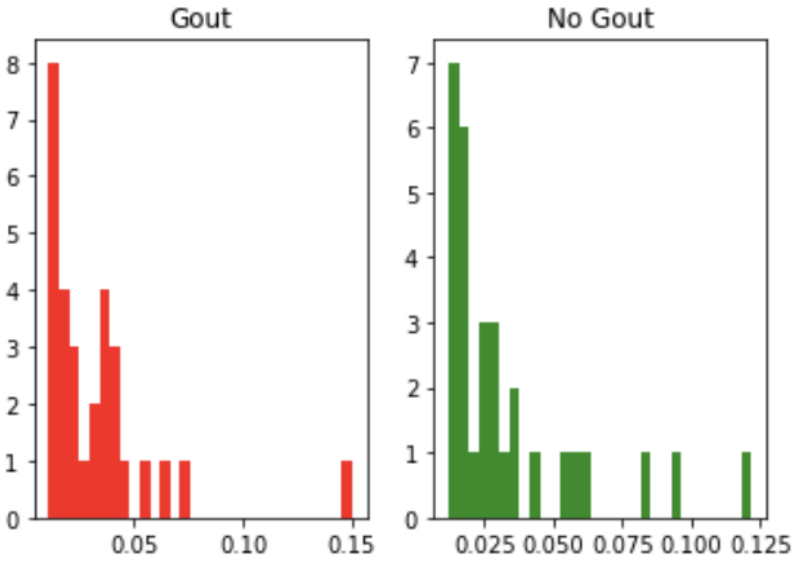
The word clouds below represent the top 20 most frequent word count by class, supporting a correlation between target and corpa. In patients with Gout we see words like 'flare' 'swell' and 'leg', in comparison the top 20 words include 'chest' 'side' and 'sob'(short of breath) in patients that do not have gout.



Looking at the top 40 words (below) for Gout (y) and non-Gout (n) patients there are many terms (words) that do not overlap in the description of gout by a patient (see the orange at the bottom). However, looking at the top of the chart, the overlapping blue and orange display a large decision threshold. We will address this through optimization techniques..

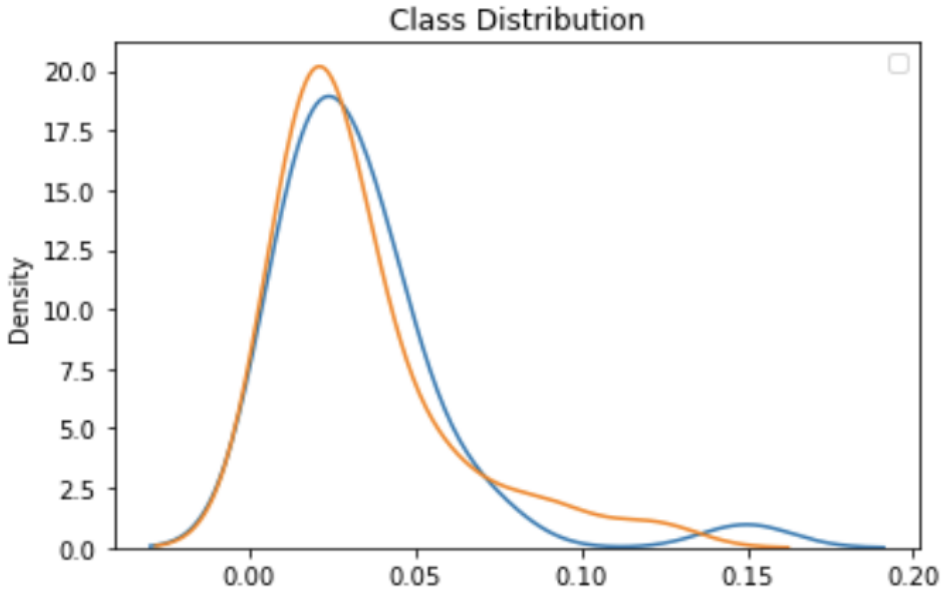


Target Correlation



The word count of each class was examined to the left. We can see the distribution for each class is very similar. I looked for similar distributions to tell me there was nothing skewed within the Gout data such as unusual number of words or an uneven distribution. Basically the token distribution and sentences are similar so comparisons will be based on the words.

Comparing the two KDE distributions below is encouraging, the corpus for gout patients is still similar to non-gout patients and will provide good training comparisons. In addition, there is a nice normal distribution curve indicating a linear-based model would perform well.



MODELING

Model Selection

Six classification models were selected for training and testing. Each for it's known performance value and unique algorithm difference from the rest.

- Logistic Regression - Linear
- Naive Bayes - Probabilistic
- Support Vector Machine - Flexible Boundry
- Gradient Boost - Performance Boost
- Random Forest - Decision Tree
- KNearest Neighbors - Geometric

Each model brings a unique algorithmic strategy. A linear based Logistic Regression model was tested first based on the normal distribution of word counts. Naive Bayes was included in testing for it's known superior performance on binomial classification of NLP problems. Support Vector Machine also was included due to the class imbalance that exists to start. Gradient Boost is newer and provides superior performance in some cases due to it's 'boosting' algorithm. Random Forest is a well-established model in wide circulation currently in real-world cases and is often preferred by experienced data scientists, therefore it is included in testing. KNN provides a less CPU demanding algorithm known to perform sufficiently well for binomial target cases in classification.

Optimization

Without optimization he models performed between 54-67%.

Several key methods were utilized to reduce potential bias and improve accuracy :

- Stratify the data
- Normalize the data
- Weight the keywords
- Combination of upsample/downsample minority/majority class

Vectorization: Looking at the wordcloud and key word count we could see there existed important differences in specific terms, keywords that were more likely associated with gout and keywords clearly indicating this is not gout. From these words we see the importance of a single word in determining the target. Therefore, the TF-IDF method was selected to vector this data due to the method's ability to 'weight' these keywords and place more value on their importance. This step eliminated the need to 'normalize' the data when addressing potential bias.

Key Term Removal: For this project we are providing a tool for determination of potential 'gout' in patients who do not already know they have gout, new diagnosis. Many of the chief complaints in the 'gout' category contained the word 'gout' in their chief complaint. These were removed to prevent this 'cheat' by the algorithm.

Class Imbalance: There existed an almost 1:100 ratio between the 2 classes, not great for either upsampling or downsampling. Upsampling would require many repetitions of the same nlp data, basically saying "hello" over and over again for example, clearly biasing the training model to a very limited set of sentences. Downsampling would limit the amount of data available for training to 300. Since this is an nlp project, larger amounts of data are important for confident results given the variety of disease states and patient complaints that exist in the real world. To address these constraints a combination of upsampling and downsampling was performed.

Regularization: With a minority class containing than 2% of the total data, bias is a concern during training. In addition to the combined up/down sampling technique a cross validation method wrapped around the sampling step to prevent overfitting (high variance) with 10 folds selected given over 8,000 pieces of data.

Accuracy: The data was split for training and validation by a 70:30 ratio. 70% used for training and 30% for testing and accuracy measurement. As with any healthcare tool, the top priority of this project is to reduce the false positives but, more importantly, prevent false negatives. Measuring accuracy and success is therefore performed using the ROC AUC and Confusion Matrices to compare the false target rates. Note we are attempting to improve on current ER rates which already have an accuracy of 90%, with the 'misses' consisting mostly of false positives, therefore the usefulness in this model is measured by the ability to prevent false negatives with the secondary goal of reducing false positives.

RESULTS

Goal Review

As a healthcare project, the goal is to first minimize false negatives (sensitivity) while maximizing the overall accuracy. In the overview of results that follow we are therefore seeking the best sensitivity (fewest false negatives) without approaching an intolerable sensitivity (false positives).

For this project we have only the original results to compare the error rates of diagnosis by the ER. This does not mean all visits to the ER have this rate, it is simply a comparison to determine if an improvement on error rates can occur with the use of ML techniques. The original results in which the ER physicians determined 'gout or 'no gout' demonstrated an overall error rate of 10% with sensitivity errors of 2.7% False Negatives and the remaining 7.3% False Positives. I sought to improve this error rate (>10%) and reduce False Negatives (<2.3%).

Comparisons

1) Without the word (term) 'gout' in the chief complaint.

While reviewing the chief complaints it was clear many of the true positives contained the word 'gout' in them. I wanted to see how well the results were based strictly on the patients complaint such as pain or swelling and therefore removed the word (term) from the complaints (corpus).

2) With the word (term) 'gout' in the chief complaint.

To do a true comparison to the original results I also ran the models as was provided to the ER physicians, which did include the term 'gout'. The results are summarized below. However it should be noted a hold-out set was tested on the models and accuracy dropped 10%. These models would require more training therefore to perform consistently well.

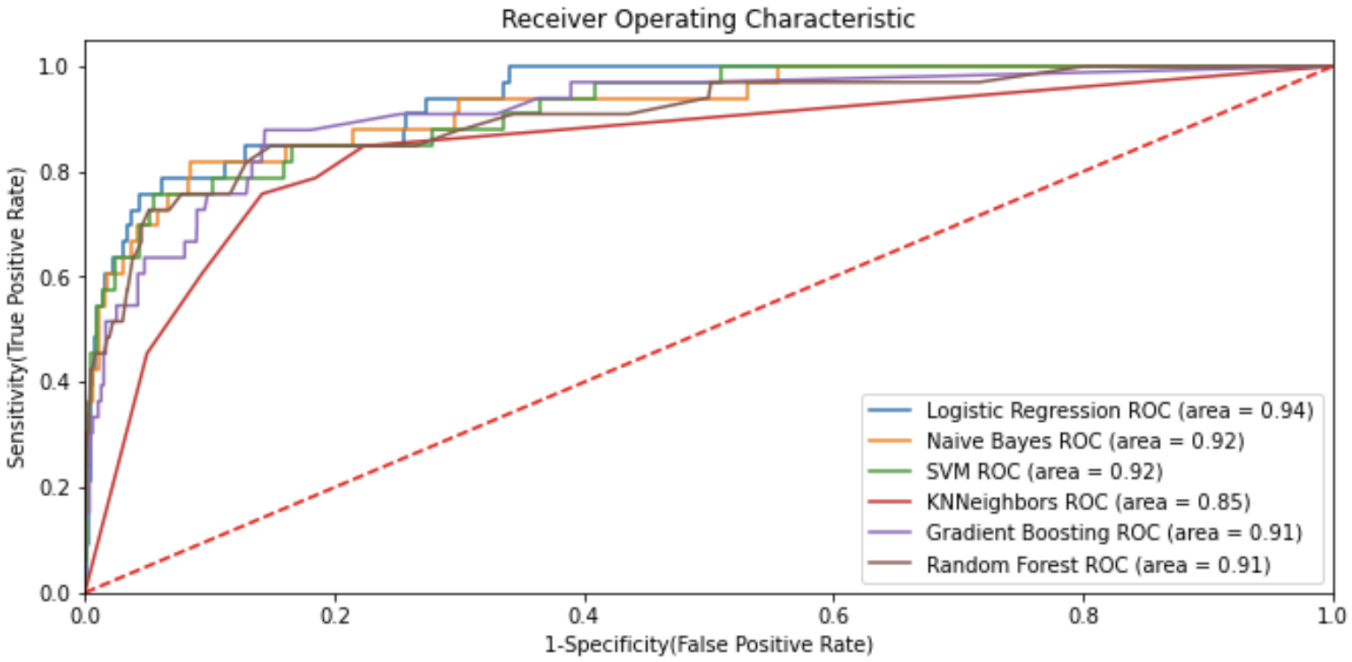
3) Bert Pre-Trained Models

I therefore ran a BERT pre-trained model which does contain much more data resources to train, the results are a return to >95% accuracy as summarized below.

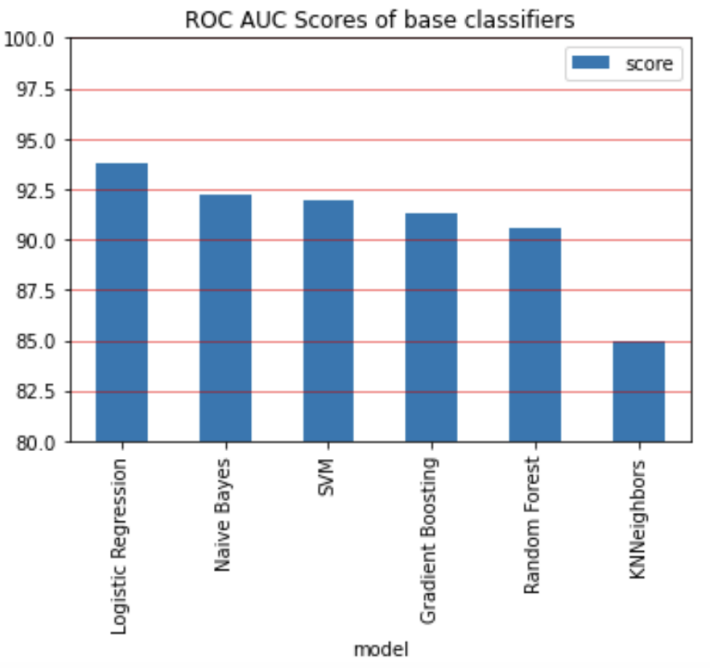
Results

1) Without 'gout' term: Prior to hyperparameter tuning

Below is the ROC AUC curves and scores for all optimized models, demonstrating the error rates:

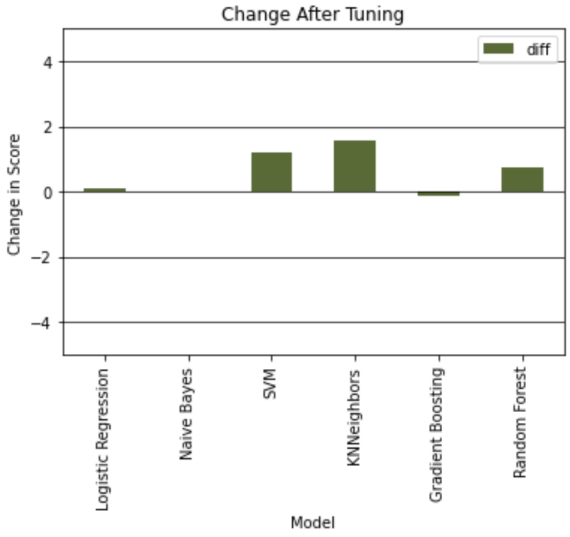


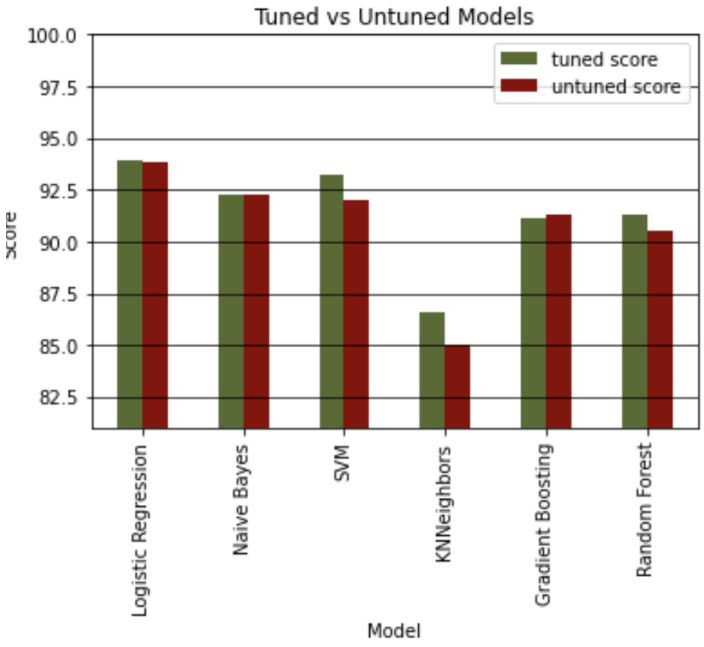
model	score
Logistic Regression	93.794682
Naive Bayes	92.226345
SVM	91.996289
KNNeighbors	84.988868
Gradient Boosting	91.289425
Random Forest	90.529375



1) Without 'gout' term: after hyperparameter tuning

After tuning the hyperparameters the following changes resulted.

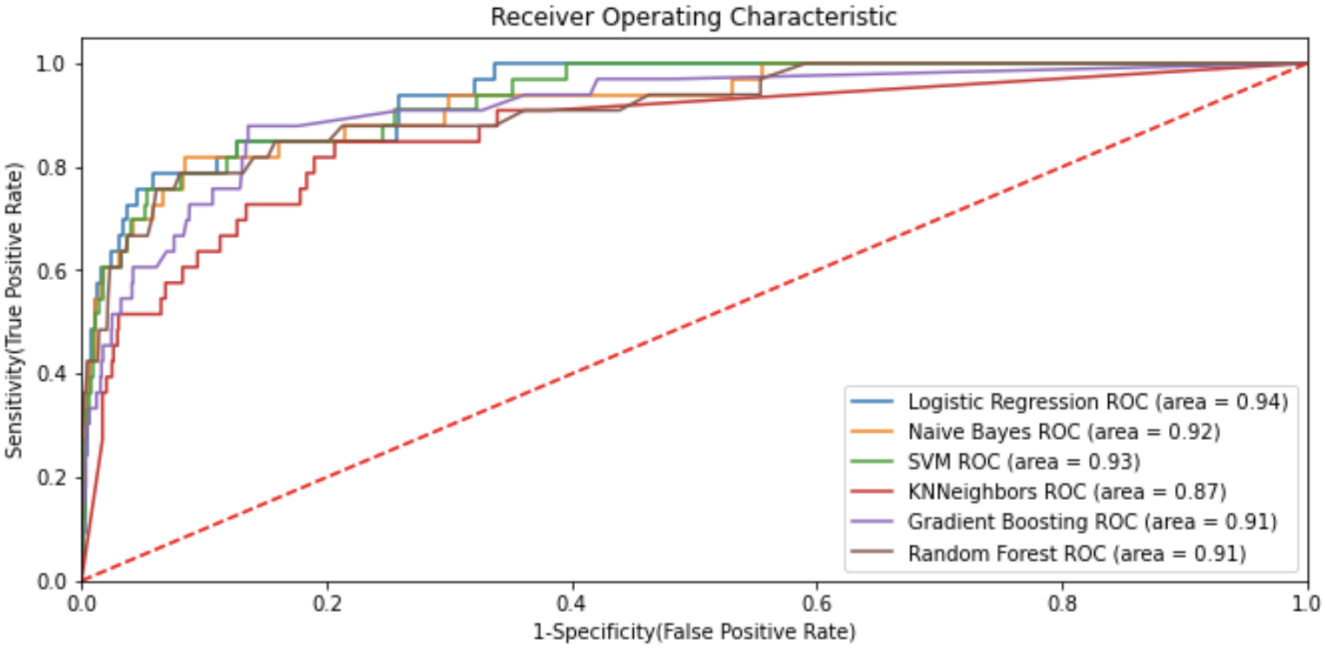




The difference in ROC AUC scores after tuning are in the plot above.

Left a comparison of scores before and after tuning.

Below we see the ROC AUC of all the models, demonstrating the overall performance measure by number of Accurate predictions that fall beneath the curve. All performed above 90%, meeting the goal of >90% (<10% overall error) except the KNN which appeared to fail the overall accuracy rate goal.

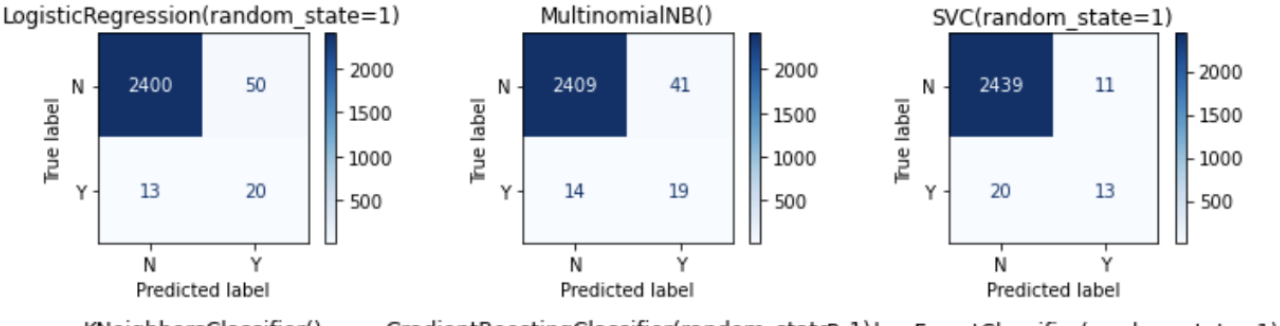


model	tuned score	untuned score
Logistic Regression	93.897341	93.794682
Naive Bayes	92.226345	92.226345
SVM	93.209647	91.996289
KNNeighbors	86.552257	84.988868
Gradient Boosting	91.137910	91.289425
Random Forest	91.288806	90.529375

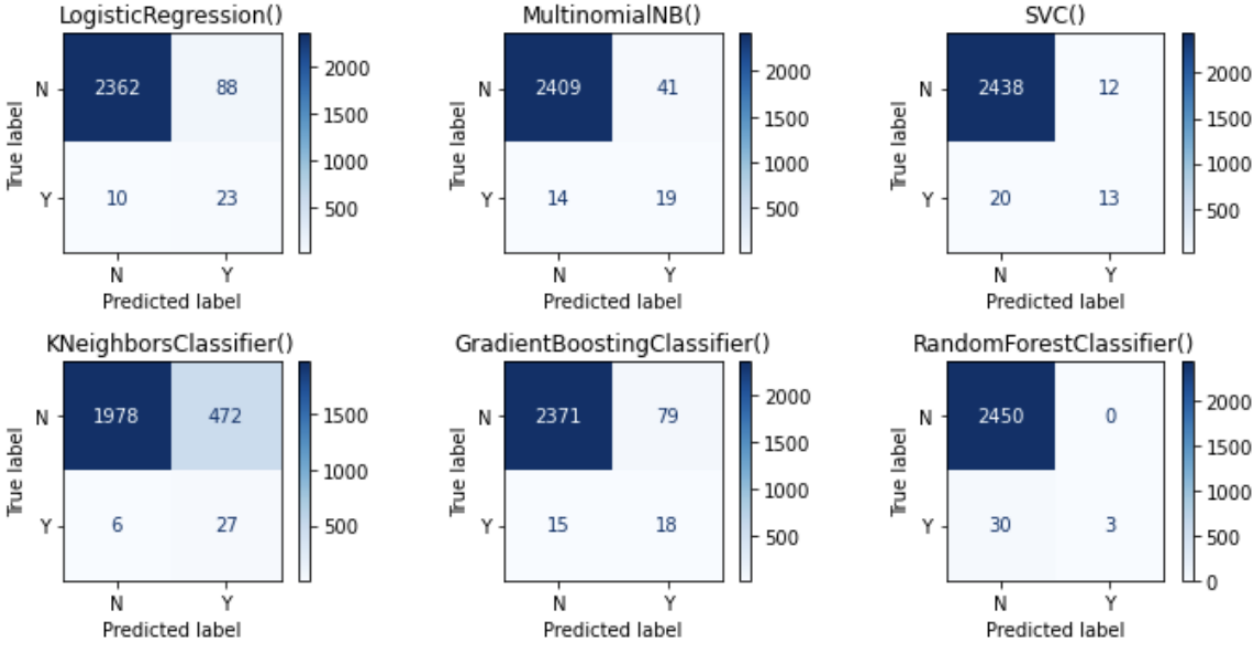
To the left is the summary of the ROC AUC scores before and after tuning.

The below Confusion Matrix provided a measure of selecting the best model for our goals. The test data consisted of 2483 data points and would require <67 False Negatives with <248 overall False Counts to meet the goal.

Prior to Tuning



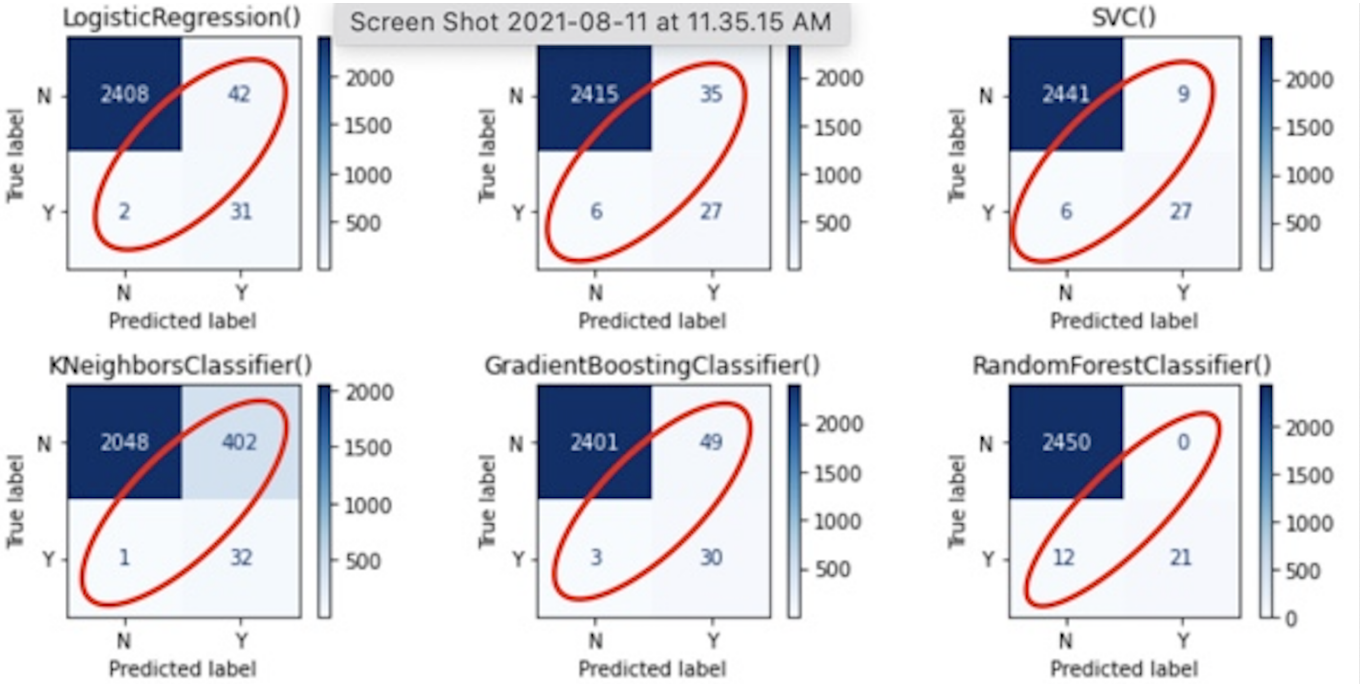
After Tuning



The top 3 specificities are KNeighbors, Logistic Regression, and Multinomial. Of the three the best sensitivity (81%) is seen in the KNeighbors model but with an overall accuracy rate of 80%. Logistic Regression demonstrates the next best sensitivity of 69% and overall accuracy of 96%. For our goal the LR model is the best at this point. Let's compare apples to apples however, we want to improve on the ER physicians predictions and need to compare the same data sets using these models to select the appropriate model. Below are the results of the models using the same 'Chief Comoplaint' (with 'gout) that the ER physician used to determine proper diagnosis. We compare these results to determine true improvements.

2) With 'gout' term: after hyperparameter tuning

So let's look at the confusion matrix for 2) with the 'gout' term included, just as the ER physicians experienced, let's compare the model performance without tampering with the data and see if the models still perform similarly.



Interestingly the models demonstrated many more false positives but with improvements in sensitivity. The top 3 specificities are KNeighbors(97%), Logistic Regression(93%), and Gradient Boost(91%). Of these 3 the Logistic Regression has the best overall accuracy of 98%. Comparing the LR model to original results with sensitivity of 77% and overall accuracy of 90% we see the improvement we're looking for in specificity, improvement of 16% in LR model and 8% accuracy overall.

3) BERT Pretrain Model

A BERT model was tested using the distill-bert pretrained model with accuracies >98%. This would be a good option if API access is within security constraints. This pre-trained model would require little ongoing maintenance and utilizes a much larger dataset for training therefore the accuracies should be consistent and easily maintained.

```
Epoch:   0%|          | 0/2 [00:00<?, ?it/s]
Train loss: 0.026148697721712938
Epoch:  50%|█████     | 1/2 [1:05:15<1:05:15, 3915.78s/it]
Validation Accuracy: 0.9867788461538461
Train loss: 0.026181237024088717
Epoch: 100%|██████████| 2/2 [5:36:35<00:00, 10097.72s/it]
Validation Accuracy: 0.9867788461538461
```

CONCLUSION

Select Models

Model selection first depends on the business needs. For healthcare this is always to reduce false negatives below .01%. With optimization (class balance, 10 fold cross-validation) and hyperparameter tuning there are several models that work well including KNeighbors and Gradient Boosting however Logistic Regression demonstrated great consistency and would be my selection from this data set. With additional data I would recommend additional testing.

The secondary business needs however could influence model selection. Will the model be allowed to be cloud based? and can the team maintain a BERT model? If so, a BERT model, with the pretraining and terrific overall accuracy, may be a better tool. Many times however healthcare systems cannot access the API connections required due to HIPAA constraints or access to human resources with the proper skills may not be viable.

Future Development

More data:
The holdout test results demonstrate a weakness with the limited data used to train the models. We saw with these results however, with additional data a desired improvement of accuracy can be attained. If additional data becomes available I would include this and rerun training and testing.

Interactive Application:
The tool needs a method to ingest and communicate results. I plan to create an interactive client facing web site to read in a complaint and provide a prediction.