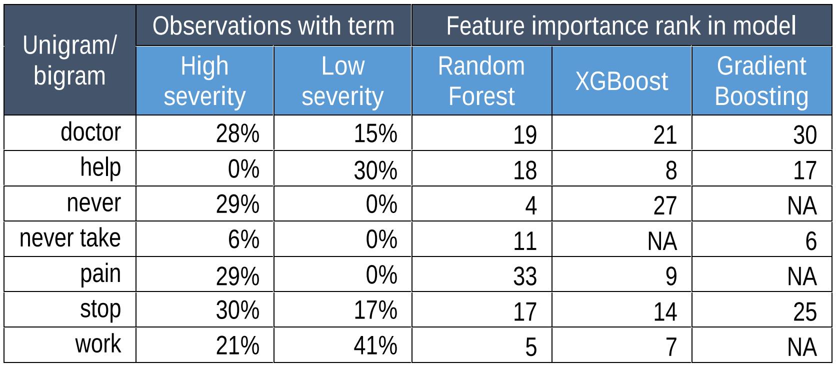
Predicting severity of adverse events

I apply machine learning and natural language processing to historical data to engineer a prediction model that correctly identifies 83% of users having high severity adverse events. This model determines a predicted probability of a user having a high severity event by utilizing text explanations and user demographics.[1] By launching this model into production, we can risk-stratify users, encourage those with a high likelihood of severe adverse reactions to seek medical attention, and ultimately, save lives. This memo (1) discusses the features engineered on unstructured text and structured demographic/drug data, (2) evaluates and describes the machine learning models tested, and (3) discusses implementation and application.

Feature engineering:

I rely upon historical unstructured text where users self-describe adverse events. These data include 11,903 observations and contain whether the event was high severity (-1) or low severity (+1).[2] 53% of these events are high severity. I create a bag of words for the 500 top unigrams/bigrams.[3,4] The table below contains a sample of unigrams/bigrams tokens that are important in differentiating between high and low severity events. For example, users mentioning “work” are more likely have low severity events and “pain” mentions are more likely to have high severity events.



Unigram/ bigram	Observations with term		Feature importance rank in model		
	High severity	Low severity	Random Forest	XGBoost	Gradient Boosting
doctor	28%	15%	19	21	30
help	0%	30%	18	8	17
never	29%	0%	4	27	NA
never take	6%	0%	11	NA	6
pain	29%	0%	33	9	NA
stop	30%	17%	17	14	25
work	21%	41%	5	7	NA

I also utilize sentiment analysis to separate high from low severity events. For each text entry, I calculate a happiness score, which provides a numerical rating based on the happiness of words in the text. I rely on Harvard sentiment dictionaries to calculate a count of negative, positive, feel, and pain words used in each event report. As shown in the Appendix, the happiness, negative, positive, and pain scores are among the top 10 most important features in the tree-based models. The Appendix also highlights the average score for these factors between high and low severity events.

I also utilize structured data. For each event, we know the gender, age, drug, and duration of time the user has been taking the drug. I create a dummy variable for all drugs with at least 100 training observations. I find that the duration of time a user has been taking the drug is the most important feature in predicting severity. On average, low severity events have a drug duration of 990 days compared to 324 days for high severity events. In addition, I find some drugs—such as Micronor and Depo-Provera—are important in explaining adverse events (both of these drugs are associated with high severity events).

Model construction and evaluation:

I separate the historical data into a 20% test, 20% validation, and 60% training sets.[5] I run the following models: ada boost, extra trees, random forest, gradient boosting, XGBoost (i.e., “extreme” gradient boosting), SVM, linear discriminant analysis (LDA), and feed forward neural network. I select optimal hyperparameters for these models using

[1] The outcome variable is the probability of being low severity. 1-prob low severity = prob high severity.

[2] For purposes of modeling, I code -1s (high severity) as 0s. This allows the models to predict the probability of low versus high severity.

[3] 500 was used as a feature cutoff point, because (1) additional words uses more computing power and (2) in DrugExperiencePart1 we found there was not a loss in predictive power between the top 25% and 50% of features used in a bag of words. We rely on unigrams and bigrams, because bigrams tended to have the best model performance in DrugExperiencePart1 (other than the random forest, where unigrams did best).

[4] I lemmatize the data and remove stop words before creating a bag of words and performing sentiment analysis.

[5] I use 5-fold cross validation to fit our training data (iterates 5 times with a 20% holdout set each iteration).

grid searching. Grid searching allows us to loop through and select the combination of hyperparameters that maximize model accuracy.

I selected gradient boosting, XGBoost, LDA and the neural network to create an ensemble model via soft voting classifiers.[6] I select these models because they did not appear to be overfit[7], had superior accuracy, precision, and recall metrics, and had nice diversity. The table below summarizes model performance in the test set for the individual and ensemble models after grid searching selected the optimal hyperparameters. Confusion matrices are available in the Appendix.[8]

Model	Accuracy	Recall (high severity)	Recall (low severity)	Precision (high severity)	Precision (low severity)
Ensemble	83.1%	82.8%	83.5%	84.9%	81.2%
Gradient boosting	82.1%	82.4%	81.8%	83.6%	80.6%
XGBoost	81.6%	83.0%	79.9%	82.3%	80.7%
Neural network	81.2%	85.1%	78.0%	81.3%	82.3%
LDA	81.1%	81.6%	80.6%	82.5%	79.6%
Random forest	78.2%	84.8%	70.7%	76.4%	80.6%
Extra trees	76.5%	85.6%	66.3%	74.0%	60.4%
Ada boost	69.2%	72.0%	66.1%	70.5%	67.7%
SVM	58.0%	21.8%	98.8%	95.1%	52.9%

As shown above, the ensemble has the highest accuracy. Compared to the ensemble, the neural network has a 2.3 percentage point improvement in identifying actual high severity events, from 82.8% to 85.1%. However, the neural network has 3.6 additional percentage points of high severity false positives (decrease from 84.9% to 81.3%). We need to run a cost-benefit analysis to evaluate the trade-off between false negatives (actual high categorized as low severity) and false positives (actual low categorized as high severity) to determine whether the ensemble or neural network is preferable.

ROC curves allows us to compare how models separate high from low severity at various thresholds. The confusion matrices above assume a threshold of 50% (below 50% is high severity), but ROC curves allow us to observe how models separate high from low severity at all thresholds. The closer a curve is to the top left of the graph, the better it distinguishes high from low severity events. As shown by the ROC curves in the Appendix, the ensemble performs the best. The four models in the ensemble are also close together, which shows there is not much variability underlying the models, and therefore, the models are robust and reliable.

Implementation and application:

After a careful analysis of the trade-off between false positives and false negatives for the ensemble and neural network, we can implement a predictive model into production. This model will work “live” in the app. As a user self-reports an adverse event, the app will provide the user with severity feedback. When the model determines a user as likely having a severe adverse reaction, the app will encourage them to seek medical attention. We can also collect information from users on outcomes. For example, we can collect information on whether/when the user saw a healthcare provider and what the diagnosis and severity was. This will give us the ability to (1) determine the impact of our model on health outcomes and (2) update our historical training data such that we can improve model performance.

On a monthly basis, we can collect additional data and update the models. By collecting additional data, we can more precisely train our models and identify additional key text and drug information in order to more accurately predict severity. We are confident that these severity prediction models will improve health outcomes and save lives.

[6] Soft voting takes a weighted average probability of the models in the ensemble to maximize accuracy. The weighted average approach allows better performing models in the ensemble to have a greater weight in the ensemble prediction, which makes this method preferred to a simple average.

[7] See learning curves in DL Python notebook (these graphs allow us to observe the models that are over/underfit). For example, the extra trees appears overfit (lines close and diverging) and the ada boost appears underfit (lines far and not converging).

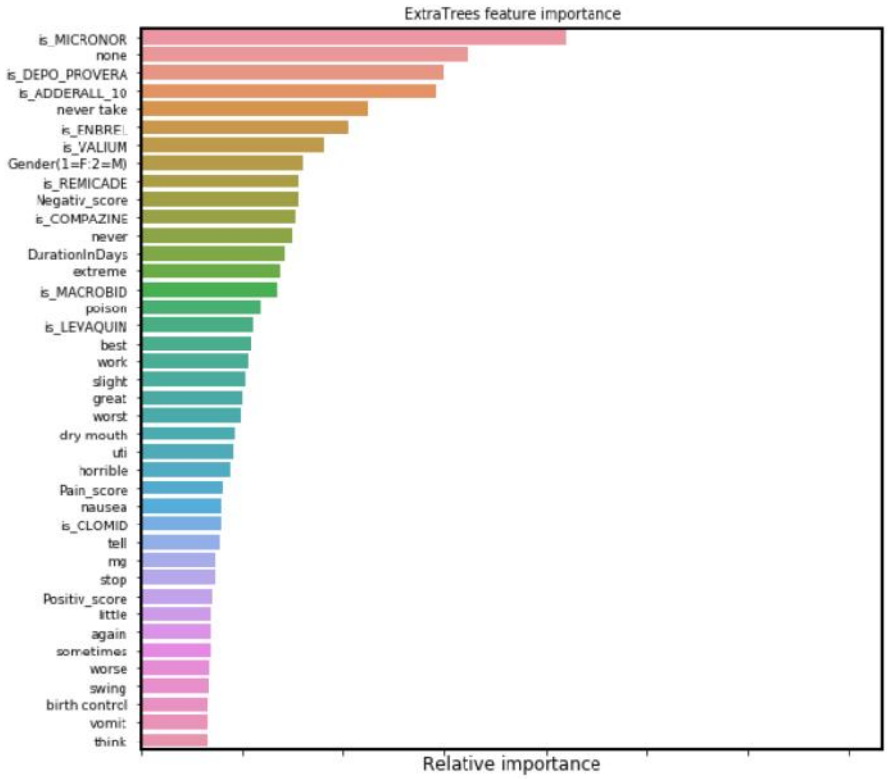
[8] Accuracy is the share of observations in the test set that are correctly classified. Recall (high severity) is the percentage of actual high severity events that the model correctly classifies as high severity (and same for Recall (low severity)). Precision (high severity) is the percentage of model classified high severity events that are actually high severity (and same for Precision (low severity)).

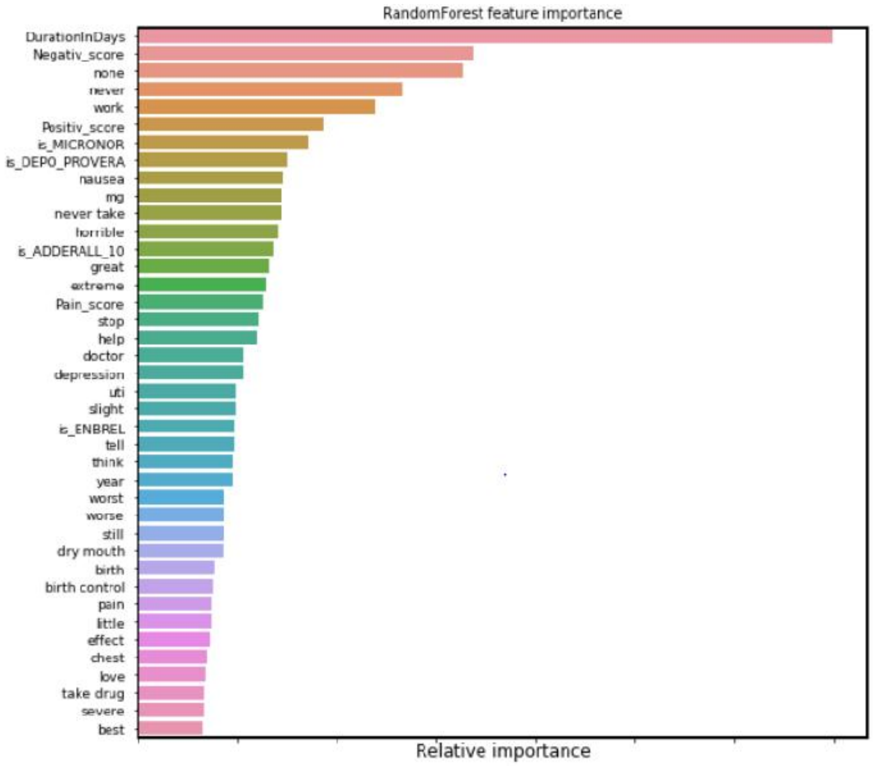
Appendix

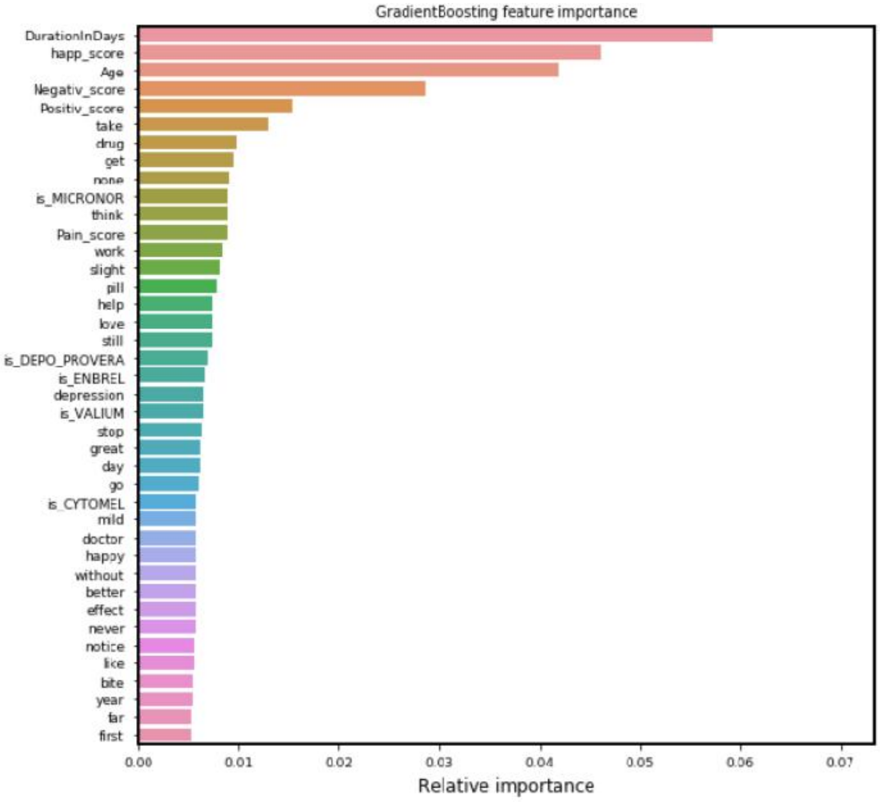
Demographic and sentiment features:

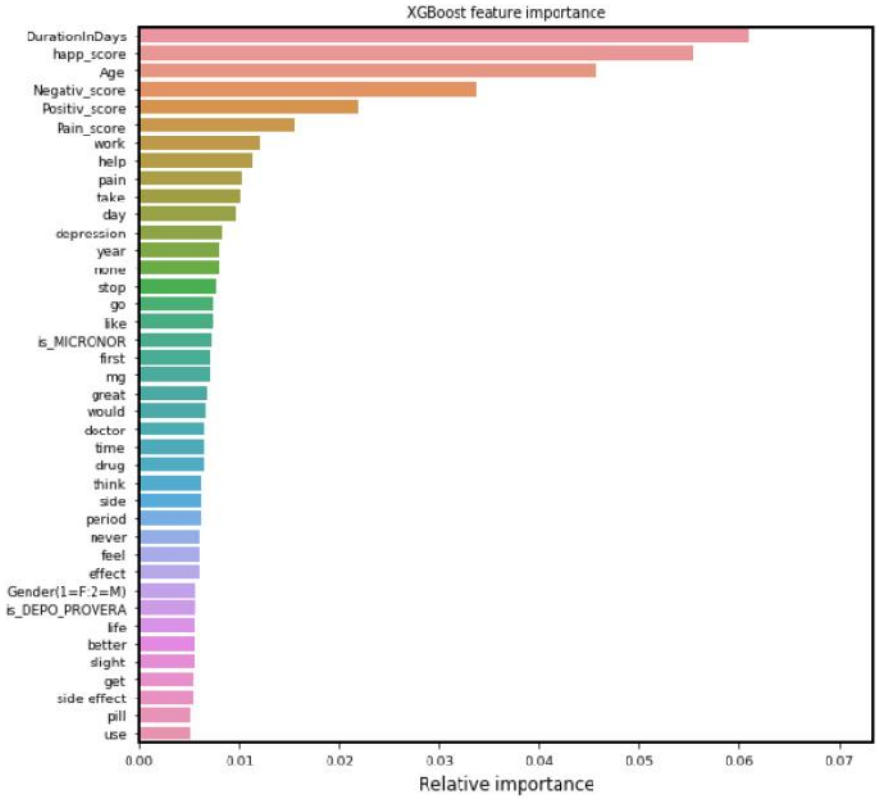
Feature	Average score/demographic	
	High severity	Low severity
is_DEPO_PROVERA	0.18	0.08
is_MICRONOR	0.05	-
Gender(1=F:2=M)	1.19	1.28
Age	37.56	37.93
DurationInDays	323.92	990.27
happ_score	234.99	216.85
Feel_score	0.03	0.03
Negativ_score	4.94	3.47
Positiv_score	0.82	1.19
Pain_score	1.59	1.14

Feature importance in models:

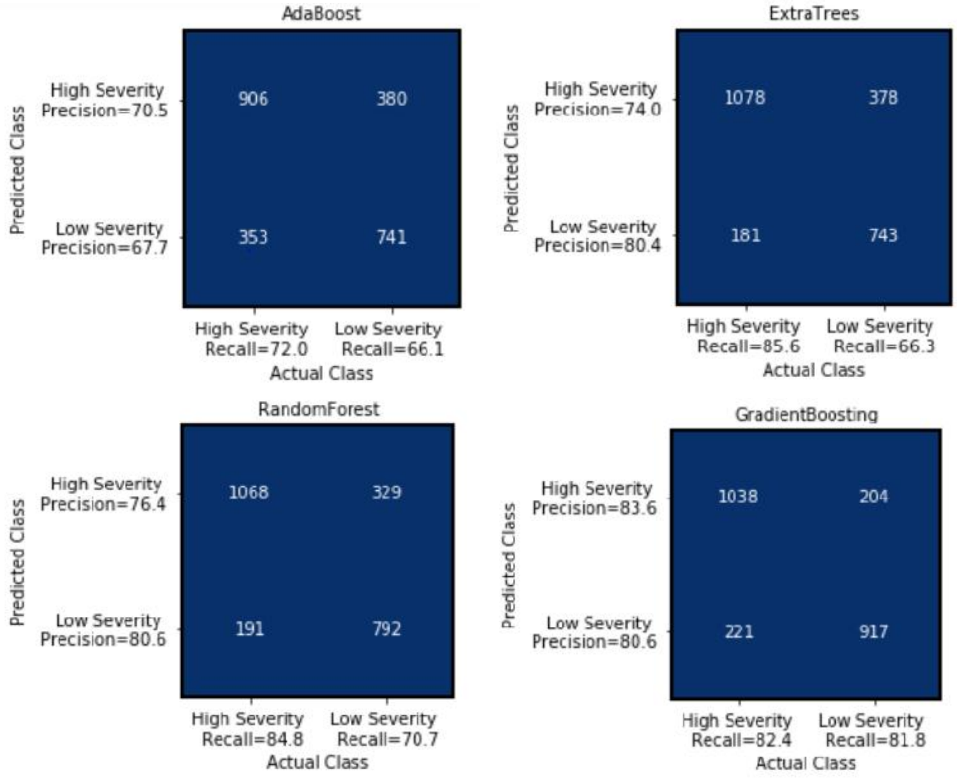


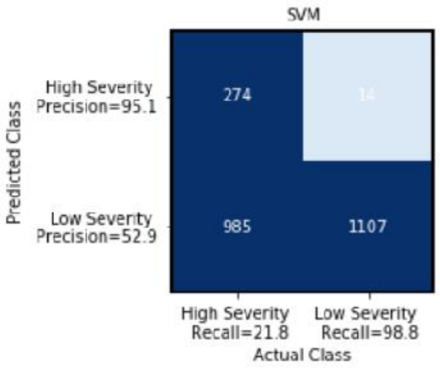

RandomForest feature importance
DurationInDays
Negativ_score
none
never
work
Positiv_score
is_MICRONOR
is_DEPO_PROVERA
nausea
mg
never take
horrible
is_ADDERALL_10
great
extreme
Pain_score
stop
help
doctor
depression
uti
slight
is_ENBREL
tell
think
year
worst
worse
still
dry mouth
birth
birth control
pain
little
effect
chest
love
take drug
severe
best
Relative importance

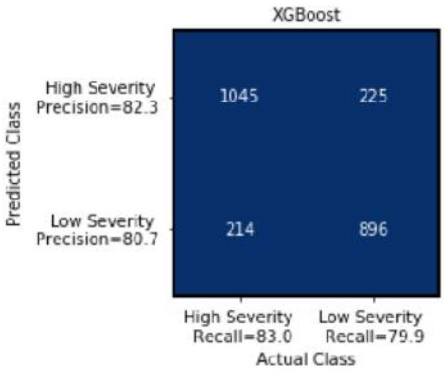

GradientBoosting feature importance
DurationInDays
happ_score
Age
Negativ_score
Positiv_score
take
drug
get
none
is_MICRONOR
think
Pain_score
work
slight
pill
help
love
still
is_DEPO_PROVERA
is_ENBREL
depression
is_VALIUM
stop
great
day
go
is_CYTOMEL
mild
doctor
happy
without
better
effect
never
notice
like
bite
year
far
first
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Relative importance

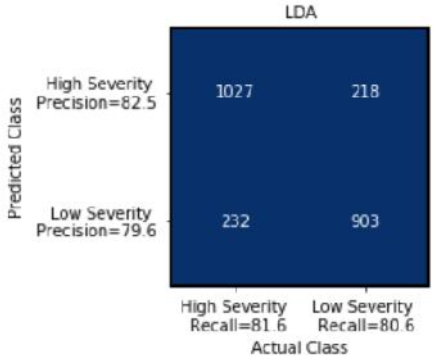


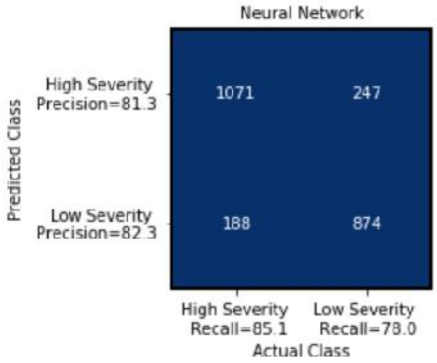
Confusion matrices for models:

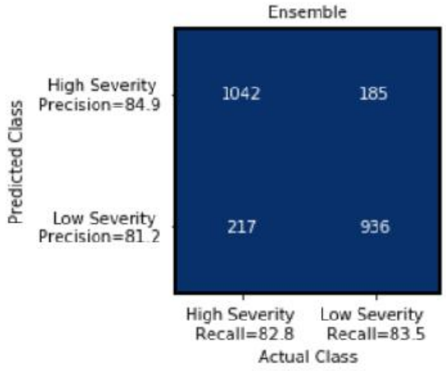



SVM
Predicted Class
High Severity
Precision=95.1
Low Severity
Precision=52.9
274
14
985
1107
High Severity
Recall=21.8
Low Severity
Recall=98.8
Actual Class

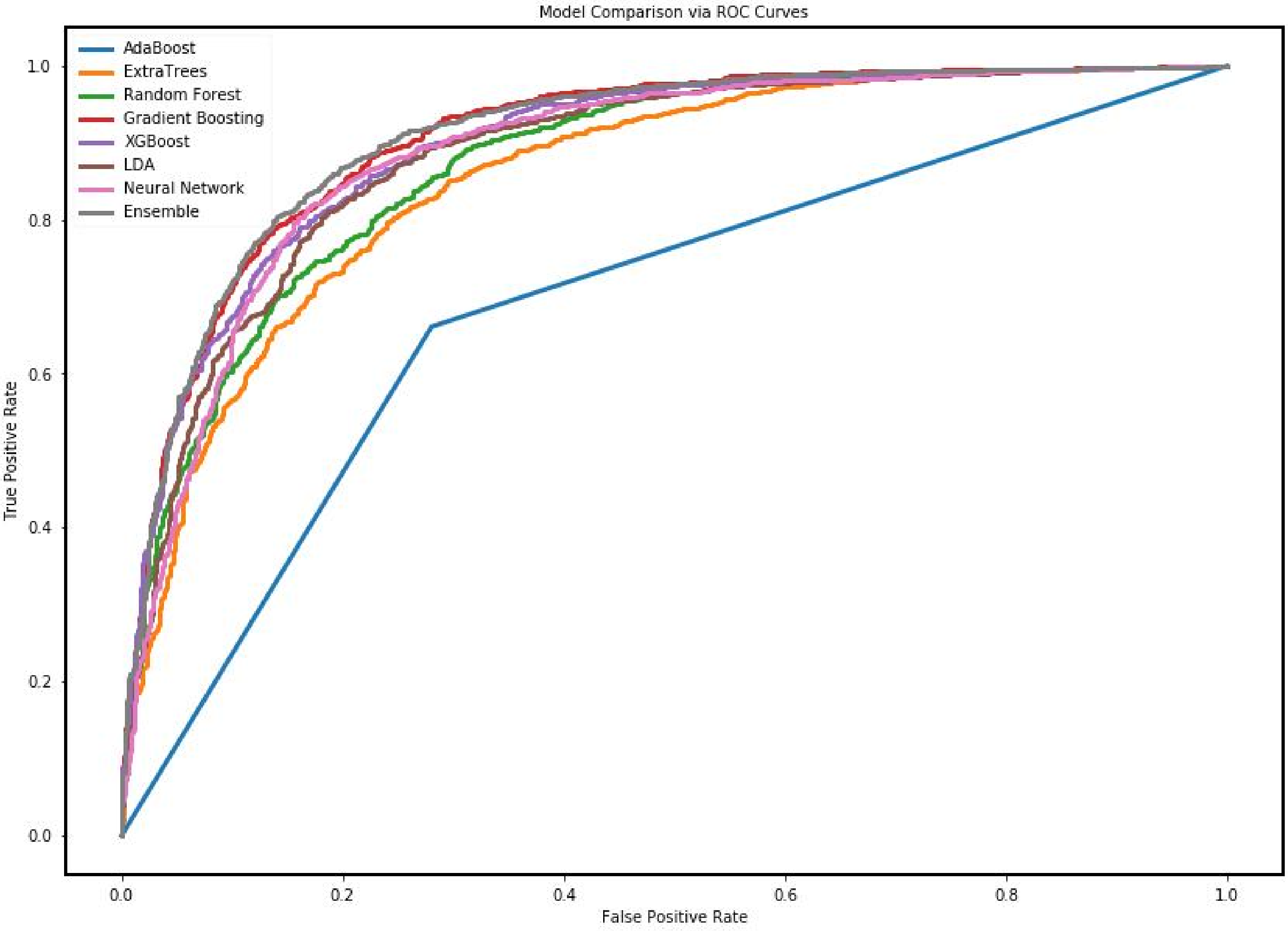

XGBoost
Predicted Class
High Severity
Precision=82.3
Low Severity
Precision=80.7
1045
225
214
896
High Severity
Recall=83.0
Low Severity
Recall=79.9
Actual Class


LDA
Predicted Class
High Severity
Precision=82.5
Low Severity
Precision=79.6
1027
218
232
903
High Severity
Recall=81.6
Low Severity
Recall=80.6
Actual Class

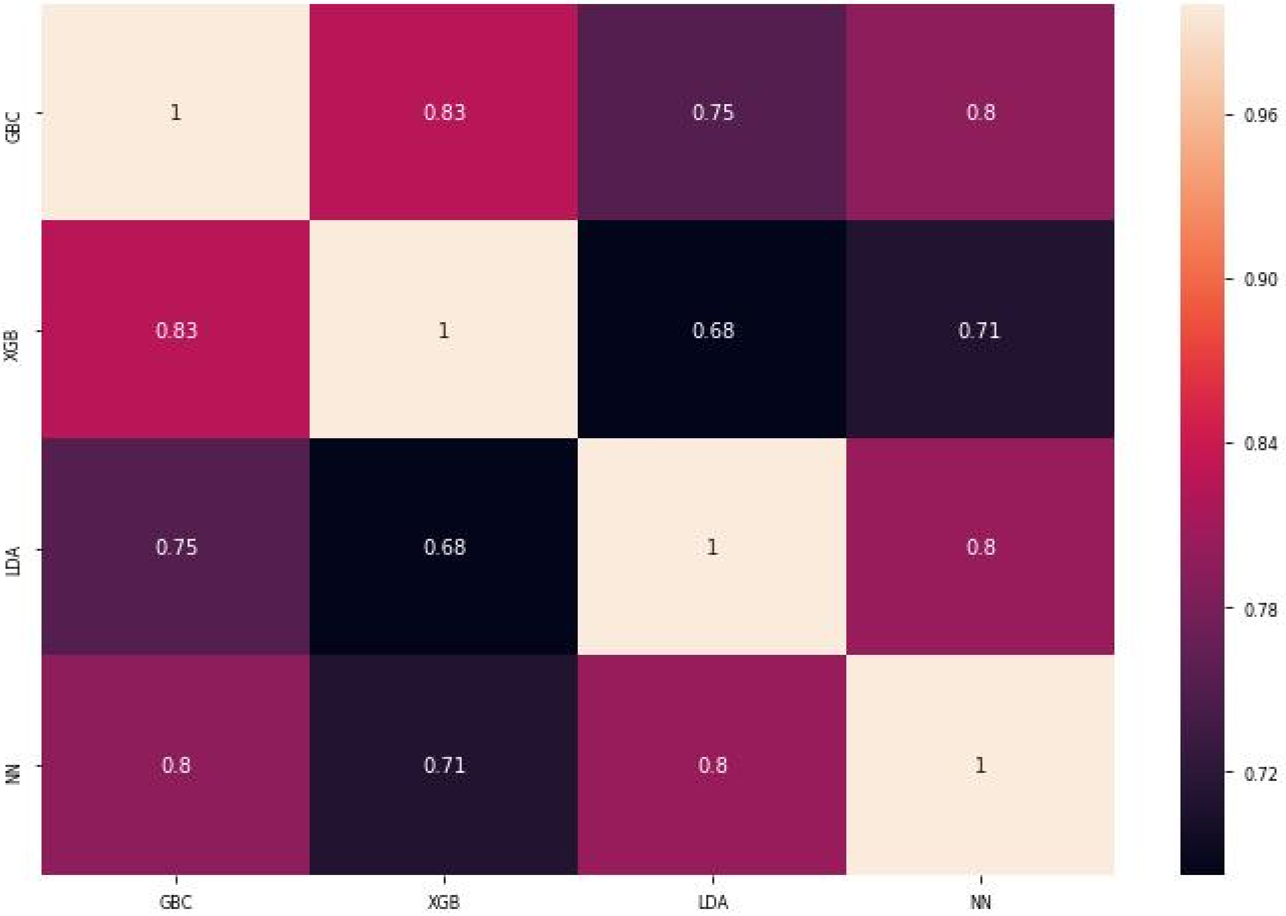

Neural Network
Predicted Class
High Severity
Precision=81.3
Low Severity
Precision=82.3
1071
247
188
874
High Severity
Recall=85.1
Low Severity
Recall=78.0
Actual Class


Ensemble
Predicted Class
High Severity
Precision=84.9
Low Severity
Precision=81.2
1042
185
217
936
High Severity
Recall=82.8
Low Severity
Recall=83.5
Actual Class

ROC curve:


Model Comparison via ROC Curves
AdaBoost
ExtraTrees
Random Forest
Gradient Boosting
XGBoost
LDA
Neural Network
Ensemble
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

Correlation matrix for ensemble:



Note: From matrix above, we can see that the ensemble of these four models relies upon variation among model predictions (the correlations above are not high). As a result, we are more likely to bracket the actual severity (i.e., the actual severity is between multiple predictions in the ensemble, and an average causes the prediction and the actual to converge).