White Paper:
Predicting Reply Rates

I. Goal

The reply rate has been established as an important step to increasing refill rates. After a person has chosen to reply to the refill reminder, whether they refill or not is almost random. Thus, the key idea here is that if one can get as many members to reply, the refill rates will increase as a spillover effect.

The objective of the model is to predict whether a patient would reply or not in order to recommend policy changes/ improvements for outreach to Kaiser. In order to do this prediction, a neural network was used.

II. Data Cleanup

Feature Selection:

We had to select the appropriate features for the model. Since we did not have a lot of raw information about each person, feature engineering had to be done.

First, demographic statistics about each patient like age, gender and race, which had a significant effect on reply rates, were included in the model. Since gender (female, male) and race (Asian, Black/ African-American, Hispanic/ Latino, Other/ Mixed, White, Unknown) are categorical variables, we did one hot encoding and excluded Male and White from the model to avoid perfect multicollinearity.

Next, the Community Needs Index (which is at the zip code level) SDOH index (which is at a census tract level) and other census-tract level scores were also included. The census-tract level scores were obtained using world bank API and were the following: percentage poverty, percent non-white population, percent of population that has poor English, Unemployment rate, percent of population with no high school education, percent of population with no insurance, percentage of renters, percent of households on SNAP benefits and percent of population with no car.

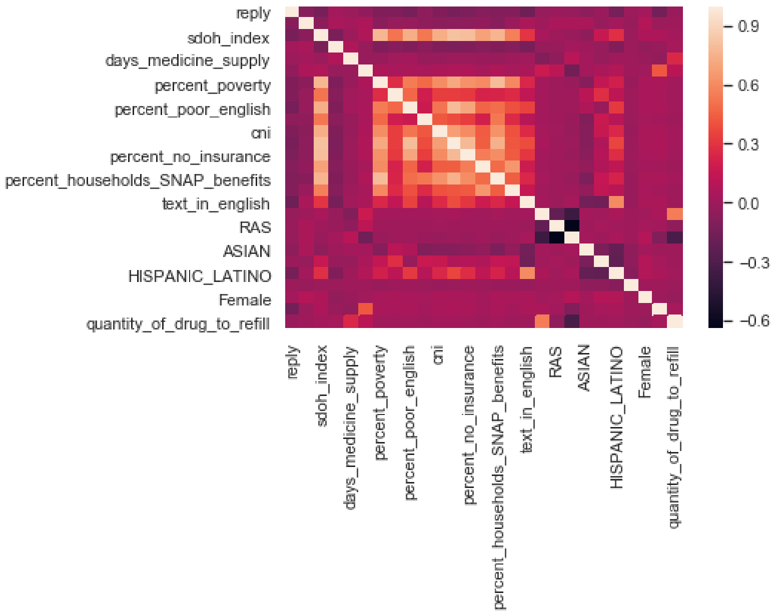
Next, the preferred language of text (English, Spanish), type of drug being refilled (DM, STATIN, DOAC, RAS), amount of copay, number of days for which the drug would last after that refill and the absolute quantity of the drug was also included in the model. Language of text and type of drug were one hot encoded and Spanish and DOAC were not included in the model due to multicollinearity.

Finally, some feature engineering was done to find the number of drugs one patient had been messaged about. The maximum possible value was 4, in which case it meant the patient had been messaged asking to refill all 4 drugs. The minimum possible value was 1, which meant the patient had been messaged to refill only 1 drug. This served as an approximate proxy for the number of comorbidities a patient might have. Next, a binary variable, which indicated whether a patient had been messaged previously or not was also included in the model.

Correlation Matrix:

The features are individually correlated with the predicted variable. However, in order to make sure we do not fall into the multicollinearity trap, features which had a correlation higher than 0.8 were dropped from the model. Based on this condition, percent of population with no high school education and percentage of renters were dropped from the model. This reduces the bias in the model and makes the learning faster.

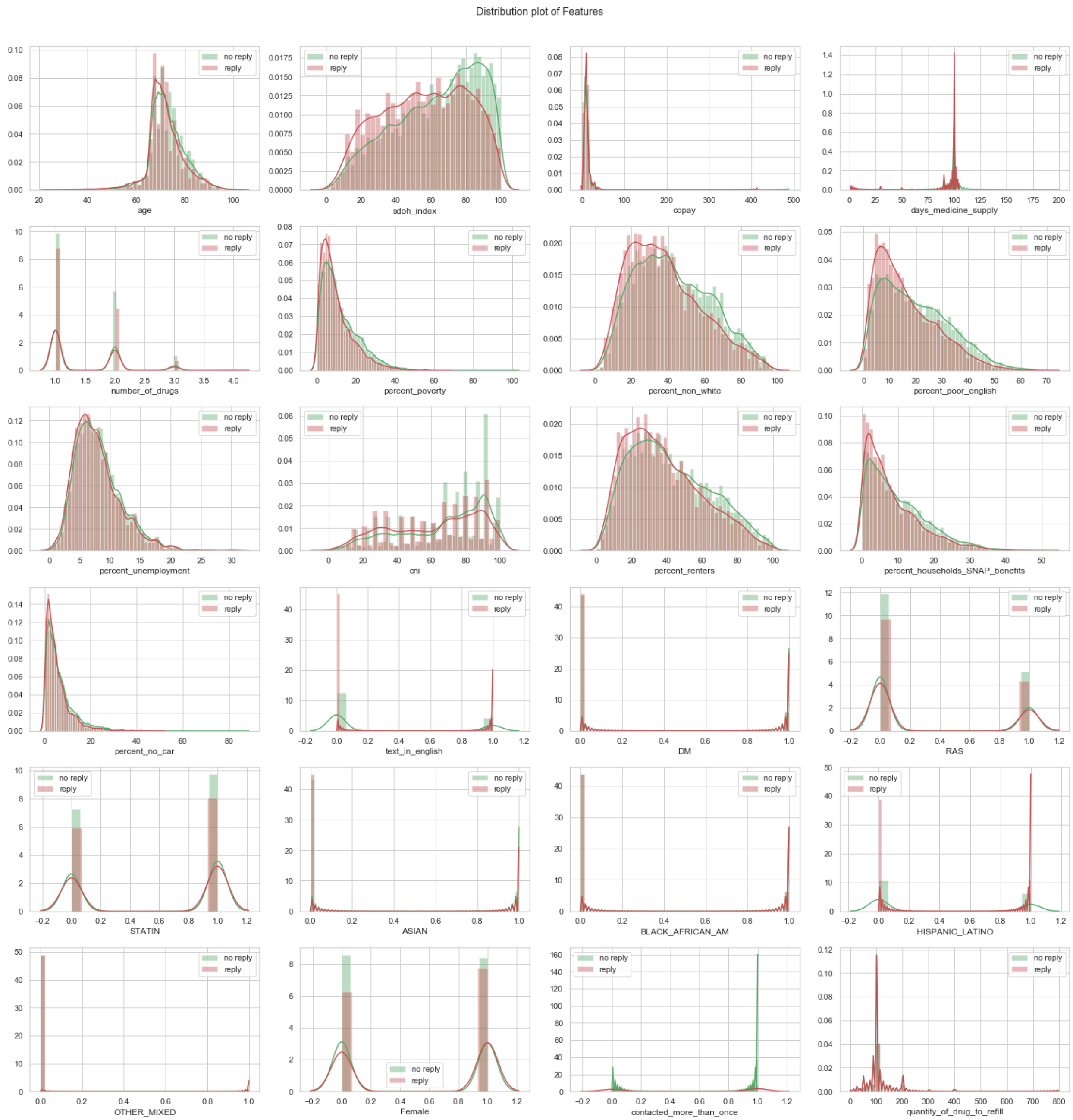
The following is a heat map of the correlation matrix of all the features in the model.



Missing Data:

In order to deal with missing values, for all the observations for which at least one value of a column was missing or Nan, the entire observation was dropped.

Overall, there were 24 variables in the model. The following are the distribution plots of reply against all the features.

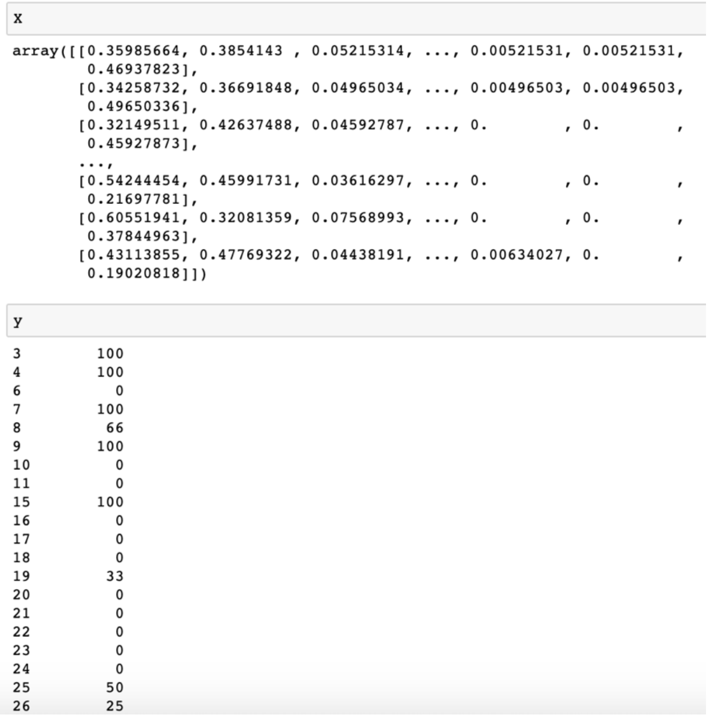


<u>Normalizing data:</u>
Since the features varied from each other in their scales, in order to unbias the weights of the features, the data was normalized using sklearn's normalize function.

<u>Modifying the predicted variable:</u>
The data was structured in such a way that each row represented an instance of a unique series of events after a reminder was sent to a patient. Since there were patients who had been messaged more than once (for different drugs or at different times), this meant that each row was not unique to each patient. In order to deal with this conflicting data, the data was contracted into a structure where each row represented a unique member for a unique drug by taking the average

of the reply binary. This means that if a member had been contacted three times asking them to refill a drug, for example STATIN and they had replied two times out of three times, the ‘reply’ value would now be (1+1+0)/ 3 = 0.66. This was converted to a percentage by multiplying by 100, thus it would become 66%. This solved the problem of contradictory data and converted the reply variable into a continuous variable.

The following is a snapshot of the data from the python script of the array of features and the predicted variable.



```
X

array([[0.35985664, 0.3854143 , 0.05215314, ..., 0.00521531, 0.00521531,
         0.46937823],
        [0.34258732, 0.36691848, 0.04965034, ..., 0.00496503, 0.00496503,
         0.49650336],
        [0.32149511, 0.42637488, 0.04592787, ..., 0.        , 0.        ,
         0.45927873],
        ...,
        [0.54244454, 0.45991731, 0.03616297, ..., 0.        , 0.        ,
         0.21697781],
        [0.60551941, 0.32081359, 0.07568993, ..., 0.        , 0.        ,
         0.37844963],
        [0.43113855, 0.47769322, 0.04438191, ..., 0.00634027, 0.        ,
         0.19020818]])

y

3         100
4         100
6           0
7         100
8          66
9         100
10          0
11          0
15        100
16          0
17          0
18          0
19         33
20          0
21          0
22          0
23          0
24          0
25         50
26         25
```

III. Method

Model used:

Once the data was cleaned, it was divided into a training (70% of total data) and a test set (30% of total data). Then, MLPRegressor from the sklearn module was used to train the model. The logistic activation function was used, alpha level was set to 0.0001 and learning rate was adaptive and maximum iterations were 2000. This model predicted a number between 0-100, which represented the likelihood of an individual to refill.

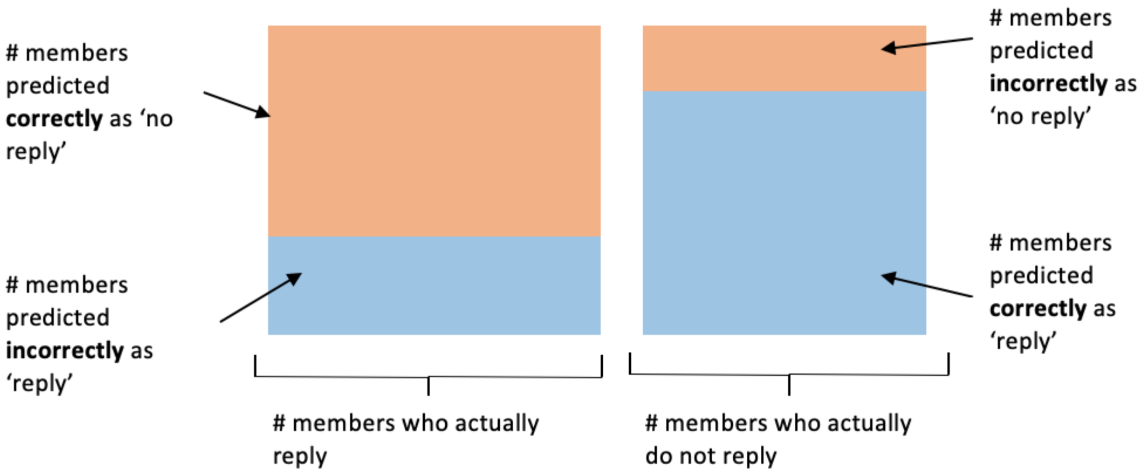
Converting Continuous Reply rate likelihood to a Binary Value (0- no reply/100- reply):

As stated previously, the primary outcome of this model was to increase outreach to the members who would be predicted as not replying. Keeping this in mind, a cutoff point, between 0 to 100, had to be decided which would place members into either a ‘reply’ or a ‘not reply’ bucket. All the actual reply values that were between 0-50 (50 inclusive) were put into a box representing ‘not reply’ and all the actual reply values between 50-100 were put into a box representing ‘reply’. Thus, in the case of the actual reply values, the cutoff point was 50. Now, a parallel cutoff point had to be decided for the predicted values. In order to decide this cutoff point, we had to decide which statistic we would use to test performance of the model, and the cutoff point would have to be such that it would maximize this statistic.

Important Statistic:
Given the objective of the model, three values were chosen as important to maximize. The table below gives a description of the statistics and the reason why they are important.

Statistic	**Reason**
% Of no reply captured by model = $\frac{\textit{\# members with predicted no reply and actual no reply}}{\textit{\# members with actual no reply}}$	We want to make sure that we are capturing at least a majority of the members who do not reply, if not all.
% Of reply captured by model = $\frac{\textit{\# members with predicted reply and actual reply}}{\textit{\# members with actual reply}}$	Similarly, we want to make sure that we do not overpredict the members who do not reply, by also simultaneously optimizing this statistic, which acts as a check statistic above.
% Of accurately predicted no replies = $\frac{\textit{\# members with predicted no reply and actual no reply}}{\textit{\# members with predicted no reply}}$	We are going to conduct extra outreach towards all the people that the model predicts as no reply. Since resources are expensive, we want to ensure that if we reach out to people, we are not wasting resources.

Below is a visual representation of the statistics. Maximizing the above three statistics would in a way aim at reducing the number of incorrect predictions.

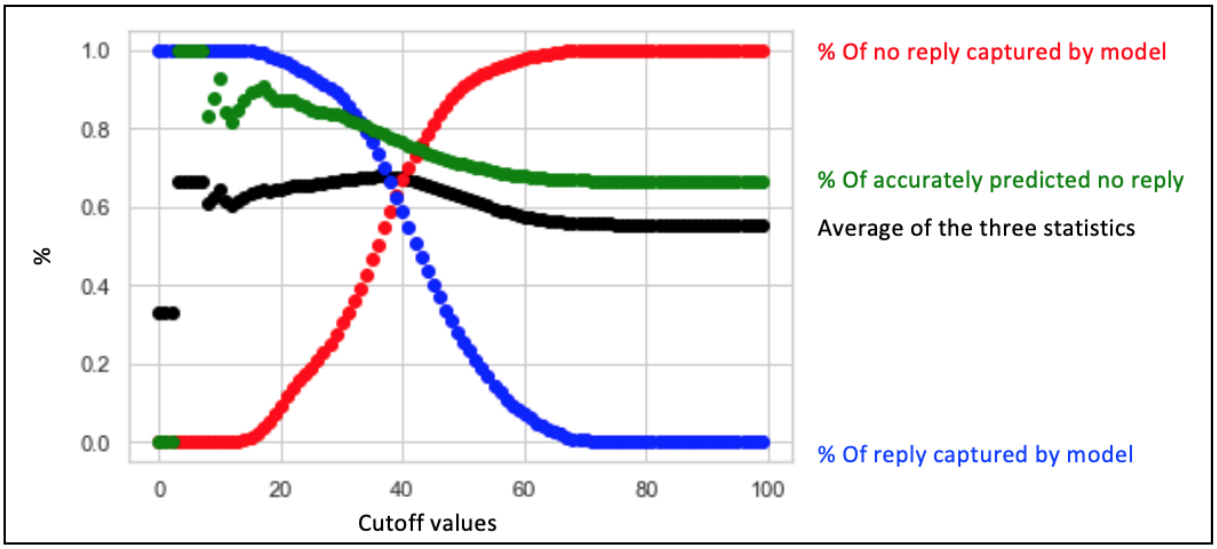


# members predicted **correctly** as 'no reply'
# members predicted **incorrectly** as 'reply'
# members predicted **incorrectly** as 'no reply'
# members predicted **correctly** as 'reply'
# members who actually reply
# members who actually do not reply

Maximizing Statistics

In order to maximize all three of the statistics together, since there is a tradeoff between the three, a cutoff point had to be decided in such a way that all three were optimally high but none too high or too low. To this end, an average of the three values was taken and the goal was to find a cutoff point which maximized this value. The average of the three values was calculated for all possible cutoff points, i.e., all integers between 0-100 inclusive. There was a global maxima for this average function, the domain of which was from 0 to 100. Thus, the point at which the global maxima occurred was chosen to be the cutoff point.

This is a plot of the three statistics and the average at every possible cutoff point. The average is the highest at point 37, which would be set to the cutoff point.



IV. Results:

The model was trained, and cross validation was conducted after splitting the data into testing and training sets randomly 8 times. From this, taking the average cutoff each time, the cutoff can be set at 37. Keeping this cutoff, of the members the model predicts as no reply, if we conducted outreach to those members, we have a 78.54% chance of having correctly used the resources of the company. Next, of all the members who did not refill, the model accurately predicts 54.31% of them. Further, of all the people who do reply, the model accurately predicts 70.48% of them.

Table of results

```
------------------------------------------------------------
 Average Cutoff: 37.0
------------------------------------------------------------
 Average % of no reply captured by model: 0.5430990560100948
------------------------------------------------------------
 Average % of reply captured by model: 0.7047707998186025
------------------------------------------------------------
 Average % of accurately predicted no reply: 0.7854051953528439
------------------------------------------------------------
```

V. Policy implications:

The model captures slightly more than half of the people who do no refill. Of the people who we would increase outreach towards, we are sure that approximately 78% of them actually need it. This is desirable since we hope to reach out to as many people who would not refill as possible. However, if we are conducting extra outreach, we would want to make sure that that outreach is being directed to the right people. There is a tradeoff between these two numbers, we could increase the number of people we capture who don't refill, by simply increasing the cutoff point. However, we would also be decreasing the accuracy of the people we do reach out to and would be wasting resources.

Thus, being on the more conservative side is ultimately a more beneficial allocation of resources.