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).

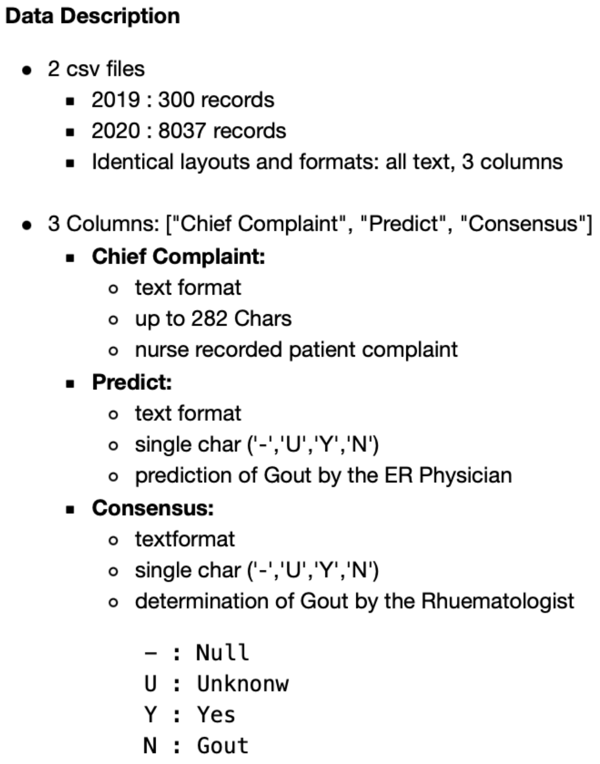
Description

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.

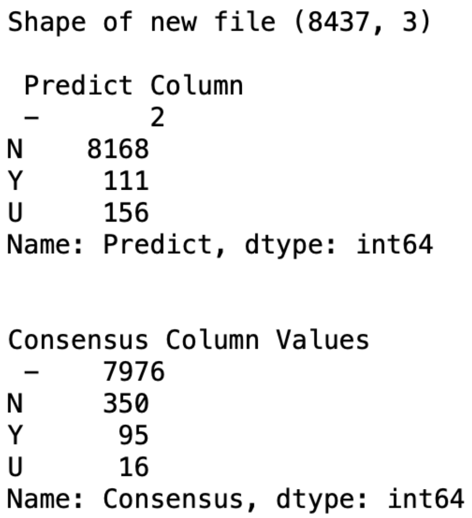


Data Description

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

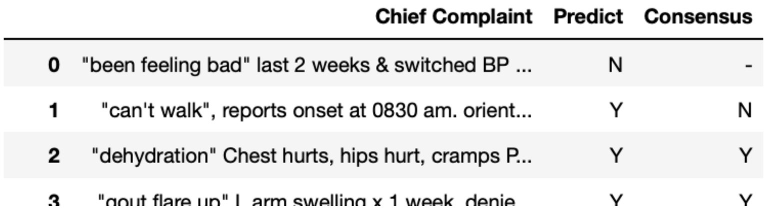
- 3 Columns: ["Chief Complaint", "Predict", "Consensus"]
  - **Chief Complaint:**
    - text format
    - up to 282 Chars
    - nurse recorded patient complaint
  - **Predict:**
    - text format
    - single char ('-','U','Y','N')
    - prediction of Gout by the ER Physician
  - **Consensus:**
    -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

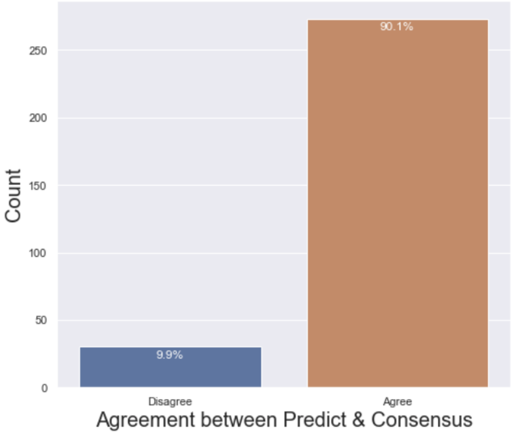
The shape of the resulting dataframe and values in the proposed targets are to the left. The consensus column is preferred as it contains results from a Rheumatologist, clear confirmation of Gout. This column contains a total of 303 records with values Y/N.



	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.

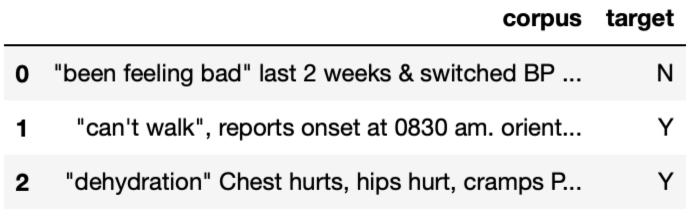
Comparing the Consensus and Predict values, 303 rows existed where both contained values of Y/N. We see about 10% disagreement which makes sense, if it were over 50% I would be concerned.

- 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. In the results we can expect to have more false positives. For this project the 10% was acceptable. 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:

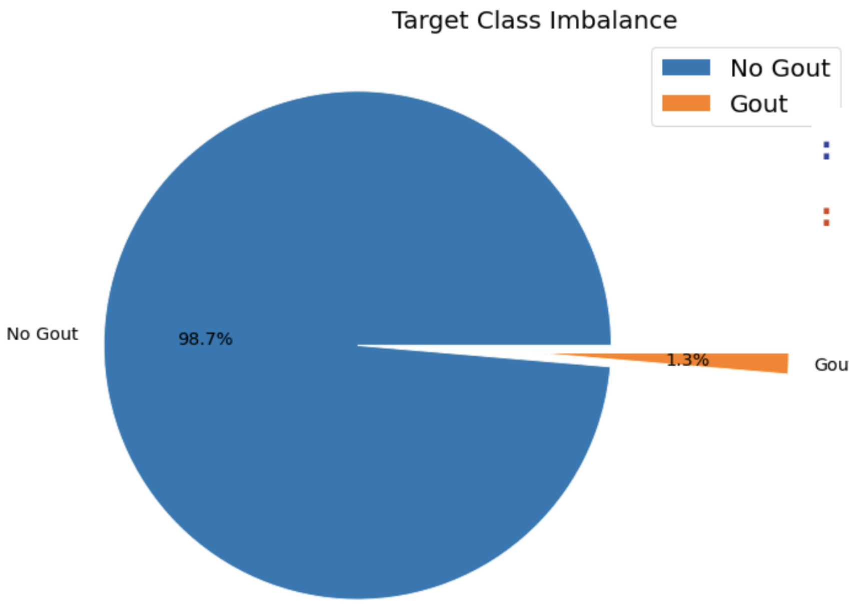
- ***Clean Text:*** remove brackets [ ], punctuation, embedded digits, quotes, and newlines.
- ***Remove Key Term:*** The word 'gout' was removed from the corpus to prevent cheating.
- ***Tokenize:*** Each patient corpus was tokenized by sentence.
- ***Stopwords:*** Stopwords were removed using nltk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

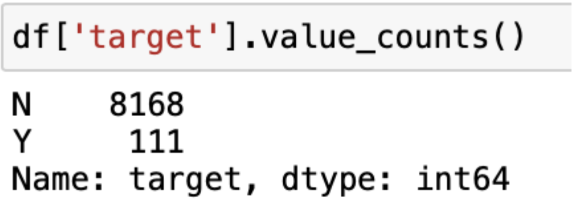
ANALYSIS

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.



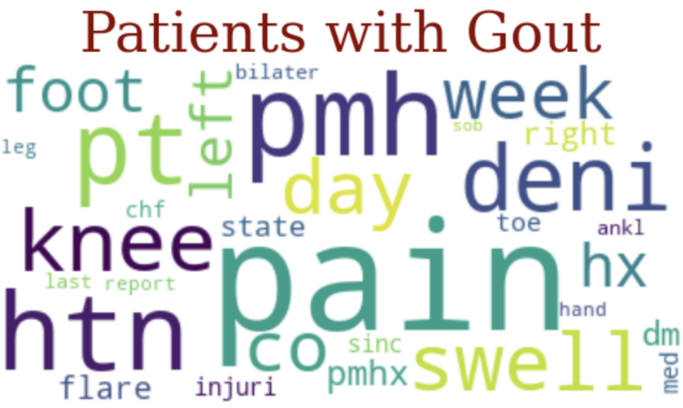


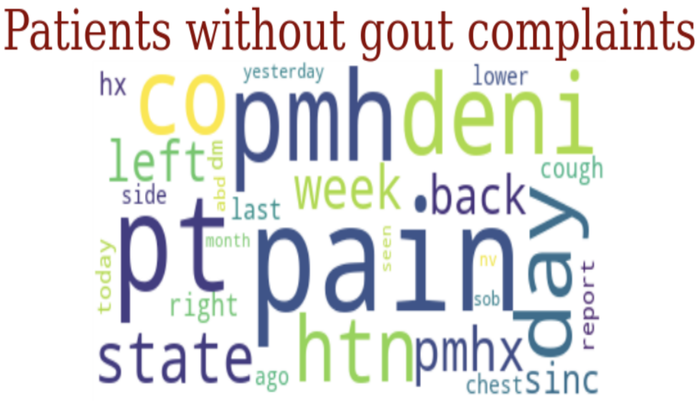
```
df['target'].value_counts()

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.

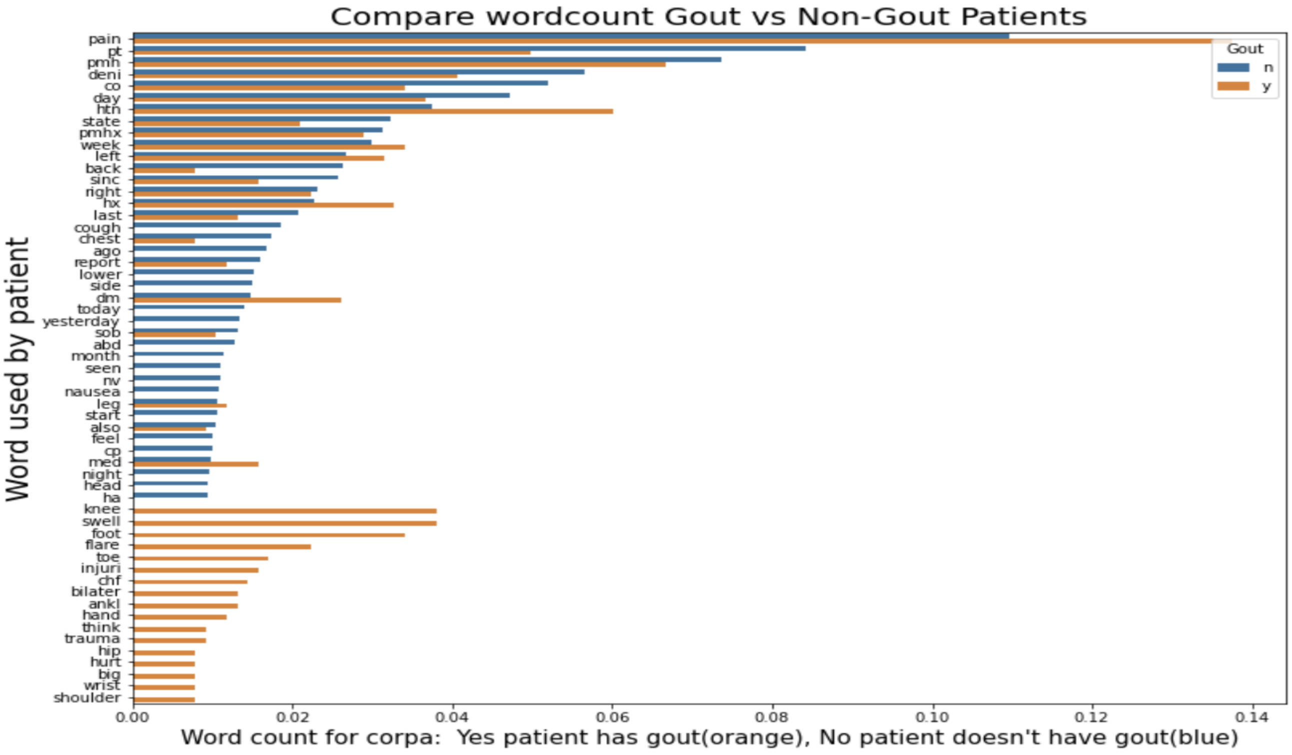
Feature Correlation

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.

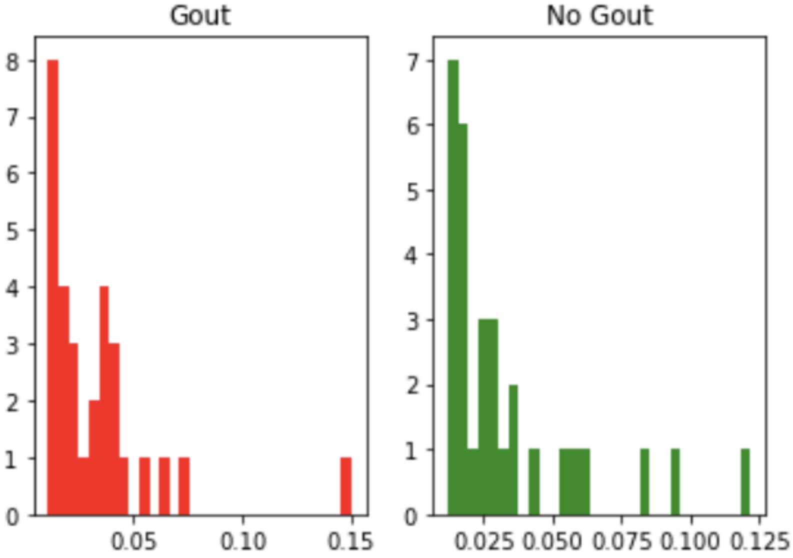




Looking at the top 40 words (below) for Gout (y) and non-Gout (n) patients we can see a nice separation between the classes. Support for good separation between classes, providing confidence that a model can be trained effectively on this data.

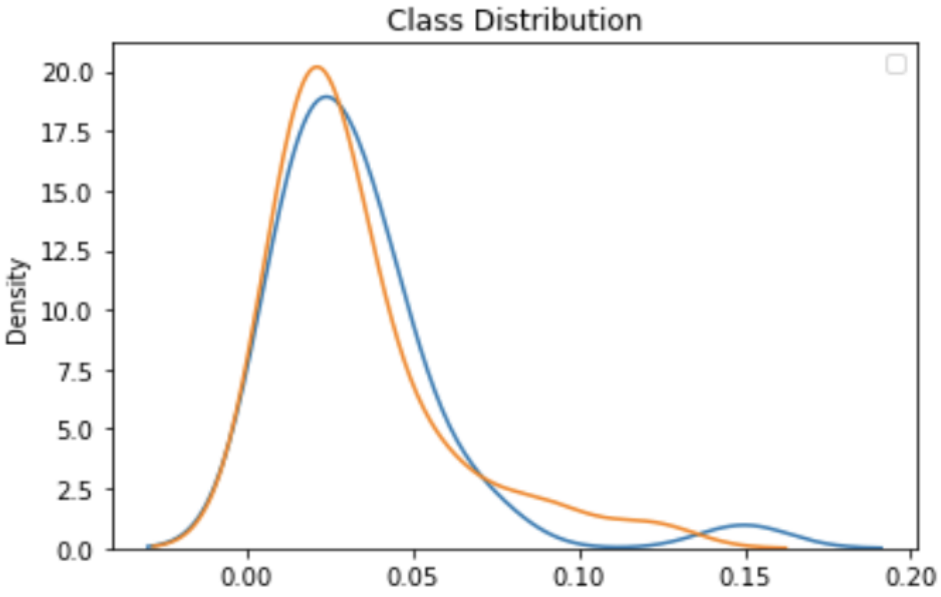


Target Correlation



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

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, while oversampling was performed first, the minor class was such a small sample, oversampling would require repeating each corpus x9, clearly skewing the linear slope causing potential inaccuracies.

Accuracy Measurement

Earlier the data showed the inaccuracies in the ER production of Gout was demonstrated in mostly False Positives so we will look to reduce this using the models that follow. One measurement used to compare True Positives vs False Positives is ROC AUC and a Confusion Matrix based on an F1 score. A reminder, ER physicians had an accuracy of 90%. The usefulness of the model would be to flag the potential misses. *What we really want to see is how the model handles the patient's chief complaint that the doctor missed.* How to capture and compare….

Methods

NLP requires some type of vectorization of the tokens, tf-idf was used in this project in order to make use of it's 'weighting' of keywords. As we saw in the analysis, there were keywords that distinguished between the two classes,, by adding weight to more important words, it should improve the model. The tf-idf vectors were created after splitting the data into test and train data sets using train-test split of 30% from

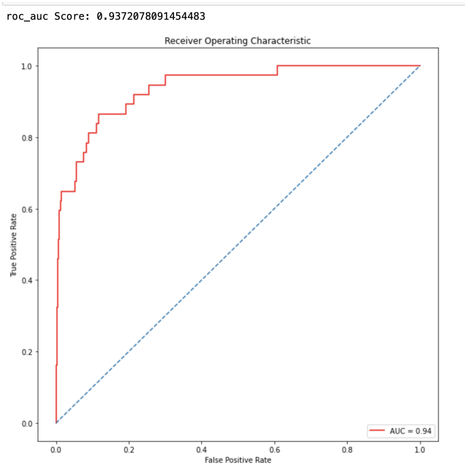
Class imbalance was addressed with oversampling using the RandomOverSampler method from imblearn libraries. The results without tuning were >90% ROC_AUC which is an improvement on the 10% miss rate of the predictions made my ER physicians. I was concerned the high amount of oversampling may be skewing the results, oversampling simply copies the minority samles at random to fill the vacancies. For this project that requires repeating values on a average of 9 times to match the majority. I performed undersampling to compare by slicing 111 records from the majority class to match the minority (downsample) and re-run the models.

Results Summary

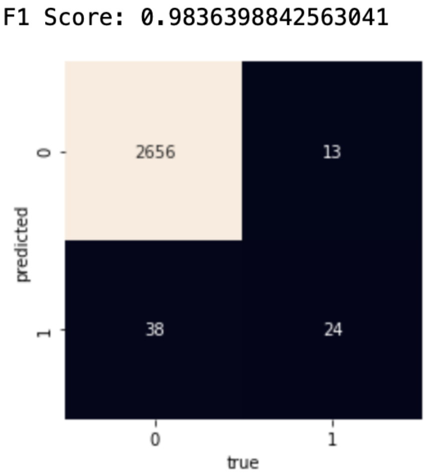
The three models selected were trained and tested using oversampling, the Logistic Regression gave the best overall performance and was then tuned and re-ran. Undersampling was performed and the models ran again for comparison. The accuracies can be reviewed as follows:

Results Summary:

Logistic Regression ROC AUC 93.7% F1 98.3% with 38 False Positive, 13 False Negatives

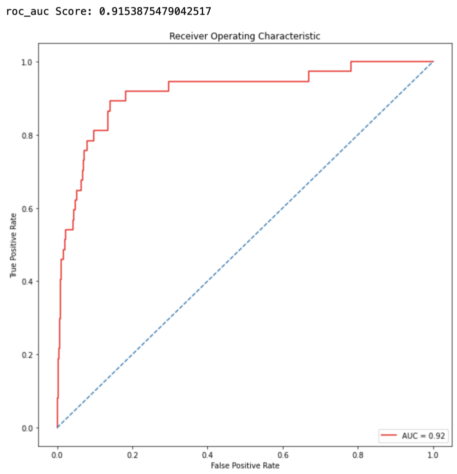


```
roc_auc Score: 0.9372078091454483
```

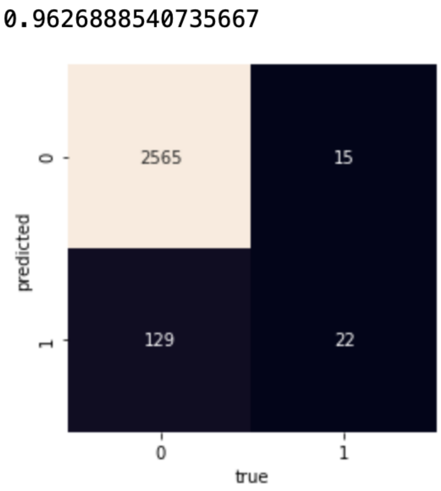


```
F1 Score: 0.9836398842563041
```

Naive Bayes ROC AUC 91.5% F1 98.3% with 129 False Positive, 15 False Negatives

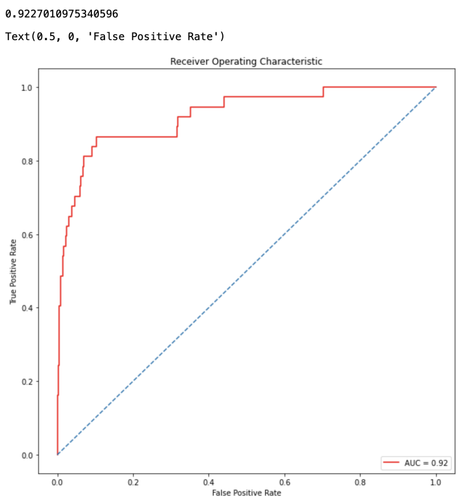


```
roc_auc Score: 0.9153875479042517
```

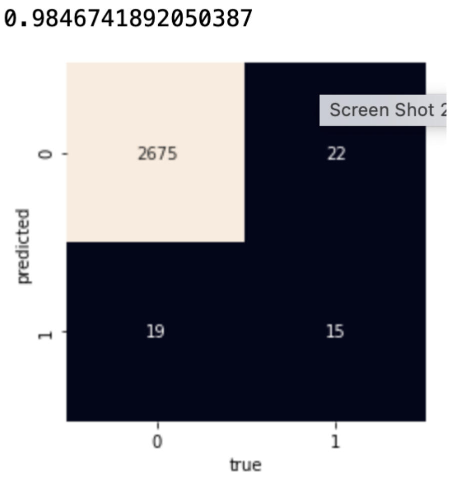


```
0.9626888540735667
```

SVM ROC AUC 92.3% F1 98.5% with 19 False Positive, 22 False Negatives



```
0.9227010975340596
Text(0.5, 0, 'False Positive Rate')
```

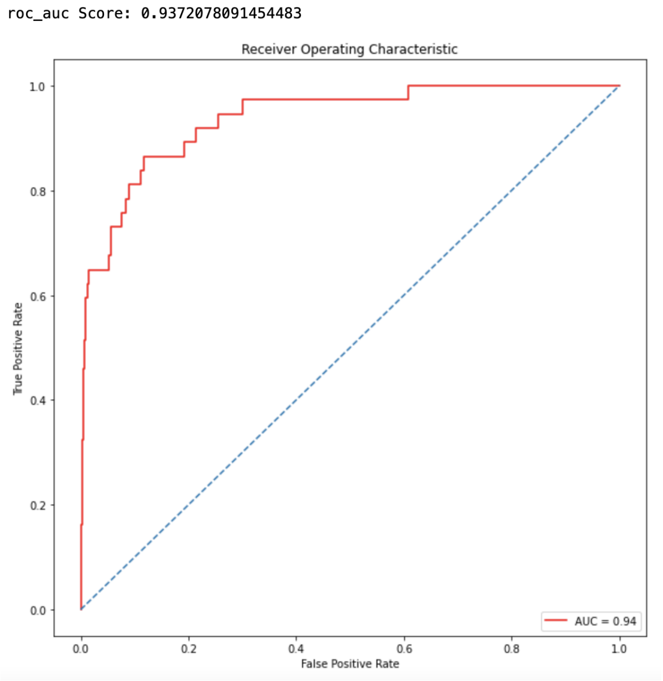


```
0.9846741892050387
```

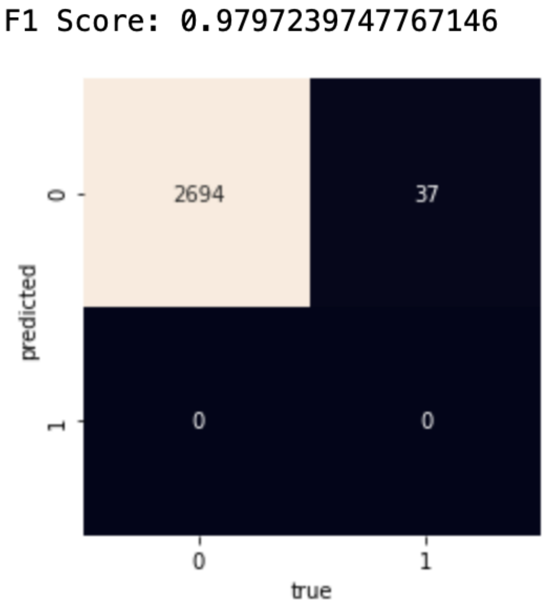
Hyperparameter Tuning

Tuned LR ROC AUC 92.7% F1 98.5% with 19 False Positive, 22 False Negatives

Further testing of the Logistic Regression model with additional hyper-parameter tuning addressed the problem with false positives. Results below show 0 false positives! With an accuracy still beating the 90% average, 93.7% ROC_ACU an improvement of the base model of .4%.



```
roc_auc Score: 0.9372078091454483
```



```
F1 Score: 0.9797239747767146
```

CONCLUSION

Model Selected

The Logistic Regression model was selected because it met the goal of reducing false positives to 0 and increased the overall accuracy to >90%, exceeding the average of the ER physicians in the data provided.

Performance

The models demonstrate accuracies >90% which is an improvement to the ER predictions. With additional data and tuning this tool would certainly improve even further.

Future Development

- Overlapping ROC AUC of all results.
- Finish code on running under sampling to compare to oversampling
- Finish exploring BERT Transformers as a prediction method
- Finish exploring BERT Pre-Training models