



The U.S. Opioid Epidemic

According to the 2015 National Survey on Drug Use and Health, 12.5 million people misused prescription opioids. Furthermore:

- 2.1 million people misused prescription opioids for the first time
- 2 million had prescription opioid misuse disorder
- 33,091 died from overdosing on prescription opioids
- 15,281 deaths were attributed to overdosing on commonly prescribed opioids
- 4 out of 5 people who were addicted to heroin (a cheaper and more affordable street drug) started out using prescription pain pills

Once addicted, it is a painfully gruesome process to overcome the symptoms of opioid withdrawal and ultimately conquer the addiction. Early stages of withdrawal are much like the flu. Victims become nauseated, lose their appetite and ability to sleep, and experience aches all throughout their bodies. **That's just the first week.** In Travis Rieder's TedTalk, *I was in opioid withdrawal for a month — here's what I learned*, he describes the month-long withdrawal experience as going through hell.

"I thought I would die. I assumed that I would die... I'm sorry. [pausing to fight back tears]
Because if the symptoms didn't kill me outright, I'd kill myself."

To make matters worse, most doctors simply recommend continuing their prescription use until they can "figure something else out." Recommendations like this give victims a sensation of despair, imagining what the experience might be like the next time they attempt to wean themselves off of the drug. Will the symptoms amplify? Will this person be addicted to opioids forever?

Opioid Addiction Treatment: Digging into the Business Issue

When an addict decides to begin rehabilitation, one of the first questions they face is “how am I going to pay for this?” Rehabilitation is expensive and individual requirements vary significantly between patients. According to addictioncenter.com, “for those requiring 60- or 90-day [in-patient rehab] programs, the total average costs could range anywhere from \$12,000 to \$60,000.”

For patients who may not need the extensive care of a rehabilitation center, these costs are not negligible. Treatment options such as methadone cost opioid-users around \$4,700 annually. Identifying and mitigating these risks contains the potential to affect both human lives and profit margins. The burden of these costs is carried by health insurance providers and ultimately passed along to customers.

Additionally, the costs of opioids are often split between the health insurance provider and the patient. Thus, if we can reduce the amount of opioids that are prescribed in the first place, we can reduce the costs that Humana has to pay for its customers' prescriptions. For a complete analysis of the potential cost savings on opioids, see our final analysis and actionable insights towards the end of this document.

The Humana Analytics Competition

Using data from a 4-year longitudinal view of events for an MAPD member, participants are required to analyze the data relating to various interactions with Humana. Events such as Rx Claim - Paid, Fully Paid Medical Claim, Inbound Call By Member, New Diagnosis, etc. are included in the dataset.

The goal is to develop a predictive model that takes in a set of patient histories up to the point of an "opioid-naive event" and assign to each patient a probability of long term opioid therapy being required in the patient's future. These probabilities will be evaluated by the ROC AUC paradigm, essentially measuring the ability of the model to correctly classify patients.

featureHealth Analytics' Analysis

Analytical Tools Used

- Interactive Python in Jupyter Notebooks
- Pandas and Numpy for feature extraction
- Scikit-learn, keras, and lightgbm for model development
- Matplotlib for data visualization
- Excel for data exportation

Data Preperation

As presented in the file HMAHCC_COMP, the data presented several integrity issues that first needed to be addressed. Most prominently, the columns of the provided file mixed several different attributes. To alleviate this issue, we split the data into the following dataframes in Python:

- **New_diagnosis** - all rows that described new diagnoses such as CPD, Hypertension, Top 5, CAD, CHF, and Diabetes
- **Call** - all rows that described calls data
- **Non_rx_claim** - all rows that described claims that were not related to prescription drugs
- **New_provider** - all rows that described a patient seeing a new doctor, physician, or medical practitioner
- **Rx_claim** - all rows that described prescription drug claims
- **Rx_paid** - all rows that described the transactions, so to speak, of prescription drug claims
- **Rx_reject** - all rows that described prescription drugs claims that were later reversed for various reasons

Deriving the Response Variable

We were provided the following definition:

Opioid Naïve will be defined as not having an opioid 'on hand' in the preceding 90-day period, based on service date and payday supply count.

We designed a function (see following page) that first, using the PAY_DAY_SUPPLY_CNT attribute, determined what days each patient had opioids on hand. The function then examined every 180 day period past their first opioid naïve event, denoted as day 0 in the dataset. If the patient had any 180 day period with drugs on-hand 162 days or more (90%), we classified the patient as LTOT.

While this is a slight deviation from the exact definition of LTOT, we intentionally used this overapproximation to aid in further separating the classes and to decrease our model's runtime. Interestingly, we found that implementing this approximation differed only by approximately 30 individuals out of the 14,000 provided patient histories ($\sim 0.2\%$), highlighting that the first six months past a qualifying event serve as a key indicator of opioid use.

Our derived response variable showed that the classes were nearly equal in distribution in the oversampled dataset we were provided, with 50.4% classified as LTOT.

```

def LTOT(self):

    def get_LTOT(ID):
        try:
            group = self.opioid_data_grouped.get_group(ID)
            frame = group[['Days', 'PAY_DAY_SUPPLY_CNT']].drop_duplicates()
            frame = frame[frame['Days'] >= 0]
            frame['drugs until'] = (frame['Days'] + \
                                   frame['PAY_DAY_SUPPLY_CNT']).astype(int)

            frame['range'] = frame.apply(lambda x : \
                                         range(x['Days'].astype(int),
                                               x['drugs until'].astype(int)), 1)

            concat = concatenated = chain(*list(frame['range']))
            concat = set(concat)

            day_frame = pd.DataFrame(columns = ['Days', 'Has Drug?'])
            day_frame['Days'] = range(max(concat)+1)
            day_frame['Has Drug?'] = (day_frame['Days'].isin(concat))

            for n in range(180, len(day_frame)+1):
                frame_slice = day_frame.iloc[n-180:n]
                drug_days = np.sum(frame_slice['Has Drug?'])

                if drug_days >= 162:
                    return (True, n-180, n)

            return (False, np.nan, np.nan)
        except:
            return (np.nan, np.nan, np.nan)

    id_list = self.raw_df['id'].drop_duplicates().values

    response_variable = pd.DataFrame(id_list, columns = ['id'])
    response_variable['LTOT'] = response_variable['id'].map(get_LTOT)

    response_variable[['LTOT', 'Beginning Date', 'End Date']] = \
    pd.DataFrame(response_variable['LTOT'].tolist(), index = response_variable.index)

    self.response_variable = response_variable.set_index('id')

```

Exploratory Analysis

Assumptions

In the preliminary research stage, we first hypothesized that it would be most effective to engineer features that would identify drug-seeking behavior. However, as we researched the subject further we found that the data did not support this assumption.

We analyzed:

- how many times a member switched providers
- how many times a member called Humana and why
- how many diagnoses a member received at once or altogether
- how many times a member was denied opioids at each level of strength

None of these features seemed to correspond with a member becoming LTOT. We then tried to understand the mechanism of addiction to opioids through outside research. We watched and read a myriad of YouTube videos and online articles to uncover what new potential features we could use in our analysis. Based on this research, we formed a new set of hypotheses:

Opioid addiction is primarily connected with

- the length of prescriptions of opioids
- the strength of the medication

Hypotheses to inspect

(a) Length of prescriptions: the longer a patient consumes opioids, the more likely they are to become LTOT.

Here we looked at the PAY_DAY_SUPPLY_CNT on day 0, where PAY_DAY_SUPPLY_CNT represents the number of days a patient takes opioid drugs. As shown in the table below, around 93% of patients with a high SUPPLY_CNT_level (over thirty days worth of opioids on-hand) are classified as LTOT.

Vice versa, only 18% in low SUPPLY_CNT_level group are classified as LTOT. We therefore concluded that the initial length of prescriptions is an important indicator of LTOT risk.

	Not LTOT	of LTOT
SUPPLY_CNT_level		
low	0.813788	0.186212
medium_low	0.681524	0.318476
medium_high	0.225567	0.774433
high	0.072139	0.927861

(b) Strength of the medication: the stronger drug used, the more likely they are to become LTOT.

Take HYDROCODONE-ACETAMINOPHEN as an example, the most commonly used opioid drug in this dataset. Within this opioid, three varieties of drug strength are available, ranging from 5-325 MG to 10-325 MG.

As shown in the table below, 70% of patients who were prescribed the strongest type on day 0 became classified as LTOTs, while only 33% of patients who took the lower strength equivalent became classified as LTOT. We concluded that the generic name (including type) of the prescribed medicine could be a significant predictor of LTOT classification.

	Not LTOT	of LTOT
HYDROCODONE-ACETAMINOPHEN TAB 5-325 MG	0.670743	0.329257
HYDROCODONE-ACETAMINOPHEN TAB 7.5-325 MG	0.537375	0.462625
HYDROCODONE-ACETAMINOPHEN TAB 10-325 MG	0.297368	0.702632

Feature Engineering

Based on the assumptions outlined above, we sought to create a set of features that captured information that indicated the strength and duration of prescribed opioids. We derived the following:

- **SUPPLY_CNT_on_day0**: opioid supply days at the time of the qualifying event
- **total_SUPPLY_CNT_prior**: total opioid supply days prior to the qualifying event
- **Supply_times**: total number of instances of opioid perscriptions
- **PAYABLE_QTY_on_day0**: amount of opioids supplied at the time of the qualifying event (amount of physical pills)
- **total_PAYABLE_QTY_prior**: amount of opioids supplied prior to the qualifying event (amount of physical pills)
- **MME_on_day0**: Morphine Milligram Equivalence (MME)
- **max_MME_prior**: maximum MME prior to the qualifying event
- **avg_MME_prior**: average MME prior to the qualifying event
- **MME_times_SUPPLY_day_0**: MME multiplied by opioid supply days at the time of the qualifying event

Principal Componets of Generic Attribute - to incorporate the information from the generic name of the opioids perscribed to each patient, we utilized the dimensionality reduction technique of principal components. First for each patient we compiled the PAY_DAY_SUPPLY_CNT of each of the 103 generic names perscribed to each patient on the day of their qualifying event.

Applying the principal components alogrithim, we reduced these 103 columns to a set of ten columns for each patient, which was calulated to capture 92.8% of variance within the dataset.

Cost terms:

These terms describe certain attributes at the time of the qualifying event:

- **total_costs_on_day_0**
- **total_net_payment_on_day_0**
- **net_payment_portion_on_day_0**
- **opioid_cost_portion_on_day_0**

Interaction terms:

These terms certain attributes at the time of the qualifying event, normalized by the supply count at the time of the qualifying event:

- **total_cost_divide_SUPPLY_day_0**
- **total_net_payment_divide_on_day_0**
- **np_portion_divide_SUPPLY_day_0**
- **oc_portion_divide_SUPPLY_day_0**
- **max_MME_prior_divide_SUPPLY_day_0**
- **avg_MME_prior_divide_SUPPLY_day_0**
- **tsc_prior_divide_SUPPLY_day_0**
- **tpa_prior_divide_SUPPLY_day_0**
- **oc_day_0_divide_SUPPLY_day_0**
- **np_day_0_divide_SUPPLY_day_0**

Interaction Term Interpretation

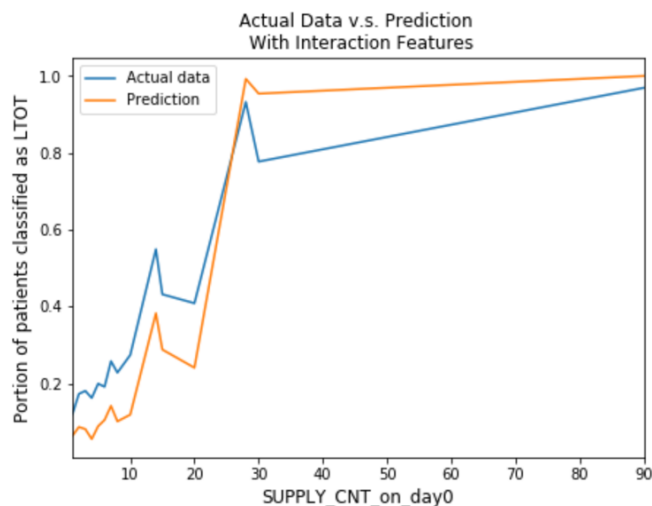
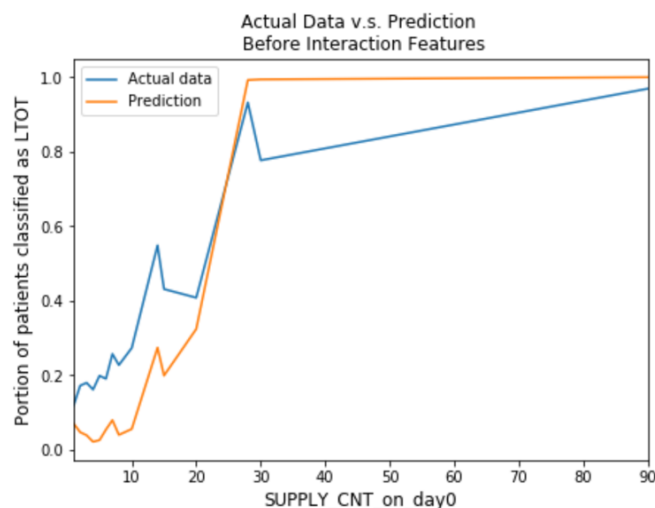
Features divided by SUPPLY_CNT_on_day0

By visualizing the difference between our validation model predictions and actual data (as shown below), we found that there was a strong connection between the opioid supply count on the day of the qualifying event, albeit with systematic residuals.

Through approximately an initial supply count of 0-20, the variations in LTOT classification are nearly a perfect mirror of the variations of our model predictions, but systematically underpredict the proportion of LTOT. Past this point, at approximately an initial supply count of 30, our model uniformly overpredicts the LTOT rate.

Together with the fact SUPPLY_CNT_on_day0 contributes the most in feature importance for several models, we concluded that the models relied too much on this particular feature, potentially being swayed by confounding variables. To reduce the impact of SUPPLY_CNT_on_day0, we divided features by SUPPLY_CNT_on_day0 and created interaction terms, providing an incremental improvements on our ROC AUC scores. This improved on the shortcomings of our previous model, visualized in the graph below.

Before (left) and After (right):



Model Exploration

As we began to explore potential models, we divided the data into three pieces:

- training set: consisting of $\frac{2}{3}$ of the features derived from HMAHCC_COMP
- validation set: consisting of $\frac{1}{3}$ of the features derived from HMAHCC_COMP
- holdout set: the features derived from HMAHCC_HOLDOUT

We would use the training set to test models, the validation set to estimate out-of-sample accuracy and ROC AUC score, and finally only use the holdout set while making our final predictions.

Logistic Regression

Because this is a classification problem, we first tested all of our features using simple Logistic Regression. Logistic Regression provided a weak predictor, but the results allowed us to efficiently gain an intuition for testing new features and various hypotheses.

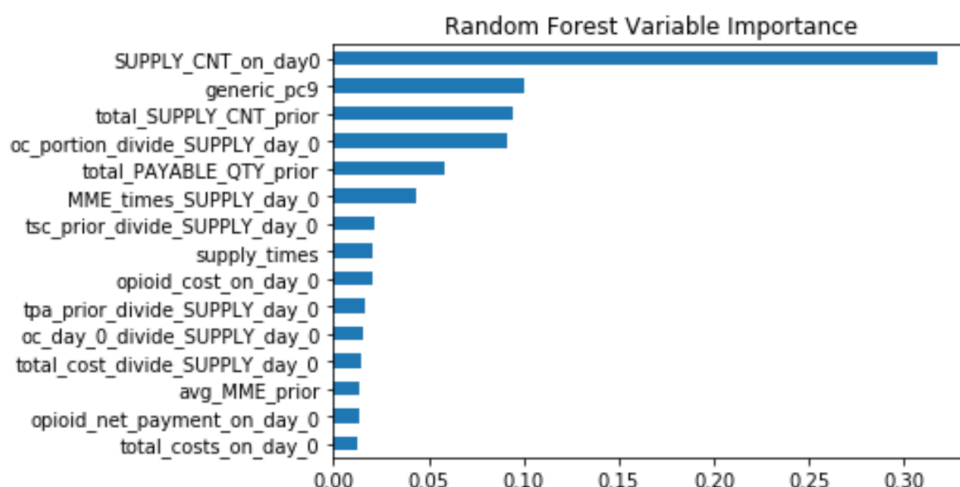
While we did not incorporate this into our final solution, it provided a baseline to derive some less obvious information provided by the dataset.

Random Forests

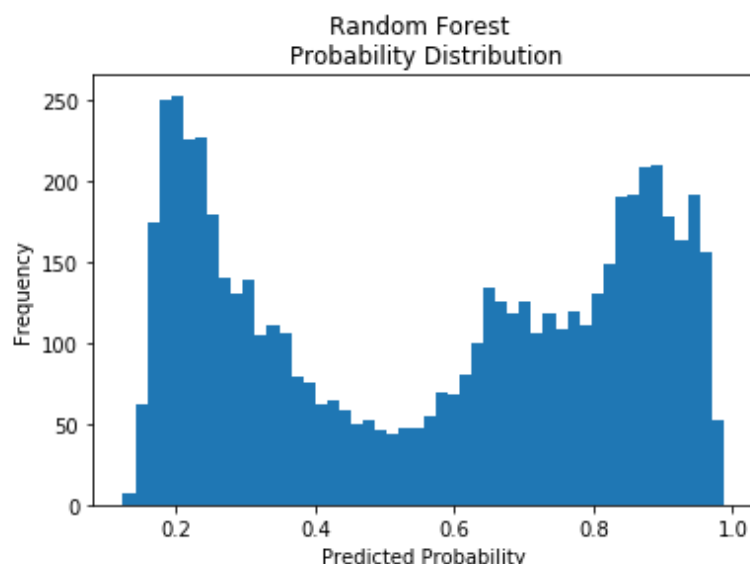
Early in our analysis we used a random forest classifier, an ensemble decision tree model that uses bootstrap aggregation (referred to as bagging) to essentially simulate a larger dataset.

In order to adjust our model's hyperparameters, we used the grid search method to adjust features such as the decision tree's Max_depth and min_samples_leaf to prevent overfitting, while other parameters such as n_estimators and max_features were set at a higher level to increase model complexity.

On the validation set, this model provided an ROC AUC score of 0.9126



While this model performed well in terms of ROC AUC score, we questioned the distribution of probabilities it predicted:

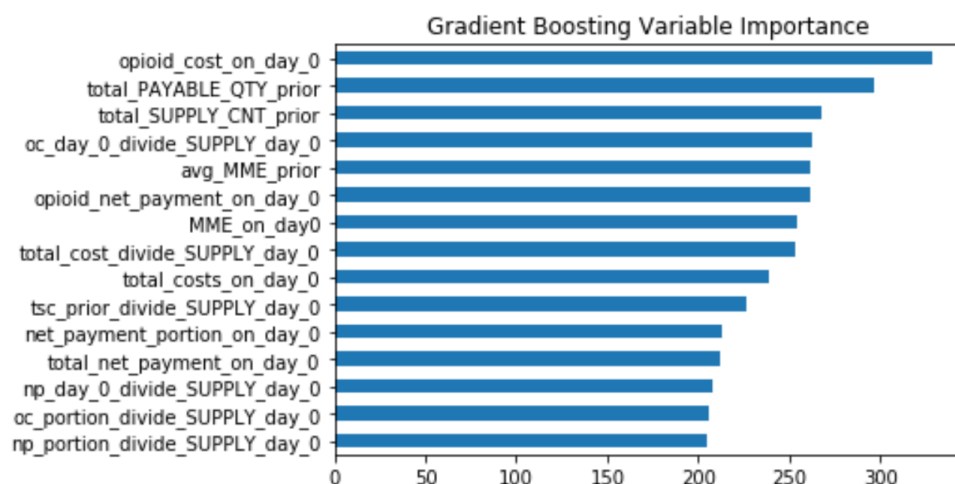


Gradient Boosting Tree

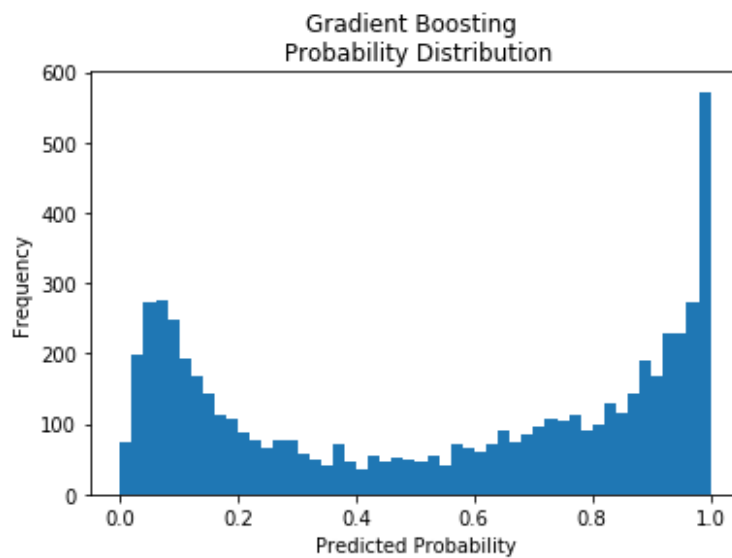
Next we tried another variation of decision trees, a gradient boosting tree. This model sequentially adds predictors to a decision tree ensemble, each one correcting its predecessor. This method attempts to fit the new predictor (or successor) to the residual errors made by the previous predictor/predecessor. In general, gradient boosting trees generally have shorter training time than random forests as they train much fewer trees.

Here we built Gradient Boosting Trees using LightGBM package, design to maximize runtime efficiency. We also applied a grid search to tune hyperparameters: we choose a higher `n_estimators` to increase complexity, and at the same time we set a lower `max_depth` at 3 to prevent overfitting.

This tuned model achieved an AUC score of 0.9278 on the validation set, which was the best performance across all other models, and again provided the additional benefit of a ranking of variable importance:



Again, while this model performed well in terms of ROC AUC score, we questioned the distribution of probabilities it predicted, even more dramatic than the previous model:

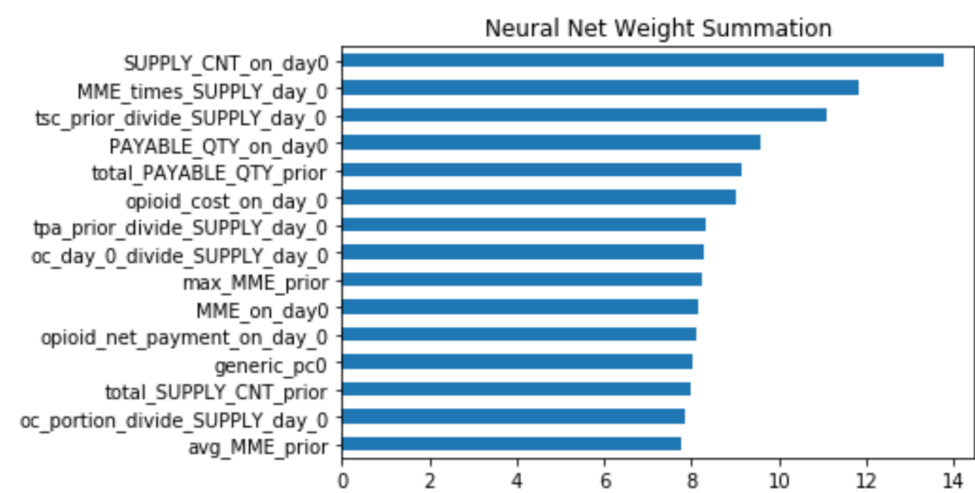


Neural Networks

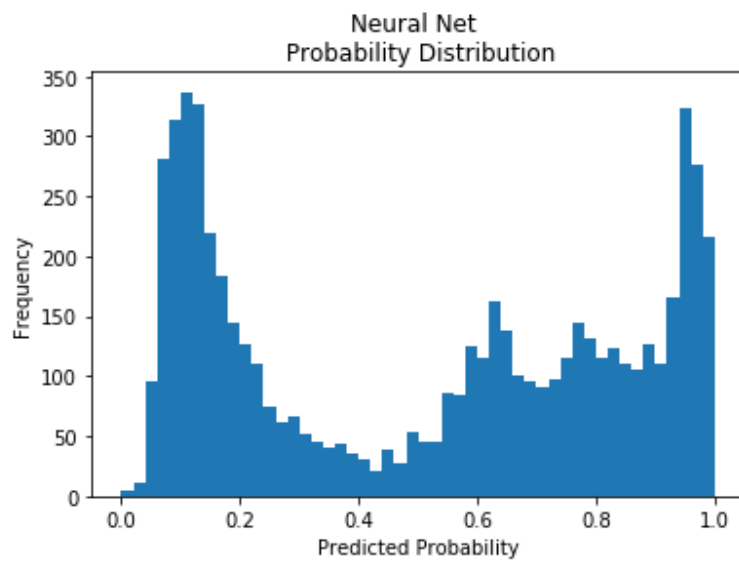
Neural Network models are built with layers of neurons. Neurons of a layer are determined by a combination of weights and inputs from the previous layer, as well as the activation function. The process allows Neural Networks to model complex nonlinear relationships between features and response variables. With proper tuning, it generally achieves decent performance.

At the tuning stage, we applied early stopping to prevent overfitting. Grid search tuning indicated that we should use 128 neurons on the first hidden layer and 64 neurons on the second hidden layer. However, this Neural Network model underperformed the two other tree-based models on this dataset, with an ROC AUC score of 0.8713.

While neural nets do not naturally produce a measure of variable importance, we were able to derive a measure of this by examining the sum of the absolute values of the weights between the input layer and the first hidden layer. While this does not account for the behavior and the additional hidden layers, it does provide some intuition:



Once again we questioned the distribution of probabilities it predicted, which almost appears trimodal (thus difficult for a threshold to separate classes):



Final Model Selection

Given these three models, all with relatively adequate ROC AUC scores, we decided to create an ensemble model that averaged the probabilities of any given pair of predicted probabilities, yielding the following scores:

Individual Models:

- **Gradient Boosting ROC AUC:** 0.9278
- **Random Forest ROC AUC:** 0.9126
- **Neural Net ROC AUC:** 0.8713

Two Model Averages:

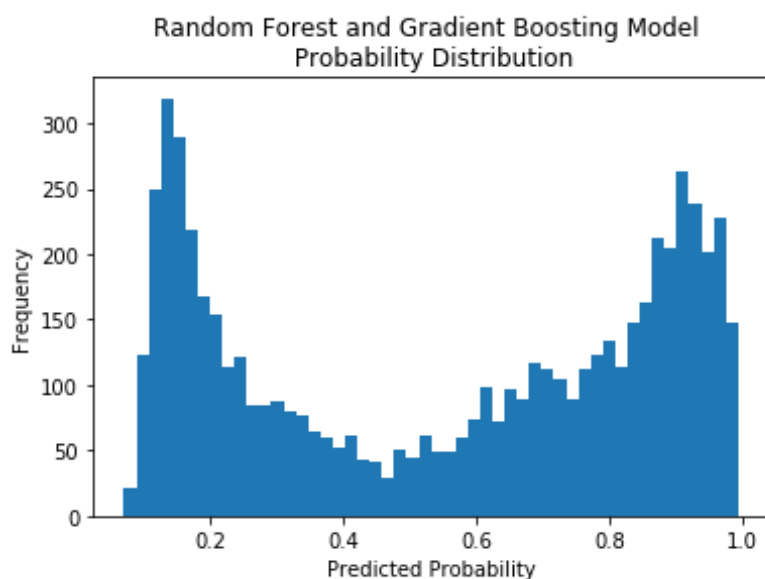
- **Gradient Boosting and Random Forest ROC AUC:** 0.9255
- **Gradient Boosting and Neural Net ROC AUC:** 0.9135
- **Random Forest and Neural Net ROC AUC:** 0.8984

Three Model Average:

- **3 Model Average ROC AUC:** 0.9154

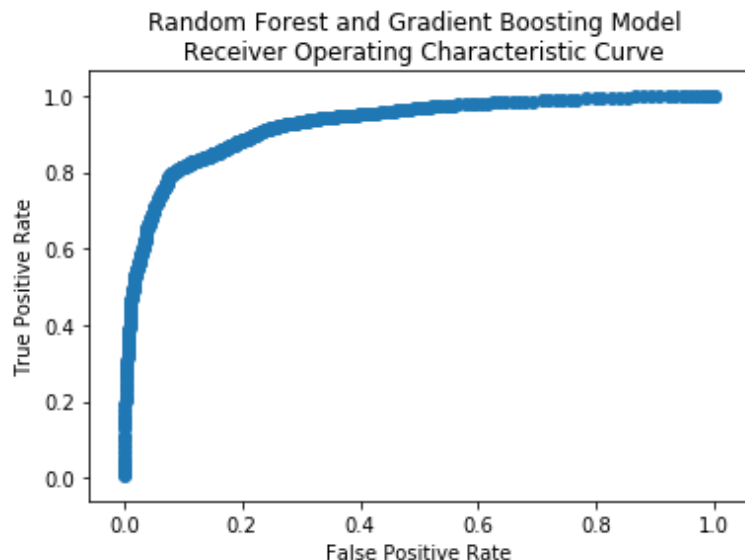
While the best overall model solely in terms of predictive power was given by gradient boosting, we choose the combination of gradient boosting and random forest as our final model for the following reason:

- the difference in ROC AUC score was nearly negligible (<0.0035)
- taking an ensemble of two decision trees gives a large degree of interpretability
- using two models with different variable importances increases the ability of the model to handle diverse input
- in examining the distribution of predicted probabilities, we found that this ensemble best matched our expectation of LTOT (see figure below)



To further gain intuition regarding the performance of our model, we calculated the result of setting different classification thresholds for our model, trying each value between 0 and 1 with a step size of 0.002

Given the validation set, this process returned a threshold of 0.586, aligning with our prior knowledge of the LTOT distribution given in our training set. From this process we were also able to construct the ROC curve of our validation set:



Additionally this allowed us to construct a confusion matrix for the validation set:

Model Prediction: RF+LGB False True

LTOT

False	2027	225
True	443	1827

Final Analysis and Actionable Insights

Opioid addiction is inherently caused by exposure to opioids and the prescribed dosage (amount and strength) at the point of being opioid naive. According to our analysis, there is no evidence of "drug seeking behavior" or any other correlations that would otherwise suggest causation. Because of this, we base our recommendations primarily on the prescription at the time of the "anchor date".

Prior to the point of prescribing a patient, we recommend introducing our model into doctors' decision-making process. Using our model as a communication tool, doctors can educate patients on their likelihood of becoming addicted according to their prescribed dosage. Furthermore, taking our model into account, doctors can recommend a reduced dosage that is less likely to induce LTOT.

Considering the variable importance (see graphs above), the two most actionable and significant variables are aspects of the initial prescription:

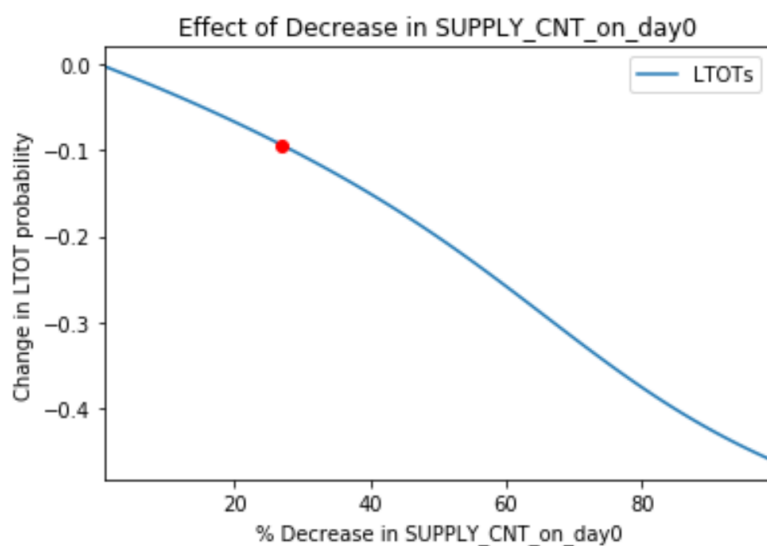
- supply count on day zero
- MME on day zero

Both of our directly actionable items can be controlled by the prescribing doctor. By using our model as a company-wide guideline for what levels of prescription are likely to produce an LTOT patient, we can incentivize doctors to make decisions that reduce the overall amount of LTOT patients. For example, a 27% reduction in supply count on day zero would result in a 10% decrease in LTOT probability, as determined by logistic regression (see chart below).

Beyond these two immediately actionable variables, the next most important to consider are:

- total payable quantity prior to day zero
- total supply count prior to day zero

It is important to note that by reducing the impact of our immediately actionable variables, we can then begin to impact the secondary variables of importance in the long term. This is because prior exposure to opioids still plays a significant role in being susceptible to LTOT classification.



Graph Interpretation: To estimate the affect of decrease in SUPPLY_CNT_on_day0. We ran logistic regressin on modified training data where we graduatly decreased the feature SUPPLY_CNY_on_day0. The line in the chart shows how much we can reduce LTOT probability by decreasing SUPPLY_CNT_on_day0, while the red point shows 27% decrease in **SUPPLY_CNT_on_day0 can lead 10% reduction in probability of becoming LTOTs.**

What are the benefits of lowering LTOTs?

1. External Impact: Improve the quality of patients' and their families' lives

As described in the beginning of our report, opioid addiction can have a significant negative impact on the quality of one's life. It is not only a brutally agonizing process to overcome the symptoms of withdrawal, but also increases the likelihood that one will end up using heroin. Traumas like these effect the lives of the patient and the family. If avoided altogether, Humana can have a positive impact the world.

2. Internal Impact: Brand Image as a Social Reformer

Due to the positive impact of reducing LTOT's, Humana can position itself as a leading social reformer in the health insurance space.

3. Internal Impact: Cost Savings

If we can decrease the prabability of LTOT's by 10%, we can save \$2.8 million of costs on opioid drugs alone.

To derive this number, we used Humana's portion of the costs of all opioid prescriptions in the dataset to calculate the average costs of opioids per patient per year (see table below). We then used the Humana member rate (as of June 2019), the true population LTOT rate, additional costs from the table below, and the decrease in LTOT probability to calculate these savings.

	non-LTOTs	LTOTs
Average opioid costs per patient per year	26.349009	190.555065

```
In [1]: Humana_member = 16600000 # as of June 2019
LTOT_rate = 0.0104 # as provided by Humana
LTOT_extra_opioid_costs = (190.555065 - 26.349009)
decrease_LTOT_probability = 0.1

# Opioid costs saving =
'$' + str(round(Humana_member * LTOT_rate * LTOT_extra_opioid_costs * decrease
_LTOT_probability, 2))
```

```
Out[1]: '$2834853.35'
```

Furthermore, for each patient that successfully avoids becoming LTOT, Humana can capitalize on the cost savings in reduced costs of the highly volatile rehabilitation treatments.

Actionable Steps:

1. Start small, initially working with one hospital as a use case. Build a program/app that shows the probability of a patient becoming LTOT whenever a doctor intends to prescribe opioids. Predicting a probability of becoming LTOT based on a patient's insurance records takes milliseconds using our production ready code, making our implementation feasible for testing.
2. At this point, we can also leverage this effort to gain press exposure and build an image of trying to positively impact society and the health insurance industry.
3. Keep tracking these patients for a period of six months to one year. Then test whether or not the number of new LTOTs decreased compared to historical data.
4. At the final stage, begin expanding our model to be used in other hospitals around the globe.

Future Considerations:

1. **Regulation:** need to consult lawyers to determine an illegible way to incorporate the model into a doctor's decision process. While further legal clarification is required, we believe the risk is limited as our model does not discriminate against personal identities.
2. **Model Adjustments and Improvements:** Humana can incorporate data after day 0 into the model and further strengthen the prediction power.

Scalable Deliverable: Production Ready Code

One of the primary strengths that our code provides is an easily applicable template that allows for the model to automatically be adjusted over time and to take as input any training and test data that has the same attributes and format given in this competition.

In the file "humana_class.py" is a Python class that takes as input a dataframe with the structure given in the provided competition and holdout sets, and gives as its output all of the discussed features, including the derivation of the response variable for a given training set where data after the qualifying event is provided. Because of the portability of using a class object, we can supply our model with new data over time, and account for factors such as geographical differences.

Additionally, the model itself is also written as a Python class, taking as its input the extracted features from a pair of test and training sets. It has the capability to produce metrics for variable importance and setting prediction class thresholds, and again is able to handle input that changes over time as well as describe how these attributes change over time.

For our complete analysis and production-ready code used for feature extraction, please see the following links.

Feature Extraction:

<http://tiny.cc/tizkez> (<http://tiny.cc/tizkez>).

Final Analysis/Model*:

<http://tiny.cc/flzkez> (<http://tiny.cc/flzkez>).

*Github typically doesn't do a good job rendering Jupyter notebooks so please download the file to view.