H1N1 Vaccine Prediction

Group 10
Eric Chen
Smitha Kannanaikkal
Allen Lee
Hyoungmin (Stella) Lee
Mariana Carini

Abstract

The purpose of this project is to use the National 2009 H1N1 Flu Survey dataset to develop a machine learning model that predicts the demand of H1N1 vaccines. Several machine learning algorithms including Naive Bayes, Decision Tree, and Random Forest classifiers were used to build the models. Preprocessing techniques such as forward feature selection and oversampling were used in constructing the model as well. A detailed comparison showed that the oversampled and forward selected Naive Bayes model proved to be the best performing model. With it, an estimated 107.1 million doses of H1N1 vaccines was predicted for the 2009 pandemic.

Table of Contents

Introduction

The ongoing pandemic has shot the world into a paralyzed state; its impact affecting the lives of millions around the globe. The invaluable insights generated by data analysts all over the world have helped us track and monitor the impact of the pandemic, and served as guidelines to bring our lives back to normalcy. Now with a vaccine quickly developing, some may see this as a sign of relief and wonder if this is the glimpse of the light at the end of the tunnel. Well, not quite. There are many logistical questions regarding the vaccine that need to be answered first, such as transport coordination, how many to produce, or who and where should get it first. In this study, we will be focusing on the quantity of vaccine production and how one might prepare for such a task by revising past data the 2009 pandemic H1N1, also known as the Swine Flu.

Business Implication

We will be using the National 2009 H1N1 Flu Survey dataset provided by the National Center for Immunization and Respiratory Diseases (NCIRD) to predict whether a respondent will receive a vaccination for H1N1. From that, we will infer an estimation for the amount of vaccines to be produced.

The ability to accurately predict the demand of vaccines is important for several reasons. In these dire times, it is reasonable to think that we should just create as many vaccines as possible so that anyone who needs it shall get it. However, this is simply not the case. Although big pharmaceutical companies like Pfizer have every intention to help the world, we have to understand that at the end of the day it is a business. From a pharmaceutical company's perspective, overproduction could mean money and costly resources wasted on production and storage, and underproduction will lead to a shortage of supplies and missed opportunities. According to this article, the US government ordered 250 million doses of H1N1 influenza vaccine for a cost of $2 billion. Even a fraction of these numbers could mean a difference of hundreds of thousands, even millions, of dollars made or lost.

Furthermore, in a report of the H1N1 Vaccine Deployment Initiative, authored by the World Health Organization (WHO), it described a number of challenges faced with regards to shipping and handling the vaccines due to its complex nature. Since the vaccines are temperature-sensitive, the logistics of cold-chain packaging must be carefully planned out to ensure appropriate conditions during transportation. To be specific, one of the main challenges

encountered was figuring out how to transport such large quantities of vaccines with limited transportation options. In addition, the report also mentioned that many healthcare providers were unable to properly plan to receive such high volumes of deliveries. To be able to generate insights about the demand for vaccines will certainly be beneficial and help us in organizing our combat against current and future pandemics.

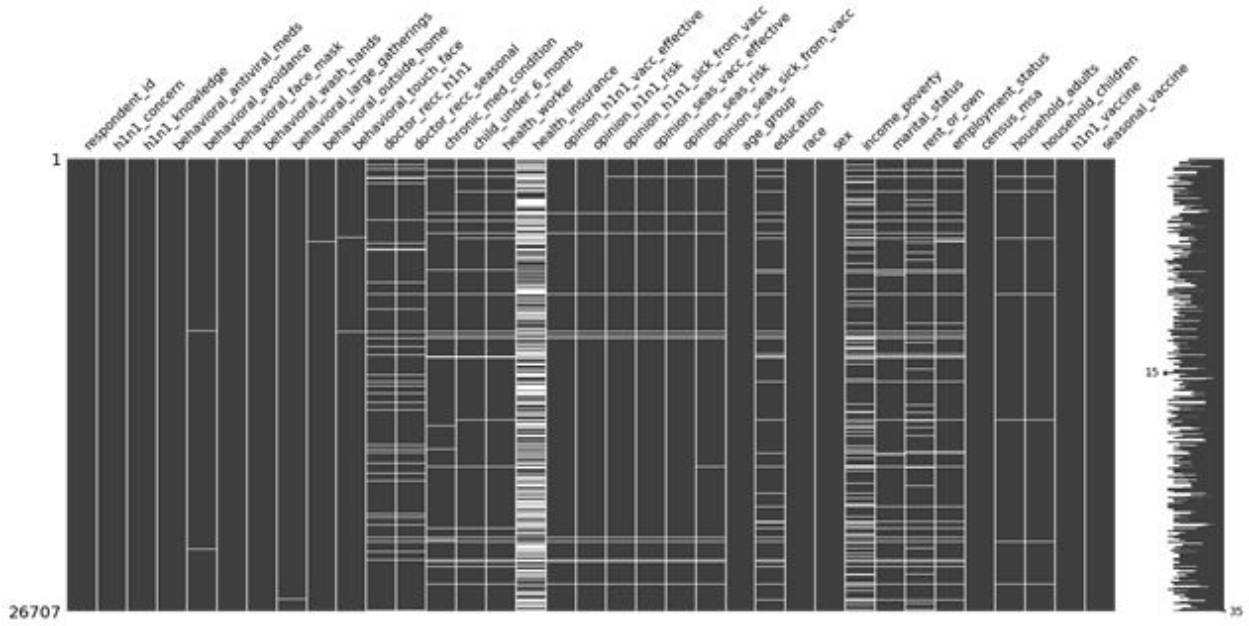
Data

Our dataset was acquired from a competition listed on DrivenData, an online platform where a global community of data scientists come together to solve difficult predictive problems. The dataset was a survey sponsored and conducted by the NCIRD in conjunction with the National Center for Health Statistics (NCHS) and the Centers of Disease Control and Prevention (CDC). The responses were generated through a random-digit-dialing telephone survey of households in the United States, and was designed to produce estimates of vaccination coverage rates for both H1N1 and seasonal influenza. The raw dataset contained 26,706 responses and 38 attributes. A table of attribute description is listed below.

Attribute	Description
h1n1_conern	Level of concern about the H1N1 flu. 0 = Not at all concerned; 1 = Not very concerned; 2 = Somewhat concerned; 3 = Very concerned
h1n1_knowledge	Level of knowledge about H1N1 flu. 0 = No knowledge; 1 = A little knowledge; 2 = A lot of knowledge
behavioral_antiviral_meds	Has taken antiviral medications. (binary)
behavioral _avoidence	Has avoided close contact with others with flu-like symptoms. (binary)
behavioral_face_mask	Has bought a face mask. (binary)
behavioral_wash_hands	Has frequently washed hands or used hand sanitizer. (binary)
behavioral_large_gatherings	Has reduced time at large gatherings. (binary)
behavioral_outside_home	Has reduced contact with people outside of own household. (binary)
behavioral_touch_face	Has avoided touching eyes, nose, or mouth. (binary)
doctor_recc_h1n1	H1N1 flu vaccine was recommended by doctor. (binary)
doctor_recc_seasonal	Seasonal flu vaccine was recommended by doctor. (binary)
chronic_med_condition	Has specified chronic medical conditions. (binary)

child_under_6_months	Has regular close contact with a child under the age of six months. (binary)
health_worker	Is a healthcare worker. (binary)
health_insurance	Has health insurance. (binary)
opinion_h1n1_vacc_effective	Respondent's opinion about H1N1 vaccine effectiveness. 1 = Not at all effect; 2 = Not very effective; 3 = Don't know; 4 = Somewhat effective; 5 = Very effective
opinion_h1ni_risk	Respondent's opinion about risk of getting sick with H1N1 flu without vaccine. 1 = Very low; 2 = Somewhat low; 3 = Don't know; 4 = Somewhat high; 5 = Very high
opinion_h1n1_sick_from_vacc	Respondent's worry of getting sick from taking H1N1 vaccine. 1 = Not at all worried; 2 = Not very worried; 3 = Don't know; 4 = Somewhat worried; 5 = Very worried
opinion_seas_vacc_effective	Respondent's opinion about seasonal flu vaccine effectiveness. 1 = Not at all effective; 2 = Not very effective; 3 = Don't know; 4 = Somewhat effective; 5 = Very effective
opinion_seas_risk	Respondent's opinion about risk of getting sick with seasonal flu without vaccine. 1 = Very low; 2 = Somewhat low; 3 = Don't know; 4 = Somewhat high; 5 = Very high
opinion_seas_sick_from_vacc	Respondent's worry of getting sick from taking the seasonal flu vaccine. 1 = not at all worried; 2 = Not very worried; 3 = Don't know; 4 = Somewhat worried; 5 = Very worried
age_group	Age group of respondent (categorical)
education	Self-reported education level (categorical)
race	Race of respondent
sex	Sex of respondent
income_poverty	Household annual income of respondent with respect to 2008 Census poverty thresholds. (categorical)
marital_status	Marital status of respondent
rent_or_own	Housing situation of respondent
employment_status	Employment status of respondent
hhs_geo_region	Respondent's residence using a 10-region geographic classification defined by the U.S. Dept. of Health and Human Services.
census_msa	Respondent's residence within metropolitan statistical areas (MSA) as defined by the U.S. Census.
household_adults	Number of *other* adults in household, top-coded to 3.
household_children	Number of children in household, top-coded to 3.
employment_industry	Type of industry respondent is employed in. Values are represented as random character strings.
employment_occupation	Type of occupation of respondent. Values are represented as random character strings.
seasonal_vaccine	Whether respondent received seasonal flu vaccine
h1n1_vaccine	Whether respondent receive H1N1 flu vaccine

Data Cleaning



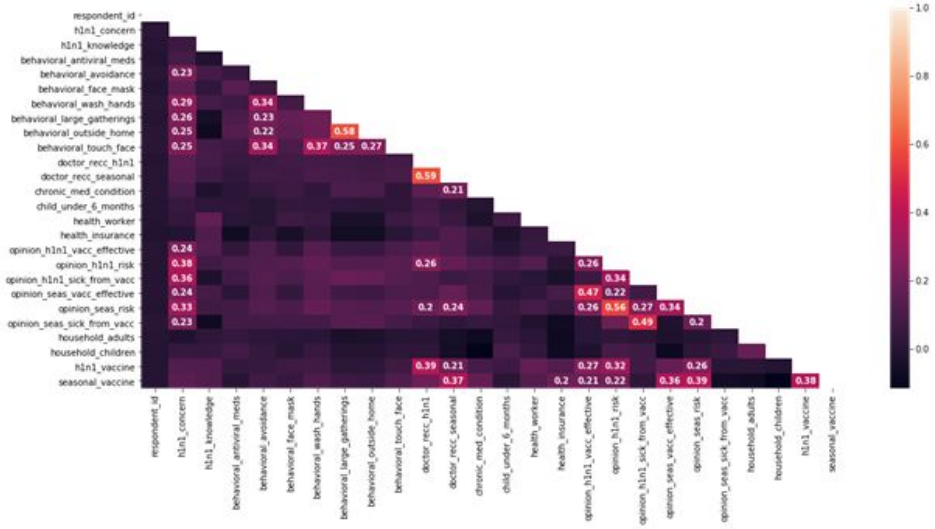
We start off by taking a quick look at the number of missing values in each attribute in our dataset and determine our strategy for how to handle them. We dropped any observations that had at least ten missing values, or one-third of its responses missing, which resulted in around 500 rows dropped. It seems that many features have some degree of missing values but the health_insurance variable has nearly 50% of its responses missing. This specific attribute will require a little bit more preprocessing work, so let us come back to it later and try to handle other variables first.

From the visualization above, we observe that there seems to exist a pattern in the missing values for doctor_recc_h1n1 and doctor_recc_seasonal. Upon further investigation, we found that all of the instances where one of them is null, the other is null as well. Since we observed such a pattern and realized that the missing values in these attributes were not random, we decided to consider the missing values as a level and filled them in with “no_response”. We also considered missing values in variables like education, income, and employment status as a level and filled them with “no_response” as well. For missing values in other binary variables, which only had a small portion of NA’s, we filled them in with the mode of the responses for that variable. As for other categorical variables with numeric levels, which also only had a small portion of NA’s, we filled them in with the median of the responses of those variables.

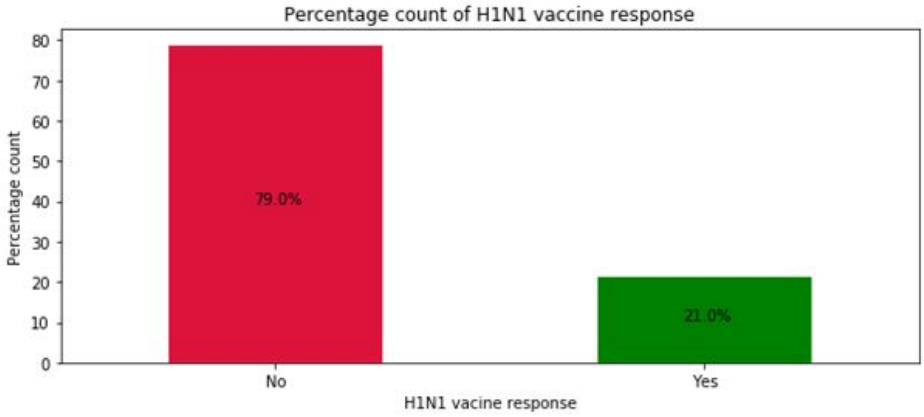
Now that we have filled in the missing values for all other features, we will begin our preprocessing steps for health_insurance. Since the proportion of missing values is so large in this attribute, we believe that filling in the NA's intuitively would not be the correct approach. Instead, we decided to predict the missing values of health_insurance with Naive Bayes algorithm using the rest of the features. To do this, we first separate our dataset into those who responded in health_insurance and those who did not. Then, we divided the dataset that contained those who did respond into a training and testing set using an 80-20 split. We ran a correlation matrix to see which features are strong predictors for the health_insurance variable, however, the results were inconclusive. Therefore, we ran the Naive Bayes model with all features to predict health_insurance. The model that we obtained was able to predict the variable with 82% accuracy and yielded an F1-Score of 0.90. Now that we have our model, we recombined the training and testing set and ran our model again using the whole dataset. Then we applied our Naive Bayes model on the set where respondents did not respond for health_insurance and predicted those missing values. For our final dataset, our group has decided to drop the respondent_id and hhs_geo_region columns as they are irrelevant. The employment_industry and employment_occupation columns were dropped as well since they are uninterpretable without the key to decode the random strings.

Feature Exploration

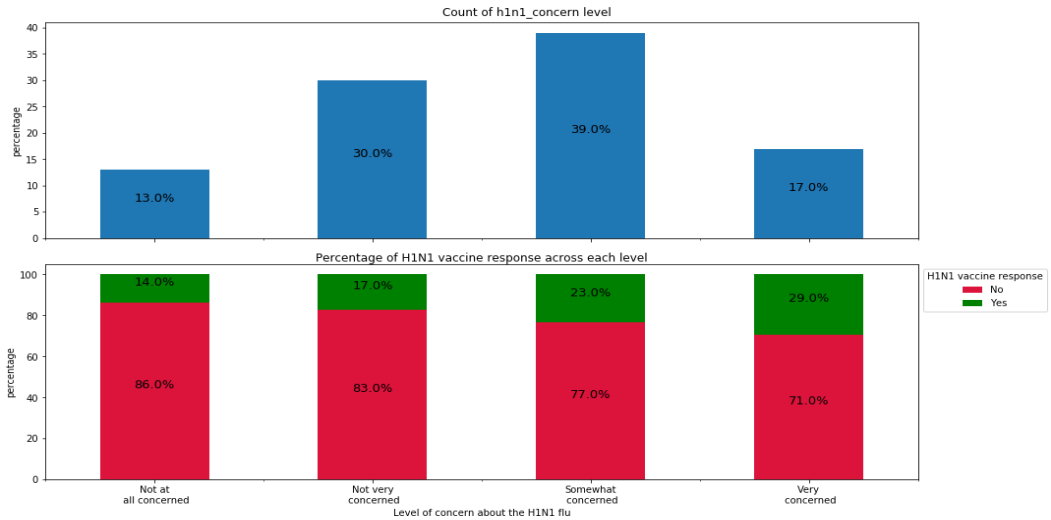
With our cleaned dataset, we performed some exploratory analysis to get a better feel for our data. First we take a look at the correlation plot below of all of our attributes. We can see that there is some positive correlation between variables like doctor_recc_h1n1 and doctor_recc_seasonal, and opinion_h1n1_risk and opinion_seas_risk. Overall, there is not much correlation between features in our dataset.



Moving on to the graph below, our class of interest in our target variable, which is a yes response to receiving an H1N1 vaccine, occurs far less than a no response. This implies that we have a problem with class imbalance and is something we will have to keep in mind when building our models.



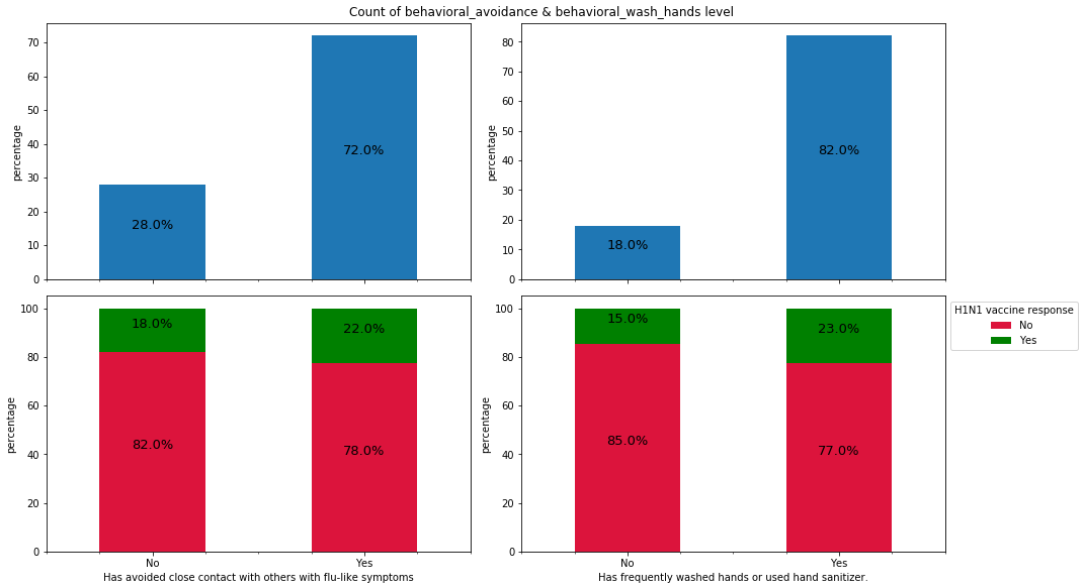
Next, we take a look at the distribution of the concern levels of H1N1 and the percentage of H1N1 vaccine responses across each level. Despite H1N1 being a pandemic, only 17% of total respondents were very concerned about it. The bottom half of the chart illustrates the percentage distribution of our target variable across each concern level, and our intuition of more people receiving the vaccine the more concerned they are with H1N1 is supported by the data.



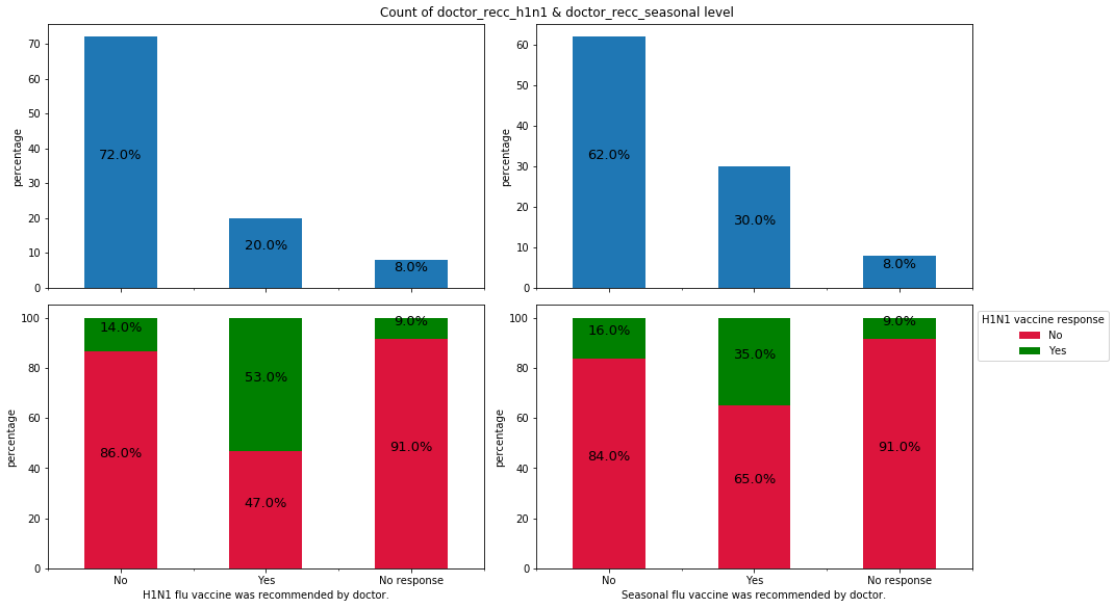
In the visualizations below, we analyze some of the behavioral attributes of respondents. The top two charts are distributions of participants' response to the following questions:

1. Has avoided close contact with others with flu-like symptoms
2. Has frequently washed hands or used hand sanitizer

As we can see, the majority of participants responded yes to these questions and the proportion of respondents receiving a vaccine is also higher in this group.



Finally, the plots below illustrate the distribution of H1N1 vaccine responses in features that indicate whether the respondent was recommended by a doctor to receive H1N1 and seasonal flu vaccines. As seen in the bottom-left graph, respondents get an H1N1 vaccine at a higher rate if they have been recommended by a doctor.



Analysis

Model Selection

Based on the nature of our dataset, we have selected the following algorithms to build our models:

- Naive Bayes
- Decision Tree
- Random Forest

Each model was built using the entire dataset and a 10-fold cross validation. We also built several versions of each model using forward feature selection and oversampling techniques. Since our target variable has class imbalance, we are not focused on the accuracy as the evaluation measure for the models. Instead, we will be evaluating each model based on other measures such as precision, recall, F1-score, and the AUC. It is also important to note that we are more interested in models that perform well for the class 1 response in our target variable, which is a participant responding yes to a vaccine. In addition, models that yield a high recall measure is also favored since we would like to minimize the number of false negatives, which are cases where a respondent was predicted to not have gotten a vaccine, but actually did. All of our models were built using Weka.

Model Building

Naive Bayes

The Naive Bayes classifier is a classification technique based on Bayes theorem and utilizes the concept of conditional probability. This algorithm assumes independence between predictors and performs well with large dataset, and it is very simple as well.

Our benchmark model yielded an accuracy of 79.99% with a weighted average F1-score of 0.806. The benchmark model appears to be strong, however, if we look at the numbers for

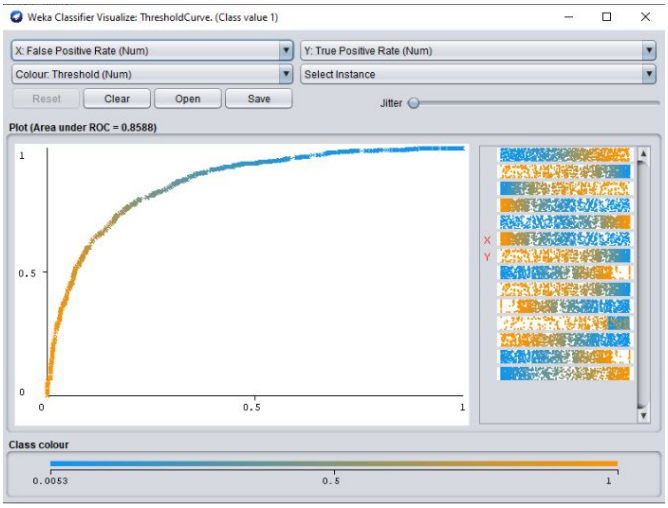
```
Correctly Classified Instances       20914               79.9954 %
Incorrectly Classified Instances      5230               20.0046 %
Kappa statistic                          0.4452
Mean absolute error                      0.2235
Root mean squared error                  0.3919
Relative absolute error                 66.6112 %
Root relative squared error             95.6871 %
Total Number of Instances            26144

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.845    0.367    0.895      0.845    0.869      0.448    0.828     0.940     0
               0.633    0.155    0.526      0.633    0.574      0.448    0.828     0.592     1
Weighted Avg.  0.800    0.322    0.816      0.800    0.806      0.448    0.828     0.866

=== Confusion Matrix ===

     a     b   <-- classified as
 17384  3185 |     a = 0
  2045  3530 |     b = 1
```

the class 1 outcome in our target variable, our model actually produced a fairly weak performance with a recall rate of 0.633 and F1-score of 0.574.

The table below is a summary of important evaluation measures produced by each version of our Naive Bayes models.

Model	Accuracy	Weighted Avg. Precision	Weighted Avg. Recall	Weighted Avg. F1-Score	Weighted Avg. ROC_AUC	Class 1 Precision	Class 1 Recall	Class 1 F1-Score	Class 1 ROC_AUC
NB_Benchmark	79.99%	0.816	0.800	0.806	0.828	0.526	0.633	0.574	0.828
NB_FS	84.35%	0.832	0.844	0.832	0.849	0.692	0.474	0.563	0.849
NB_OS	73.97%	0.740	0.740	0.740	0.822	0.740	0.740	0.740	0.822
NB_OS_FS	78.25%	0.784	0.784	0.784	0.856	0.793	0.769	0.781	0.856

For the second model we conducted, we implemented a forward selection on our features and the overall accuracy was the highest amongst all models. However, as mentioned before we are not interested in accuracy performance due to the class imbalance within our target variable. The class 1 recall rate for our Naive Bayes forward selected model is actually worse than our benchmark and the same goes for the F1-score. In our third model, we applied oversampling technique and we can see that the evaluation measures for class 1 outcome are significantly higher. Lastly, we created a fourth model with both oversampling and forward selection techniques applied. We came to the conclusion that this is the best approach for our Naive Bayes model since it has the highest numbers in class 1 measures and area under ROC curve. The AUC for the fourth model is shown below.



Decision Tree

A Decision Tree is an easy to understand algorithm that creates a flow-chart like structure that represents a series of logical rules. Each node of the tree is an attribute of the dataset that splits the observations into subgroups based on purity measures. Decision trees are good at handling categorical data and missing values, but it does not perform well if there is a lot of noise in the data. Since we have cleaned our data and all of the attributes are categorical, the decision tree algorithm is an excellent option for building our model. In our benchmark decision tree model, we get a value of 0.521 for class 1 recall, 0.586 for class 1 F1-score, and 0.798 for class 1 area under ROC curve.

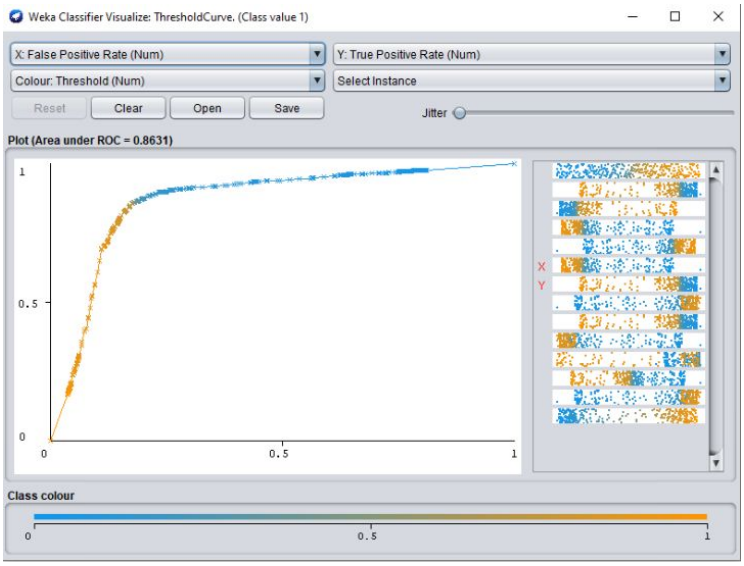
```
Correctly Classified Instances       22043               84.3138 %
Incorrectly Classified Instances      4101               15.6862 %
Kappa statistic                          0.4914
Mean absolute error                      0.2171
Root mean squared error                  0.3552
Relative absolute error                 64.7126 %
Root relative squared error             86.7103 %
Total Number of Instances            26144

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.930    0.479    0.878      0.930    0.903      0.497    0.798     0.896     0
                 0.521    0.070    0.670      0.521    0.586      0.497    0.798     0.546     1
Weighted Avg.    0.843    0.392    0.833      0.843    0.836      0.497    0.798     0.821

=== Confusion Matrix ===

     a     b   <-- classified as
 19137  1432 |     a = 0
  2669  2906 |     b = 1
```

The table below displays the evaluation measures for each version of our decision trees.

Model	Accuracy	Weighted Avg. Precision	Weighted Avg. Recall	Weighted Avg. F1-Score	Weighted Avg. ROC_AUC	Class 1 Precision	Class 1 Recall	Class 1 F1-Score	Class 1 ROC_AUC
DT_Benchmark	84.31%	0.833	0.843	0.836	0.798	0.670	0.521	0.586	0.798
DT_FS	85.01%	0.841	0.85	0.843	0.839	0.693	0.533	0.602	0.839
DT_OS	83.83%	0.839	0.838	0.838	0.863	0.827	0.855	0.841	0.863
DT_OS_FS	83.91%	0.840	0.839	0.839	0.860	0.830	0.825	0.861	0.860

After running forward feature selection on our model, the class 1 recall barely improved to 0.533 and we see slight improvements in class 1 F1-score and AUC as well. In the third version of our decision tree model, we applied oversampling technique and we see a much greater improvement in all measures across the chart with class 1 recall at 0.855, F1-score at

0.841 and a AUC value of 0.863. The last model was built using both forward feature selection and oversampling methods. Its performance is quite similar to that of the model with just oversampling applied. It appears that the forward feature selection method for our decision tree models produces very little effects whether it's applied on the benchmark model or oversampled model. Although the DT_OS and DT_OS_FS have similar performance, we concluded that the best performing model is the third one with only oversampling since it has a slightly higher class 1 recall rate. The ROC curve for DT_OS is shown below.



Random Forest

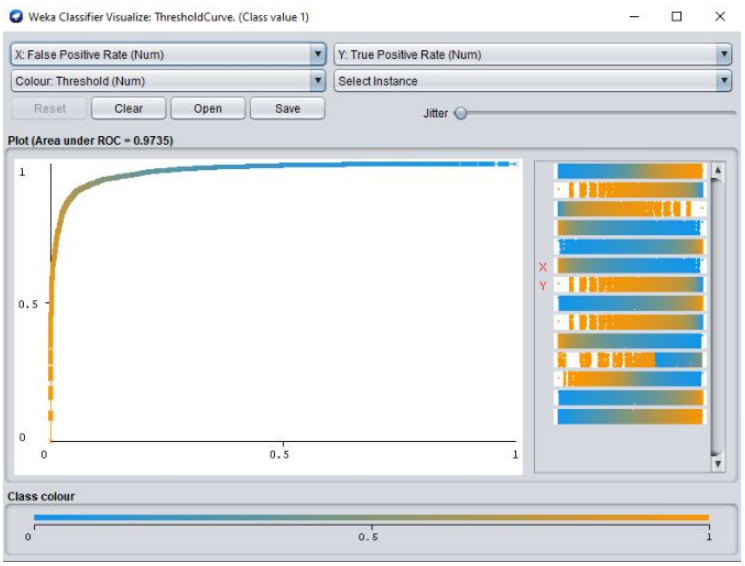
The Random Forest classifier is an algorithm that consists of multiple decision trees, hence the name. Based on the performance of our decision tree model, which was pretty good, we expect the random forest classifier to perform just as well if not better. Random forests are often difficult to visualize and interpret due to the nature of the algorithm. As expected, our benchmark random forest model performed similarly to our decision tree model.

```
Correctly Classified Instances       22141               84.6886 %
Incorrectly Classified Instances      4003               15.3114 %
Kappa statistic                          0.4743
Mean absolute error                      0.2327
Root mean squared error                  0.3316
Relative absolute error                 69.3412 %
Root relative squared error             80.9581 %
Total Number of Instances            26144

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.952    0.540    0.867      0.952    0.907      0.492    0.868     0.955     0
               0.460    0.048    0.721      0.460    0.562      0.492    0.868     0.665     1
Weighted Avg.  0.847    0.435    0.836      0.847    0.834      0.492    0.868     0.893

=== Confusion Matrix ===

     a     b   <-- classified as
 19577   992 |     a = 0
  3011  2564 |     b = 1
```

The evaluation measures for each version of random forest model are listed below.

Model	Accuracy	Weighted Avg. Precision	Weighted Avg. Recall	Weighted Avg. F1-Score	Weighted Avg. ROC_AUC	Class 1 Precision	Class 1 Recall	Class 1 F1-Score	Class 1 ROC_AUC
RF_Benchmark	84.69%	0.836	0.847	0.834	0.868	0.721	0.460	0.562	0.868
RF_FS	84.78%	0.840	0.848	0.842	0.879	0.671	0.554	0.607	0.637
RF_OS	91.23%	0.913	0.912	0.912	0.974	0.893	0.937	0.914	0.974
RF_OS_FS	91.48%	0.916	0.915	0.915	0.974	0.896	0.939	0.917	0.974

Our second random forest model with forward feature selection did not yield any significant improvement over the benchmark model. However, the third and fourth model, which utilized oversampling and a combination of oversampling and forward selection, produced similar and much better performance compared to the benchmark. Though it is exciting to see such high performance across all evaluation measures, many of them above 0.9, we also have to be cautious of a possible overfitting issue. For now, we have concluded that the RF_OS_FS is our best performing random forest model and the ROC curve is shown below.



Overfitting

In this section, we revisit our concern of an overfitting problem with our random forest model. We investigate this by rerunning our model again and instead supply the model with a test set. If the accuracy of the model with no test set supplied is higher than the accuracy of the model with test set, then we have evidence of overfitting. To do this, we split our data into

training and testing sets using an 80-20 split while ensuring that the proportion of yes/no responses in our target variable is the same. We then re-ran our best performing version of our random forest model. To be extra cautious and to have a fair comparison, we also applied the same procedure to our best performing Naive Bayes and decision tree models and our final evaluation of models is discussed in the next section.

Model Evaluation

Only Training Set

Model	Accuracy	Weighted Avg. Precision	Weighted Avg. Recall	Weighted Avg. F1-Score	Weighted Avg. ROC_AUC	Class 1 Precision	Class 1 Recall	Class 1 F1-Score	Class 1 ROC_AUC
NB_OS_FS	79.25%	0.793	0.793	0.792	0.865	0.800	0.780	0.790	0.865
DT_OS	85.00%	0.851	0.850	0.850	0.870	0.836	0.871	0.853	0.870
RF_OS_FS	91.74%	0.918	0.917	0.917	0.974	0.903	0.935	0.919	0.974

Test Set Supplied

Model	Accuracy	Weighted Avg. Precision	Weighted Avg. Recall	Weighted Avg. F1-Score	Weighted Avg. ROC_AUC	Class 1 Precision	Class 1 Recall	Class 1 F1-Score	Class 1 ROC_AUC
NB_OS_FS	79.56%	0.840	0.796	0.808	0.868	0.519	0.777	0.612	0.868
DT_OS	77.70%	0.825	0.777	0.791	0.758	0.491	0.747	0.592	0.758
RF_OS_FS	81.20%	0.835	0.812	0.820	0.864	0.552	0.713	0.622	0.864

Based on the tables above, we concluded that there was indeed an overfitting issue with the random forest model since the accuracy of the model built only with the training set is much higher than the model where a test set was supplied. There is also an 8% difference between the accuracy of decision tree models, so there might be a slight issue with overfitting as well. Overall, the Naive Bayes model is relatively stable and it also yielded the highest class 1 recall rate.

Conclusion

Interpretation & Recommendation

When building our models, we were focused on minimizing the number of false negatives which as a result would lead to a higher recall rate. A false negative in our case is when we predict a respondent would not receive an H1N1 vaccine, when in reality they will receive the vaccine. For the purpose of this study, we assumed that the cost of a false negative is the price of the vaccine, which is around $25[1]. By multiplying $25 and the number of false negatives produced by the model, we can get a sense of loss in revenue that pharmaceutical companies or healthcare providers could have made. Since the Naive Bayes model has the highest recall rate, the model should be best at minimizing the potential loss in revenue.

The summary output of the Naive Bayes model is shown below. From the confusion matrix, we take the number of instances classified as positives and divide it by the total number of instances in our dataset to get the proportion of respondents receiving a vaccine which is around 32.5%. Since the survey was conducted at random, it is fair to generalize this percentage to the U.S. population. From this, we estimate that the United States will need roughly 107.1 million doses of H1N1 vaccine, which is an estimation that would yield the lowest potential loss in revenue from a business perspective.

```
=== Summary ===

Correctly Classified Instances        4160               79.5563 %
Incorrectly Classified Instances      1069               20.4437 %
Kappa statistic                          0.4897
Mean absolute error                      0.2708
Root mean squared error                  0.3869
Relative absolute error                 54.1507 %
Root relative squared error             77.3722 %
Total Number of Instances             5229

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.801    0.223    0.928      0.801    0.860      0.508    0.868     0.955     0
                 0.777    0.199    0.519      0.777    0.622      0.508    0.868     0.658     1
Weighted Avg.    0.796    0.218    0.840      0.796    0.808      0.508    0.868     0.891

=== Confusion Matrix ===

    a    b   <-- classified as
 3279  816 |    a = 0
  253  881 |    b = 1
```

[1] According to this article: https://abcnews.go.com/Business/big-business-swine-flu/story?id=8820642

However, after lengthy discussions our group has come to the realization that a high precision rate is preferred as well. A high precision rate is attained by minimizing the number of false positives. A false positive in this case is predicting that a respondent would opt in for an H1N1 vaccine when in fact they will not receive the vaccine. This means producing a vaccine that would not be used. The cost of a false positive then is the manufacturing cost, which is roughly $8[2]. Both precision and recall rate have an effect on the total cost, therefore if reducing total cost is the main objective, then we recommend building additional models using the F1-score as the primary evaluation measure.

Lessons Learned & Takeaways

This project was an extremely valuable learning experience and a first introduction to machine learning for many of us. Collectively, we applied our knowledge from the course and properly handled missing values in our dataset, even using one of the algorithms we learned to do so. The class imbalance within our target variable also served as a great practice for oversampling techniques that was covered in this course. Moreover, we were able to rationalize which of the introduced model evaluation measures to use based on the problem that we are trying to solve. We were also able to identify an overfitting issue that was present in one of our models. Our goal for the study was to predict the demand for vaccines and possibly replicate our model for similar situations in the future. However, it is unclear at this point if our model could be generalized for COVID-19 or future pandemics.

[2] We divided $2 billion by 250 million doses to get an estimate of cost per dose