2015-04-30

Predicting Patient Satisfaction With Ensemble Methods

Elisa Renee Rosales
Worcester Polytechnic Institute

Predicting Patient Satisfaction With Ensemble Methods

by
Elisa Rosales

A Capstone Project Report
Submitted to the Faculty
of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the
Degree of Master of Science
in
Applied Statistics
by

Date: April 30, 2015

APPROVED:

Prof. Randy Paffenroth, Project Advisor

Prof. Luca Capogna, Dept. Head

Abstract

Health plans are constantly seeking ways to assess and improve the quality of patient experience in various ambulatory and institutional settings. Standardized surveys are a common tool used to gather data about patient experience, and a useful measurement taken from these surveys is known as the Net Promoter Score (NPS). This score represents the extent to which a patient would, or would not, recommend his or her physician on a scale from 0 to 10, where 0 corresponds to "Extremely unlikely" and 10 to "Extremely likely". A large national health plan utilized automated calls to distribute such a survey to its members and was interested in understanding what factors contributed to a patient's satisfaction. Additionally, they were interested in whether or not NPS could be predicted using responses from other questions on the survey, along with demographic data. When the distribution of various predictors was compared between the less satisfied and highly satisfied members, there was significant overlap, indicating that not even the Bayes Classifier could successfully differentiate between these members. Moreover, the highly imbalanced proportion of NPS responses resulted in initial poor prediction accuracy. Thus, due to the non-linear structure of the data, and high number of categorical predictors, we have leveraged flexible methods, such as decision trees, bagging, and random forests, for modeling and prediction. We further altered the prediction step in the random forest algorithm in order to account for the imbalanced structure of the data.

Acknowledgements

First and foremost, I would like to thank my advisor Prof. Randy Paffenroth for being a supportive and encouraging mentor who challenged and enabled me to think creatively throughout the duration of this project. I would also like to thank Prof. Joseph Petruccelli for his patience and understanding throughout my capstone project journey. Many thanks to Prof. Zheyang Wu and Prof. Balgobin Nandram for providing me with the necessary tools needed for statistical analysis.

I would like to also thank Silverlink Communications, Inc. for providing me with the data and for allowing me to put my career on hold as I finished this project.

I am especially grateful to my mentors Prof. Suzanne Weekes and Prof. Marcel Blais for believing in me even when I didn't believe in myself. I would not be where I am today without their support.

Lastly, I would like to thank my family for supporting my decision to move to Massachusetts by myself to pursue my academic dreams. Kristian, a special thanks to you for always being there for me throughout this experience.

Contents

Chapter 1

Introduction

In 2014, a large national health plan distributed a patient satisfaction survey via an automated call to over 600,000 Medicare members who had recently visited their primary care physician. The health plan wanted to gather this information in order to assess and improve the quality of patient experience, specifically the patient's experience between his or her physician. The measurement of interest from this survey was the Net Promoter Score (NPS), which measures the extent to which a patient would, or would not, recommend his or her physician on a scale from 0 to 10, where 0 corresponds to "Extremely unlikely" and 10 to "Extremely likely". With nearly 200,000 completed surveys, the health plan sought to understand the following:

1. **What distinguishes populations who score a low NPS (0 or 1 responses) from a high NPS (9 or 10 reponses)?**
   Understanding what factors differentiate less satisfied members (low NPS) from very satisfied members (high NPS) may help identify any health care disparities present in the population. For instance, both socioeconomic status and race/ethnicity have been shown to be significant predictors of both the perceived and actual quality of health care received [1]. If we can provide evidence that these disparities exist, then we can help the health plan justify allocating their budget toward programs, such as cross-cultural education, to improve provider-patient communication.

2. **Which survey questions are most predictive of NPS?**
   This survey contains several questions similar to those found on the CAHPS (Consumer Assessment of Health Care Providers and Systems) survey, which is distributed annually to a random subset of the health plan's population [2]. These results are used by the Centers for Medicare and Medicaid Services (CMS) to help determine payment incentives that reward high-performing health care providers [2]. If we treat the NPS measured from our patient satisfaction survey as a summary metric of the CAHPS survey, then understanding the relationship between the patient's responses to other questions and the NPS may help identify areas of improvement. For instance, if patients who respond "no" to, "Did your doctor review all of your prescription medicines with you?" are more likely to report a low NPS, then we could recommend requiring all physicians review their Medicare patients' medications.

3. **How well can we predict NPS using only the responses to the survey and demographic factors?**
   Being able to predict whether a patient or segment of the population is at risk of receiving poorer quality of health care can be the first step in controlling disparities within the health plan. An extensive study on the disparities within a large national account such as this will be costly and time-consuming. However, leveraging NPS as a proxy for a patient's perceived quality of health care, we can build a model using the survey responses collected from the automated call, along with demographic data collected from outside resources, to study population dynamics in a more efficient and cost-effective way.

With only around one-third of the surveyed population completing this survey, the results of our analysis will not be representative of the health plan's entire Medicare population. This is due to the selection bias inherent in the data as we only consider the members who responded to all of the survey questions. Instead, our results will be representative of patients who are most likely to engage in an automated outbound call outreach from their health plan. This is a limitation of the data to which we have access. However, even this limited, and perhaps biased sample, provides insights in to patient satisfaction.

The nonlinear relationship between the predictors and the response variable, along with the large number of predictors, many of which were categorical and indicator variables, led us to choose highly flexible, nonparametric models. Specifically, we consider decision trees, which split the predictor space into a number of rectangular regions and fit a simple model, such as a constant, in each region [5]. We found that single decision trees did not perform well based uponn cross-validation experiments, so our key results revolve around *ensemble methods*. The bootstrap aggregation (bagging) and random forest ensembles improved the performance of the single tree methods by essentially taking a number of "dumb" classification trees and making one "smart" learner out of them [6].

In the next chapter of this report, we will define the predictors and discuss their relevance to the model. In Chapter 3, we explore the data and discuss why the overlap in the class distributions cause the Bayes Classifier to perform poorly. In Chapter 4 we will discuss the single tree methodologies used to model and predict NPS in great detail and will identify appropriate measures of performance. In Chapter 5, we will review the methodologies of the ensemble methods and will explain how the tuning parameters of the models were chosen. Finally, in Chapter 6 we will determine which model is most appropriate and why. We will also summarize our findings for each of the initial questions of interest. The appendix contains a summary of the balanced random foret method, which we hope to implement in our future work. Additionally, the appendix contains preliminary modeling of this data where we used a multi-class response.

Chapter 2

Data

In order to understand what distinguishes patients with low versus high satisfaction, we first needed to gather relevant data on the population, along with each patient's responses to the survey questions. The health-care communications company that designed and implemented this automated outbound survey collected each member's responses in a file called a SIRF. The SIRF also contained the patient's gender, date of birth, and the physician's contact information. These variables were the minimum data requirements needed from the health plan to implement the call.

However, the more important demographic information regarding the income and ethnicity of the patient was not available. Since we hypothesized that these variables would likely drive patient satisfaction, it was important to find adequate proxies. We used outside sources, such as 2010 U.S. Census data, to indirectly measure socioeconomic status and ethnicity. Although these variables were indirect measurements, they proved to be significant predictors of a patient's NPS as we will show in the results section. Further, one of the key aspects of our results, and a reason we choose trees, was to enable the fusion of these disparate data sets.

Survey Question Variables

This satisfaction survey consists of various questions aimed at understanding the patient's experience with his or her physician. As previously mentioned, the Centers for Medicare and Medicaid Services (CMS) use the responses from a survey (very similar to this satisfaction survey) to help determine payment incentives for high-performing health care providers. Thus, the survey results will provide the health plan with a high-level overview of their Medicare population's overall patient experience. Below are the paraphrased questions of the survey that patients responded to on the automated call. All but questions 2 and 12 were answered by "Yes", "No" or "Doesn't Apply". The responses for questions 2 and 12 are listed below.

- **Question 1** (B1) "Did you have any problem setting up your appointment?"
- **Question 2** (B2) "How long did you have to wait to see your doctor after your scheduled time?" *Responses: Doesn't apply, 0-15 minutes, 15-30 minutes, 30-60 minutes, 60+ minutes.*
- **Question 3** (B3) "Did you have any trouble getting a referral?"

- **Question 4** (B4) "Did you have trouble with your insurance covering any medicines your doctor prescribed?"
- **Question 5** (B5) "Did you have any problems getting approval for any tests or procedures your doctor said you needed?"
- **Question 6** (B6) "During your visit, did your doctor review all of your prescription medicines with you?"
- **Question 7** (B7) "Did your doctor seem to know about your visits with specialists?"
- **Question 8** (B8) "Do you have problems with your balance or walking, or have you recently had a fall?"
- **Question 9** (B9) "Did your doctor talk with you about how to prevent falls?"
- **Question 9a** (B9a) "Did your doctor suggest any treatments, such as using a cane or walker, having your blood pressure checked, or having regular vision or hearing tests?" *Note: Only patients responding "Yes" to survey question 9 were asked this follow-up question*
- **Question 10** (B10) "Did your doctor talk with you about bladder control?"
- **Question 10a** (B10a) "Did your doctor suggest any treatments, such as bladder training, exercises, prescription medicine or surgery?" *Note: Only patients responding "Yes" to survey question 9 were asked this follow-up question.*
- **Question 11** (B11) "Did your doctor advise you to start, increase, or maintain your level of exercise or physical activity?"
- **Question 12** (B12) "Using a scale from 0 to 10, where 0 means not at all likely and 10 means extremely likely, how likely would you be to recommend your PCP to a friend or family member?" *Responses: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10*

Demographic Variables from the SIRF

Although informative as to the overall satisfaction of the Medicare population, the survey responses alone do not address disparities that may exist between gender or regions. Therefore, the variables gender, age, language, and region, which were taken directly from the SIRF, were included as predictors.

- **Gender** (MBR_GDR): Patient's gender, either male or female
- **Age** (MBR_AGE): Patient's age at the time of the survey and is calculated based on date of birth and date that the survey was completed
- **Language** (MBR_LANGUAGE): Patient's language, either English or Spanish, and is taken directly from the SIRF

- **Midatlantic** (MIDATLANTIC): Indicates whether a patient lives in New York, New Jersey, Pennsylvania, Delaware, Maryland, Washington D.C., Virginia, or West Virginia
- **Midwest** (MIDWEST): Indicates whether a patient lives in Illinois, Indiana, Iowa, Kansas, Michigan, Minnesota, Missouri, Nebraska, North Dakota, Ohio, South Dakota, or Wisconsin
- **New England** (NEWENGLAND): Indicates whether a patient lives in Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, or Vermont
- **South** (SOUTH): Indicates whether a patient lives in Florida, Georgia, Maryland, North Carolina, South Carolina, Alabama, Kentucky, Mississippi, Tennessee, Arkansas, or Louisiana
- **South or Midwest** (SOUTH_MW): Indicates whether a patient lives in the southern region, midwestern region, or other.
- **Southwest** (SOUTHWEST): Indicates whether a patient lives in Texas, Oklahoma, New Mexico, or Arizona
- **West** (WEST): Indicates whether a patient lives in Alaska, Washington, Oregon, Hawaii, California, Nevada, Idaho, Montana, Wyoming, Utah, or Colorado
- **Puerto Rico/Virgin Islands** (PR_VI): Indicates whether a patient lives in Puerto Rico or the Virgin Islands

Demographic Variables from Outside Resources

As we mentioned in the introduction, socioeconomic status and race/ethnicity have been shown to be significant predictors of the quality of health care received. Although we were not able to directly measure each patient's income or race, we used a variety of proxy variables from outside sources as indirect measurements of income and race. As we will demonstrate in the results section, these predictors were, perhaps somewhat surprisingly, more predictive of NPS than the responses from the survey questions.

- **Food environment index** (FOOD_ENVIRONMENT_INDEX): This variable scores every county in the U.S. from 0 to 10 with 10 being the best food environment based on variables from the United States Department of Agriculture such as store and restaurant availability, restaurant expenditures, local food assistance programs, and number of local farms, and variables from Feeding America, a hunger relief charity, concerning food insecurity [4]. Each individual is linked to a food environment index based on his or her zip code as specified in the SIRF. This variable was included to indirectly measure a patient's socioeconomic status since typically lower income populations tend to have less access to high quality food resources.
- **Racial Diversity Index** (DIVERSITY_INDEX): This index was created by recent research interns at a health care communications company [4]. Each patient in the data set was linked

to a diversity index based on his or her zip code. This variable is used as a proxy for a patient's race and ethnicity.

- **Income Ratio** (MED_ZIP_STATE_INCOME): This is another measure created by research interns at a health-care communications company [4]. They first measured median income on both a ZIP code and state level based on 2010 Census data. Due to variation in cost of living, the median income by itself was not particularly meaningful, so they took the ratio of the ZIP code median income to the state median income to create a measure of relative wealth. Each member in the population was given an income ratio based on their zip code. This variable also serves as a proxy for the patient's income and socioeconomic status.

- **Number of Primary Care Providers** (NUMERIC_VALUE): This counts the number of primary care providers per 100,000 people and was measured based on 2010 U.S. Census data. This variable may also indicate a patient's socioeconomic status since many low-income communities may lack resources, such as physicians. [4]

Provider and Contract Variables

There were several variables available in the SIRF pertaining to the patient's provider and the patient's insurance policy. These variables were included in an attempt to improve prediction accuracy of NPS.

- **Contract Size** (CONTRACT_SIZE): We hypothesized that patients who have a widely used insurance policy will experience less difficulty in scheduling appointments or understanding coverage than members belonging to less common policies. This variable was calculated using each patient's insurance policy (contract) number. There were 66 distinct policies in the population, and in general, when a predictor contains q possible unordered values, there are $2^{q-1}-1$ possible binary partitions. This makes methods, such as decision trees, very computationally expensive. Therefore, we created a variable that determines whether a patient belongs to a large, medium, or small contract. Insurance policies with over 70,000 members in the population were labeled as "large" contracts. Policies with 10,000 to 70,000 members were labeled as "medium", and policies with fewer than 10,000 members were labeled as "small".

- **Provider Size** (PROVIDER_SIZE): We also hypothesized that patients whose physicians are responsible for a very large number of patients may have a less personal experience with his or her physician, leading to less patient satisfaction. This variable was calculated using each patient's provider's tax ID. There were 20,843 distinct providers in the population, therefore, I created a variable that determines whether a patient belonged to a large, medium, or small "provider size". Providers with over 10,000 patients in the population were labeled as large providers. Providers with 3,000 to 10,000 patients were labeled as medium, and providers with fewer than 3,000 were labeled as small.

Automated Call Specifications Variables

The remaining variables were used to understand whether such factors as the scripting of the call or the time between doctor visit and survey completion impacted NPS. If these did affect NPS, then the health care communications company could alter their call policy and scripts to ensure that NPS is not negatively impacted by controllable aspects of the call.

- **SIRF Type** (SIRF_ID): Indicates whether the patient received version 1 or version 2 of the scripted call.
- **Time to Survey** (TIME_TO_SURVEY): Measures the number of days between the doctor visit and time of automated call.

Chapter 3

Data Exploration

3.1 Bayes Classifier

To begin our analysis, we consider the classic Bayes Classifier. The basic idea behind this naive classifier is to determine the most likely class, given its predictor values [6]. The Bayes Classifier is optimal, but not a practical algorithm since the requisite probabilities are not known, in general [6]. In other words, given a predictor vector x_0, this classifier assigns Y to class j for which $Pr(Y = j|X = x_0)$ is largest. For instance, when there are two classes, the Bayes Classifier corresponds to predicting class one if $Pr(Y = 1|X = x_0) > 0.5$ and class two otherwise [6, p. 37].

In general, the overall Bayes error rate is given by

$$1 - E\left(max_j Pr(Y = j|X)\right)$$

When classes are perfectly separable, the Bayes error rate is 0, making accurate class predictions effective. A non-zero Bayes error rate is introduced when the classes overlap, meaning that the accuracy of class predictions will be limited.

In general, the error of a model can be decomposed into reducible and irreducible error [6].

$$\begin{aligned} E(Y - \hat{Y}) &= E\left[f(X) + \varepsilon - \hat{f}(X)\right]^2 \\ &= \left[f(x) - f(\hat{X})\right]^2 + \text{Var}(\varepsilon) \end{aligned}$$

Reducible error can be minimized by selecting an appropriate model for the data. However, irreducible error serves as an upper bound on the accuracy of predictions. Here, the Bayes error rate represents the irreducible error of the Bayes Classifier. The higher the overlap between classes, the higher the irreducible error and the more difficult the classification problem becomes [6].

To get an idea of the performance of a theoretical Bayes Classifier on this data set, we consider the histograms of the two most important features as specified by the random forest algorithm (see results section). In Figure 3.1,the blue histograms represent the distribution of the patients who responded 9 or 10 to the NPS question, while the green histograms summarize the distribution of patients who responded 0 or 1. The left plot contains the distribution of the Food Environment Index, while the right plot contains the distribution of the Med_Zip_to_State variable. In both plots,

we only consider patients who were surveyed within 60 days of their doctor visit (TimetoSurvey ≤ 60). Unfortunately, it is clear that not even the Bayes Classifier will be able to succeed in finding an adequate decision boundary for this data as the distributions completely overlap. The high Bayes error rate, or irreducible error, will lead to very poor prediction accuracy. This illustrates how difficult classifying these patients will be.

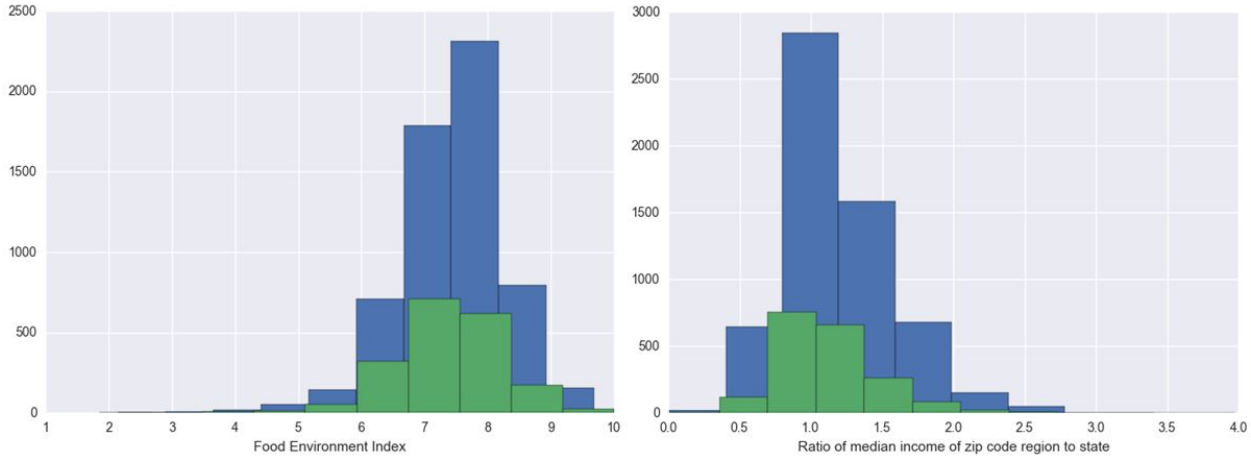

Figure 3.1: *The left and right plots contain the histograms of the Food Environment Index and the Med_Zip_to_State variable, respectively. We only consider the subset of patients who were surveyed within 60 days of their doctor visit (TimetoSurvey* ≤ 60*). The green bars represent the patients who responded 0 or 1 to the NPS question while the blue represent those who responded 9 or 10. Notice that the distributions of both classes completely overlap, indicating that even a theoretical Bayes Classifier would perform poorly with this dataset.*

3.2 Principle Component Analysis

We use Principle Component Analysis to visualize a low dimensional representation of the data that preserves the most variance [6]. If an observable separation of the classes exist in this space, then we can build our models using the principle components, rather than the raw data. Additionally, we plot the singular values of the data matrix to determine whether data reduction of the data is effective.

In the upper-left corner of Figure 3.2 is a plot of the singular values calculated on 1,000 random samples of the training data. All but three predictors (*Contract_Size*, *Provider_Size*, and *South_MW*) were included at this point in the analysis. Notice how the singular values slowly approach 0. This indicates that dimension reduction of the data will not be effective. Plotted in the upper right corner is first principal component versus the second, in the lower-left is the second component versus the third, and in the lower right is the first component versus the third. No identifiable clusters between the two classes were observed; therefore, we chose to build our models on the raw data, rather than on its principal components.

The variables *Contract_Size*, *Provider_Size*, and *South_MW* were later added as predictors to the model in an effort to increase prediction accuracy of the model. Figure 3.3 contains a plot of the singular values and principle components of the data set with these addtional predictors. Again, the singular values appear to slowly approach 0, indicating that dimension reduction of the data would not be effective. Plotted in the upper right corner is first principal component versus against the second, in the lower-left is the second component versus the third, and in the lower right is the first component versus the third. Although very interesting clusters formed, there was no identifiable separation between the two classes. Therefore, we chose to not build our models on the principal components of the data.

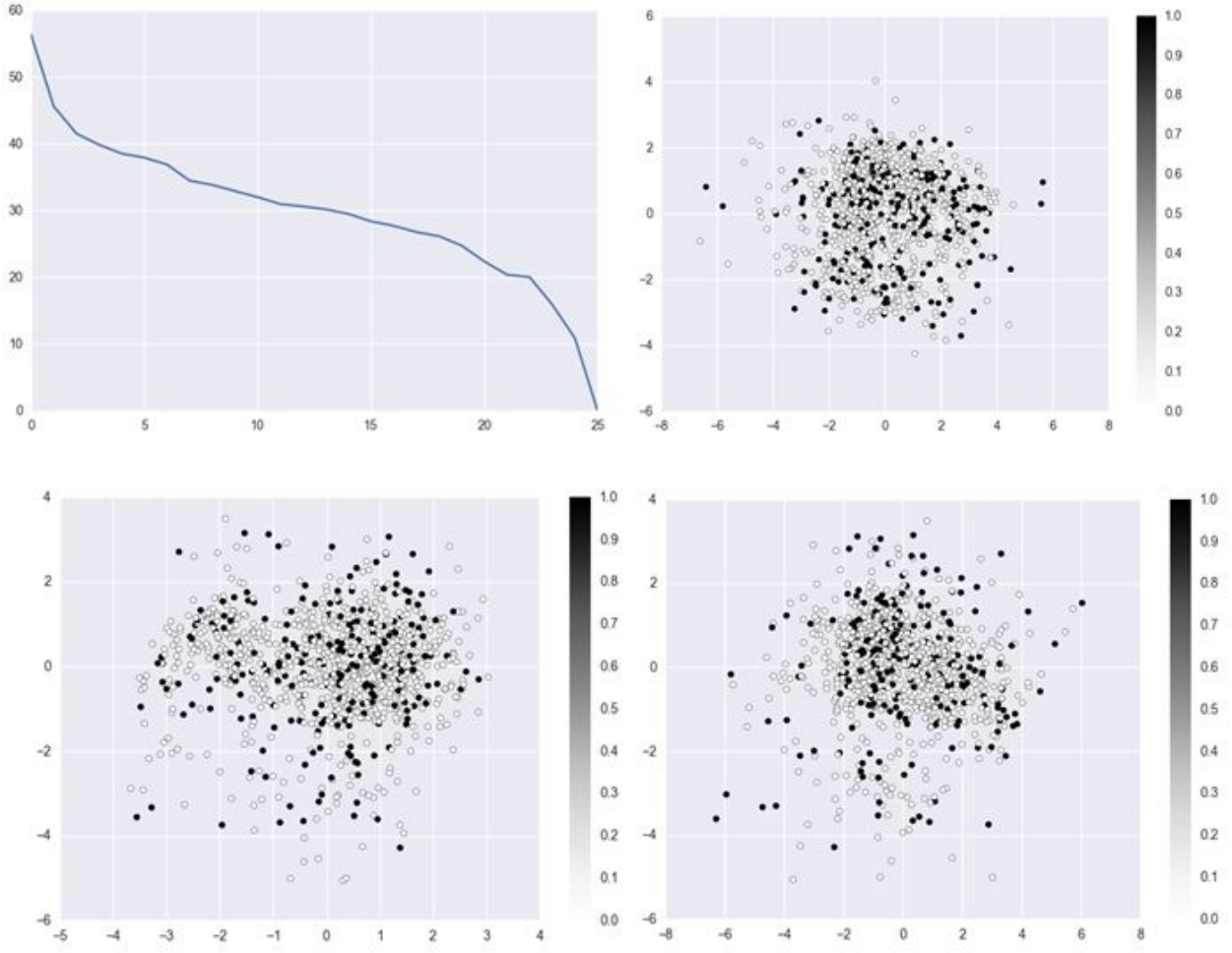

Figure 3.2: *In the upper-left corner is a plot of the singular values calculated on 1,000 random samples of the training data. All but three predictors (Contract_Size, Provider_Size, and South_MW) were included at this point in the analysis. Notice how the singular values slowly approach 0. This indicates that dimension reduction of the data will not be effective. Plotted in the upper right corner is first principal component versus the second, in the lower-left is the second component versus the third, and in the lower right is the first component versus the third. No identifiable clusters between the two classes were observed; therefore, we chose to build our models on the raw data, rather than on its principal components.*

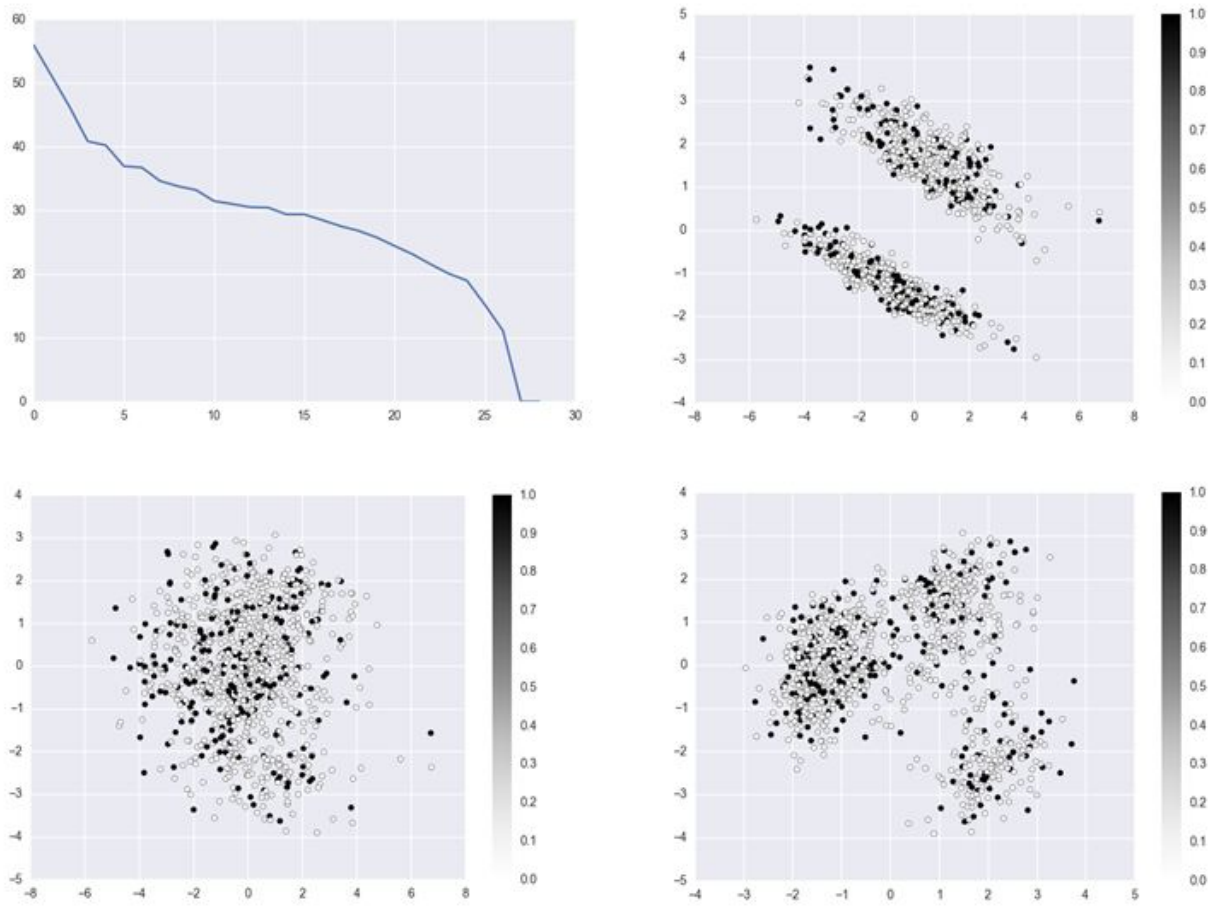

Figure 3.3: *In the upper-left corner is a plot of the singular values calculated on 1,000 random samples of the training data. The variables Contract_Size, Provider_Size, and South_MW were added as predictors to hopefully increase prediction accuracy of the model. Again, the singular values slowly approach 0, indicating that dimension reduction of the data would not be effective. Plotted in the upper right corner is first principal component versus against the second, in the lower-left is the second component versus the third, and in the lower right is the first component versus the third. Although very interesting clusters formed, there was no identifiable separation between the two classes. Therefore, we chose to not build our models on the principal components of the data.*

3.3 Correlation matrix

As another tool for preliminary analysis, we look at the correlation matrix of the data to understand which predictors are redundant, and which have the possibility of bringing new information. Correlation measures the linear relationship between two variables and is defined as,

$$\text{Corr}(X,Y) = \frac{\sum_{i=1}^{n}(x_i-\hat{x})(y_i-\hat{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\hat{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\hat{y})^2}}. \tag{3.1}$$

Visualizing the correlation matrix in Figure 3.4 allowed us to assess which predictors are correlated with each other or with the response. It does not appear that any of the predictors are significantly correlated with the response. However, the demographic and region data, located at positions 0-12, had measurable correlation. Additionally, the survey questions, located at positions 16-29, also appeared to be correlated. While linear models perform best when the predictors are uncorrelated, flexible models, such as decision trees, are not adversely affected by correlated predictors.

Note predictor variable PR_VI was not included in the construction of this correlation matrix. Since every member in included in this survey was from the United States, the PR_VI variable had 0 variance and so its correlation could not be calculated. Due to this variable being non-informative, it was removed from the predictor dataset.

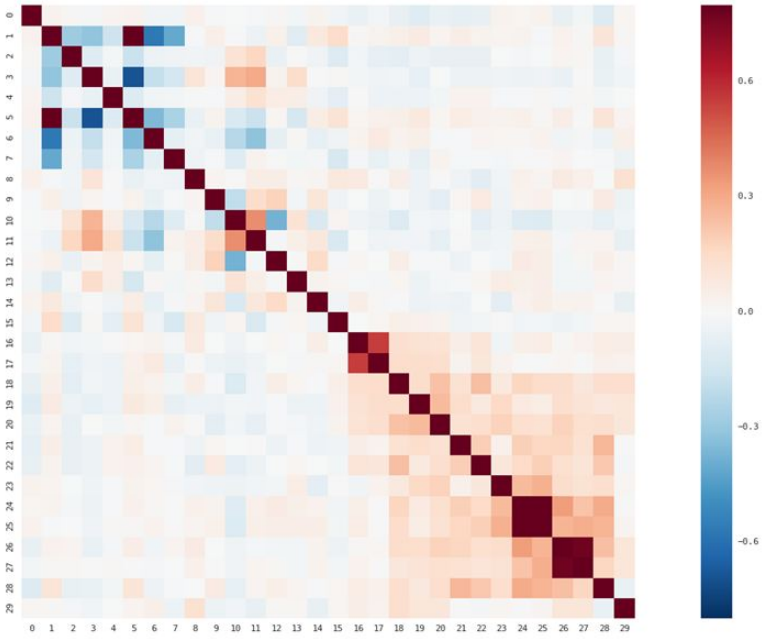

Figure 3.4: *The correlation between each of the predictors and the response was calculated and plotted in a correlation matrix where a correlation of -1 corresponded to dark blue, a correlation of +1 corresponded to dark red, and a 0 correlation was white. Lighter shades of blue and red indicate weaker correlation.*

Chapter 4

Methodology

4.1 Overview of Methods

Due to the large number of categorical variables introduced by the survey questions, along with the non-Gaussian structure of the NPS response variable, we sought to model this data using highly flexible and nonlinear methods.

We initially scaled the 0-10 NPS responses down to a 1-5 scale and used a validation set to compare the prediction accuracy of our models. In other words, the training data was randomly split into a training set and a validation set. The model was trained on a fraction of the data and its performance was evaluated on the validation set. We built single decision tree classifiers and regressors and found that the full trees highly overfit the training data. Therefore, they did not perform well on the validation set. Pruned trees yielded a better overall classification rate on the validation set, but failed to adequately model the NPS responses ranging from 2 - 4. Single decision trees are known to have high variance, so we also used bootstrap aggregation, or bagging, on the decision trees to reduce the variance of the model. The bagging method takes the average of B fully grown classification trees built on random samples (with replacement) of the data [6]. The average leads to a reduction of the variance of the trees. This method was better able to account for the NPS responses from 2 - 4, but the overall classification rate on the validation set was still quite low. Bagging tree methods often result in a set of correlated trees [6], so the random forest method was used to de-correlate the trees. Similar to the bagging method, random forests build a set of fully grown trees on random samples (with replacement) of the data. However, only a fraction of the predictors are considered at each split in the trees [6]. This method yielded a moderate increase in classification rate on the testing set, but there still remained significant room for improvement. Therefore, new predictors were introduced to the data and the bagging and random forest methods were rerun. These predictors did not significantly increase the classification rate of either method.

As the full classification problem is so difficult, we decided to also attempt to solve a simpler problem. In particular, we attempted to rescale the response variable, focusing only on the extreme responders to determine how well the methods could identify the extreme cases in the absence of the less extreme cases. The scores were scaled from 0 to 1, where 0 corresponded to the 0 and 1 NPS scores and 1 corresponded to the 9 and 10 NPS scores. The overall classification

rate of the bagging and random forest methods using the binary responses measurably increased; however, these models failed to adequately model the low NPS responders. Therefore, we sought a new performance measurement that would more accurately summarize the models' ability to account for "minority class" of low NPS reponders. We chose a performance measure that is based on the confusion matrix of the observed versus predicted response. The bagging and random forest methods with the binary responses yielded very poor results based on this new performance measure, despite having a high classification rate.

Recognizing the need to account for the imbalanced structure of the response data led us to alter the prediction scheme of the random forest algorithm. This method can return the prediction probabilities for a given input vector x0. The cutoff value refers to the rule at the prediction step of the algorithm, which states that if the probability that x0 belongs to class 1 is greater than 0.5, then it will be predicted as class 1 [7]. We consider different values of the cutoff to favor the minority class in order to reduce the number of false negatives. This method yielded favorable results relative to the other methods; however, there remains significant room for improvement.

4.2 Regression Trees

Regression trees are a highly flexible, nonlinear method that partition the feature space into a number of rectangular regions and fits a simple model (typically a constant) in each region [5, p. 305]. I utilize a common tree method, known as CART, which segments the feature space via recursive binary partitioning. This ensures that the regions are non-overlapping, which allows for easier interpretation of the partitions [5, p. 305]. When the target is to estimate the NPS score within a continuous range between 1 and 5, then the predictions from the regression trees will result in continuous responses, despite the training set responses being discrete.

Regression trees are constructed in a way that minimizes the error sum of squares within each partition [5, p. 307]. Suppose there are p predictors and N observations. Let x_i consist of all predictor values for each observation, and y_i consist of the corresponding response. The data takes the form:

$$x_i = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Np} \end{pmatrix}, y_i = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

Now, suppose there are M partitions $R_1, R_2, \ldots, R_M$ whose response is modeled by a constant c_m. Then, f takes the form:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \tag{4.1}$$

where $I(x \in R_m)$ is an indicator function of x that depends on whether or not x belongs to a partition R_m. The estimator $\hat{c}_m$ that minimizes the error sum of squares is the average response of region R_m:

$$\hat{c}_m = \text{ave}\,(y_i | x_i \in R_m) \tag{4.2}$$

Using top-down recursive binary partitioning, the algorithm begins at the top of the tree and then successively splits the predictor space. This algorithm is typically referred to as "greedy" because each node is split in the best way at that point in the tree-building process, and the algorithm does not look ahead and pick the best split that will produce a better tree [5, p. 307]. Now, splitting variable j at a split point s will result in two regions,

$$R_1(j,s) = \{X|X_j \leq s\} \text{ and } R_2(j,s) = \{X|X_j > s\} \tag{4.3}$$

and the choice of splitting point s and variable j is constructed by solving,

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right]. \tag{4.4}$$

The inner minimization is solved by

$$\hat{c}_1 = \text{ave}(y_i|x_i \in R_1(j,s)) \text{ and } \hat{c}_2 = \text{ave}(y_i|x_i \in R_2(j,s)) \tag{4.5}$$

This process is repeated within each newly split region until a stopping criterion is reached, such as having one or two observations at a given terminal node [6, p. 307]. Unfortunately, complex full tree methods tend to over-fit the data; however, cost-complexity pruning is a tool that can prevent over-fitting [6, p. 307]. Suppose subtree $T \subset T_0$ is obtained by pruning the full tree T_0 by collapsing any number of its internal nodes. Using m to index the terminal nodes, let $|T|$ denote the number of terminal nodes in T and let

$$\begin{aligned} N_m &= \#\{x_i \in R_m\}, \\ \hat{c}_m &= \frac{1}{N_m} \sum_{x_i \in R_m} y_i, \\ Q_m(T) &= \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2. \end{aligned}$$

Now, the cost complexity criterion is

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|. \tag{4.6}$$

For each α, we want to find the subtree $T_\alpha \subseteq T_0$ that minimizes $C_\alpha(T)$. For each α, there is a unique smallest subtree T_α that minimizes $C_\alpha(T)$. To find T_α, we successively collapse the internal node that produces the smallest per-node increase in $\sum_m N_m Q_m(T)$ and continue until we produce the single-node tree [5, p. 308]. Now, α serves as a tuning parameter and can be estimated using 5- or 10-fold cross-validation. The estimate $\hat{\alpha}$ is chosen to minimize the cross-validated sum of squares [5, p. 308].

We include this summary of the tree pruning process for completion; however, we used cross-validation to determine the optimal "tree depth". Figure 4.1 contains an example of a pruned regression tree of the NPS data scaled 1-5. This example illustrates how the predictions of a regression tree will be continuous, despite the input responses being discrete.

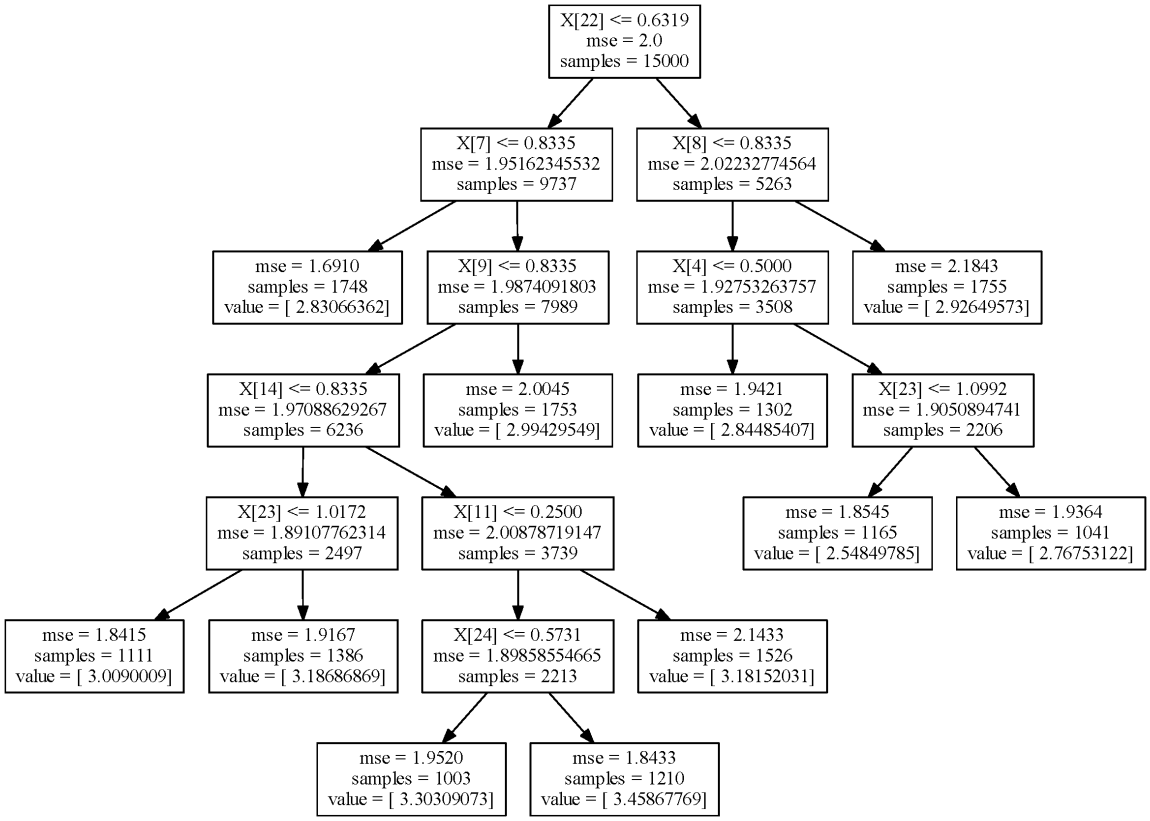

Figure 4.1: *This regression tree demonstrates how the predictions of a regression tree will be continuous, despite the input responses being discrete. This tree was built on a random sample of the training data with NPS responses scaled 1-5 and was pruned down to a maximum tree depth of 6. The "value" reported at the terminal nodes, or leafs, of the tree corresponds to the average NPS score at that node, while the "mse" measures the error within the node.*

4.3 Classification Trees

Classification trees differ from regression trees in two major ways. The first difference pertains to the criteria for splitting the nodes, and the second pertains to the pruning criteria. While regression trees split nodes based on the MSE, classification trees are focused on probabilities [5, p. 308]. In other words, observations in region m are classified as $k(m) = \text{argmax}_k \hat{p}_{mk}$, where

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k). \tag{4.7}$$

Node impurity is commonly measured using the following:

$$Q_m(T) = 1 - \hat{p}_{mk(m)} \quad \text{Misclassification Error} \tag{4.8}$$

$$Q_m(T) = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \quad \text{Gini Index} \tag{4.9}$$

$$Q_m(T) = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk} \quad \text{Cross-entropy or deviance} \tag{4.10}$$

Cross-entropy and the Gini index are more sensitive to class probabilities. Specifically, both Gini and cross-entropy take on smaller values when $\hat{p}_{mk}$ is closer to one or zero, and so these values can be used as a measure of node purity. Accordingly, Gini and cross-entropy are the generally preferred methods for tree building. The misclassification rate is typically used to prune trees when prediction accuracy is the goal of the pruned tree. [5, p. 309]

As an initial foray into classification, there are important parameters to set, such as tree depth [6]. Accordingly, we use 10-fold cross validation to construct trees with maximum tree depths ranging 1 to 39. Figure 4.2 displays the boxplots of the percentage of correct classifications, or classification scores, for each tree. It appears that a maximum depth of 3 yields a high classification score. Figure 4.3 is the resulting classification tree, whose nodes were split using the Gini criterion, on the training data with a maximum depth of 3.

Due to the imbalanced structure of the response variable, the overall classification accuracy turns out to be a very misleading measure of performance. The blue dotted line on Figure 4.2 represents the proportion of the members in the training set who belong to the majority class (NPS score = 9 or 10). A naive estimator that always predicts the majority class with this data will be correct 74% of the time. Therefore, a more appropriate measure of performance was needed as we discuss in the next section.

4.3.1 Performance Measurements for Imbalanced Data

In imbalanced data sets, the more "interesting" class is typically the minority class and, accordingly, is denoted as the 1 or "positive" class [3]. In this application, it is of great interest to understand what drives members to score their physicians with a low NPS so that we can ultimately prevent patients from having unsatisfactory experiences. However, we do not want to incorrectly identify large segments of the population as unsatisfied as this will reflect poorly on both the physicians and the health plan. Therefore the trade-off between correctly identifying unsatisfied members and controlling the misclassification rate of satisfied members must be quantified in order to adequately compare the performance across models.

We begin by relating the aforementioned trade-off problem to the confusion matrix in Table 4.1. We observed that several observations actually belonging to the positive minority class (low NPS) were being predicted as the negative majority class. In other words, our decision trees were resulting in a large number of False Negatives. We would like to see our classifiers give high prediction accuracy of the positive class (Acc^+), while maintaining a reasonable accuracy for the majority class (Acc^-). Weighted accuracy serves as a measurement for this trade-off. The β

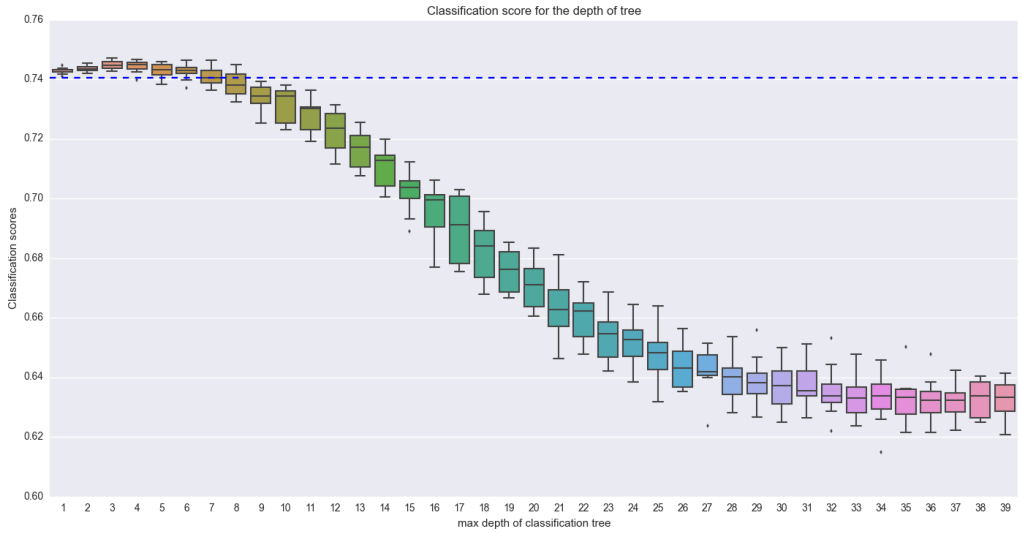

Figure 4.2: *We constructed classification trees with depths ranging from 1 to 39 and used 10-fold cross validation for each tree to generate 10 classification scores. Boxplots helped visualize the distribution of the 10 classification scores at each tree depth. The optimal depth resulting in the highest classification score appears to be 3. However, the blue dotted line represents the proportion of the patients in the training set who belong to the majority class (NPS score = 9 or 10). A naive estimator that predicts always predicts the majority class will be correct 74% of the time. A more appropriate performance measure is needed as we will discuss later in the text.*

parameter can range from 0 to 1 and represents the "importance" of the true negative rate relative to the true positive rate. For instance, if the misclassifying a true positive was more costly, then β would be higher for Acc^+ and lower for Acc^-. Since we seek to maintain a balance between these error rates, we will set β to 0.5. Accordingly, there are several metrics that one may consider [3], including:

$$\begin{aligned}
\text{True Negative Rate}(Acc^-) &= \frac{TN}{TN+FP} \\
\text{True Positive Rate}(Acc^+) &= \frac{TP}{TP+FN} \\
\text{Weighted Accuracy} &= \beta Acc^+ + (1-\beta)Acc^- \\
\text{Precision} &= \frac{TP}{TP+FP} \\
\text{Recall} &= \frac{TP}{TP+FN} \\
\text{F1 measure} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\end{aligned}$$

Precision measures the proportion of correct positive *predictions*, while Recall measures the proportion of positive *observations* that were correctly classified. The F measure is a measurement

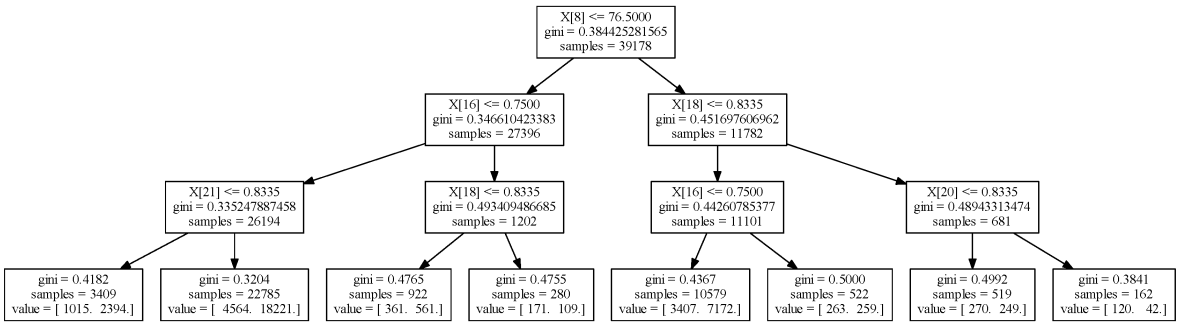

Figure 4.3: *This is the classification tree with a depth of 3, which was chosen by 10-fold cross validation with classification score as the performance measure. The number in the first position of the "value" array corresponds to the number of members in that terminal nod that are classified as the minority (NPS= 0 or 1) class. The purity measure of this tree is Gini, and none of the terminal nodes appear to have low purity.*

	Predicted Positive Class	Predicted Negative Class
Actual Positive class	TP (True Positive)	FN (False Negative)
Actual Negative class	FP (False Positive)	TN (True Negative

Table 4.1: *This is the confusion matrix of a binary response model. The rows represent the actual class that an observation belongs to, while the columns represent the class that the model predicted. Thus far, our models have resulted in a high number of False Negatives due to the class imbalance.*

used to describe the trade-off between precision and recall and ranges from 0 to 1, where 1 corresponds to a good recall-precision trade off [3].

In addition to numerical performance measurements, we consider the ROC curve which is a graphical representation of the trade-off between the false positive and false negative rate for every possible cutoff. Methods resulting in an Area Under the ROC curve (AUC) close to 1 correspond to high prediction accuracy, while methods with an AUC close to 0.50 correspond to poor prediction accuracy and are hardly better than a coin toss [3]. Accordingly, the primary performance metric we will focus on for the cross-validated sets will be the F-measure, while the secondary metric will be weighted accuracy. We chose these measures together because the F-measure will highlight the model's ability to control for false negatives, while the weighted accuracy summarizes the overall prediction accuracy of the model. The ROC curves will be used to assess model performance on the testing data set in the results section.

4.3.2 Tuning the Parameters of the Classification Tree

Below, the boxplots summarize the F1 measures resulting from performing 10-fold cross validation on maximum tree depths ranging from 2-40. It appears that the F1 measure begins to stabilize at 25 trees, thus, the maximum depth tuning parameter is chosen to be 25. The next tuning parameter of interest was the node splitting criterion. The Gini Index slightly outperformed cross-entropy

and was thus chosen as the final splitting criterion. Table 4.2 summarizes the results of each classification tree's performance. The weighted accuracy and F measures were also calculated using 10-fold cross validation. We included in this table classification trees with the optimal tuning parameter with respect to both classification accuracy and F-measure for comparison. The deeper trees, denoted as CT_E25 and CT_G25 in Table 4.2, did not yield high weighted accuracy values, and although their F-measure was measurably larger than that of the more pruned trees, denoted as CT_E3 and CT_G3, the $F1$ score fell on the low end of the scale. This implies that our model did not do an adequate job finding a trade-off between recall and precision.

As we have shown, single trees are inappropriate for predicting classes with this data. It is well known that single decision trees suffer from high variance. Therefore we turn to variance-reducing ensemble methods.

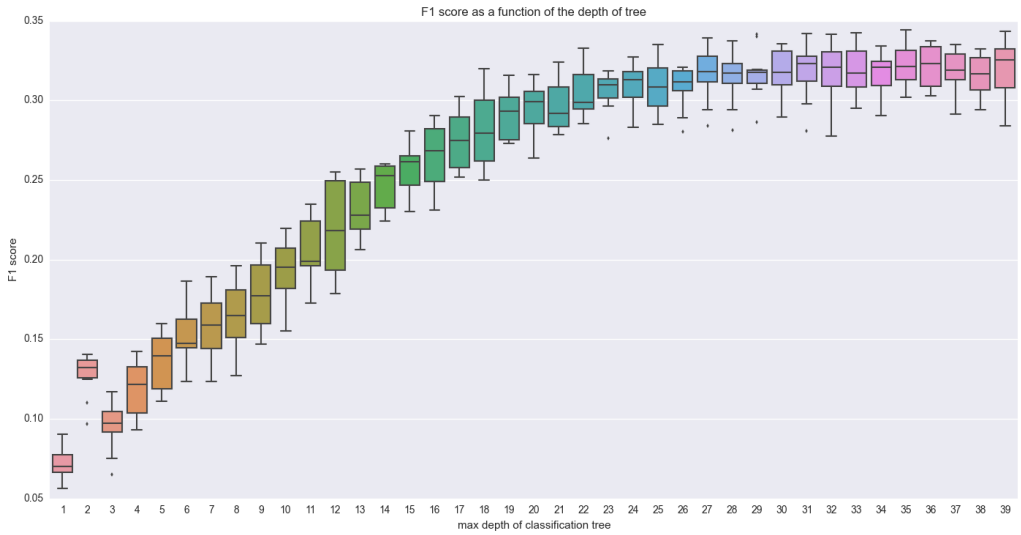

Figure 4.4: *We constructed classification trees with depths ranging from 1 to 39 and used 10-fold cross validation for each tree to generate 10 F-measures. Boxplots helped visualize the distribution of the 10 classification scores at each tree depth. The F-measure appears to stabilize around 25; therefore, a tree depth of 25 is chosen.*

Model Label	Criterion	Maximum Tree Depth	Weighted Accuracy	F-measure
CT_E25	Cross-entropy	25	0.66 (+/- 0.02)	0.30 (+/- 0.02)
CT_G25	Gini	25	0.65 (+/- 0.02)	0.31 (+/- 0.03)
CT_E3	Cross-entropy	3	0.74 (+/- 0.00)	0.12 (+/- 0.03)
CT_G3	Gini	3	0.74 (+/- 0.00)	0.13 (+/- 0.03)

Table 4.2: *We obtained the weighted accuracy and F-measures of the classification trees with depths chosen by 10-fold cross validation, based on performance measurements classification score and F-measure. The deeper trees yielder a higher F-measure but lower accuracy.*

Chapter 5

Ensemble Methods

One of the biggest disadvantages of single trees is their high variance [6]. Small changes to the data can result in a very different tree. The major reason for this instability is "the hierarchical nature of the process: the effect of an error in the top split is propagated down to all of the splits below it." [5, p. 312] By introducing randomness to the trees, we can reduce the overall variance. Ensemble methods achieve this reduction in variance by essentially taking many "dumb" learners and making one "smart" learner out of them [8]. Widely used ensemble methods for trees are bootstrap aggregation and the random forests.

5.1 Bagging

The bootstrap aggregation, or bagging, method consists of taking a training set of N samples and applying a "dumb" learning model, such as a classification tree, to a bootstrapped sample of the training data [6]. The final prediction of the bootstrapped model is either the average or majority vote of all B learners. Bagging methods work very well on models that have low bias but high variance, such as decision trees [5]. We will show how bagging uses averaging to effectively reduce the variance of a model.

Consider fitting a model on the training data $\mathbf{Z} = \{(x_1, y_1), (x_2, y_x), \ldots, (x_N, y_N)\}$. Now, suppose $\hat{f}(x)$ is the prediction at input x. For each bootstrap sample $\tilde{\mathbf{Z}}^b, b = 1, 2, \ldots, B$, a model is fitted, giving prediction $\hat{f}^{*b}(x)$ [5, p. 282]. The bagging estimate is defined by

$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b}(x). \tag{5.1}$$

Let $\hat{\mathscr{P}}$ denote the empirical distribution with each of the data points (x_i, y_i) having equal probability $\frac{1}{N}$. The "true" bagging estimate is defined by $E_{\hat{\mathscr{P}}} \hat{f}^*(x)$, where $\tilde{\mathbf{Z}} = \{(x_1^*, y_1^*), (x_2^*, y_2^*), \ldots, (x_N^*, y_N^*)\}$ and each $(x_i^*, y_i^*) \sim \hat{\mathscr{P}}$. The above expression is a Monte Carlo estimate of the true bagging estimate, approaching it as $B \to \infty$ [5, p. 282].

For regression, the same full grown regression tree is fit many times to bootstrap-sampled versions of the training data and the average result is the predicted value. For classification, a com-

mittee of trees each cast a vote for the predicted class or the class that maximizes the probability can be chosen.

5.1.1 Tuning the Parameters of the Bagging Method

The first tuning parameter considered was the number of trees to include in the ensemble. We applied bootstrap aggregation to classification trees and set the number of trees in the ensemble from 1 to 100. Using 10-fold cross validation, the F measure was calculated on each of the 10 validation sets. To better visualize the distribution of these F measures given the number of trees in the ensemble, we constructed the boxplots shown in Figure 5.1 and observed that the F1 scores appear to stabilize around 50 trees. Table 5.1 summarizes the mean F1 scores obtained from 10-fold cross validation along with a standard error of $2 \times std(mean)$. There was not a measurable difference in the weighted accuracy or F measure between the models split using cross-entropy and Gini.

Although both bagging methods resulted in an increase in weighted accuracy, the F1 measure actually dropped significantly. Recall that the overall proportion of the training data that belongs to the majority class is 74%. It appears that the bagging method is not adequately accounting for the minority class.

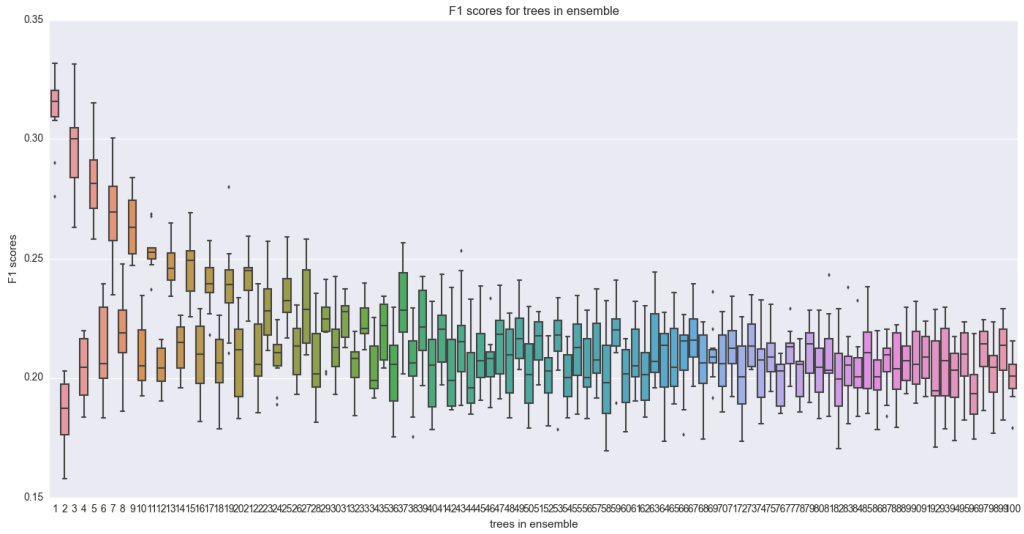

Figure 5.1: *We constructed bagging trees with the number of estimators ranging from 1 to 100 and used 10-fold cross validation for each ensemble to generate 10 F-measures. Boxplots were used to visualize the distribution of the 10 classification scores at each bagging tree. The F-measure appears to stabilize around 50 trees; therefore, the tuning parameter of the number of trees was chosen to be 50.*

Model Label	Criterion	Number of Estimators	Weighted Accuracy	F-measure
BA_E50	Cross-entropy	50	0.74 (+/- 0.00)	0.22 (+/- 0.02)
BA_G50	Gini	50	0.74 (+/- 0.01)	0.23 (+/- 0.04)

Table 5.1: *We obtained the weighted accuracy and F-measures of the bagging trees with the number of estimators chosen by 10-fold cross validation, based on F-measure. There was no significant difference in the performance measures of the trees split by Gini or Cross-entropy. It appears that this method does not do an adequate job predicting the minority class.*

5.2 Random Forests

Bagging often yields significant improvement in prediction accuracy over single trees by using averaging to reduce the variance of the model. However, when there is a very strong predictor in the data set, it is likely that most of the trees in the forest will split on that variable, leading to a set of highly correlated trees [6]. In other words, the bagged trees are identically distributed, but not necessarily independent and could have positive pairwise correlation ρ, making the variance of the average B trees:

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{5.2}$$

As B increases, the second term goes to zero, but the first remains, and the size of the correlation of pairs of bagged trees limits the benefits of averaging the high variance trees. [5, p. 588]

Random forests improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the bias too much. This is achieved in the tree-growing process through random selection of the input variables. [5]

Before each split, $m \leq p$ of the input variables are chosen at random as candidates for splitting. B trees from the random forest are denoted $\{T(x;\Theta_b)\}_1^B$. The random forest regression predictor is

$$\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T(x;\Theta_b) \tag{5.3}$$

where Θ_b characterizes the $b-$th random forest tree in terms of split variables, cut-points at each node, and terminal-node values. Reducing m will reduce the correlation between any pair of trees in the ensemble, and hence reduce the variance of the average. Typically $m = p/3$ for building regression trees and $m = \sqrt{p}$ for building classification trees [6]. For this data set, $m = \sqrt{29} \simeq 5.4$, therefore each node in each individual tree will only consider around 5 predictors at each split.

5.2.1 Tuning the Parameters of the Random Forest

Although methods such as Bagging and Random Forests do not over-fit based on the number of trees in the ensemble [5], reducing the number of trees results in making these methods more computationally efficient. Thus, the first tuning parameter considered was the number of trees to include in the ensemble. We built random forests of classification trees and ranged the number of trees in each forest from 1 to 60. Using 10-fold cross validation, the F measure was calculated on

each of the 10 validation sets. To better visualize the distribution of these F measures given the number of trees in the ensemble, we constructed the boxplots shown in Figure 5.2 and observed that the F1 scores appeared to stabilize around 50 trees.

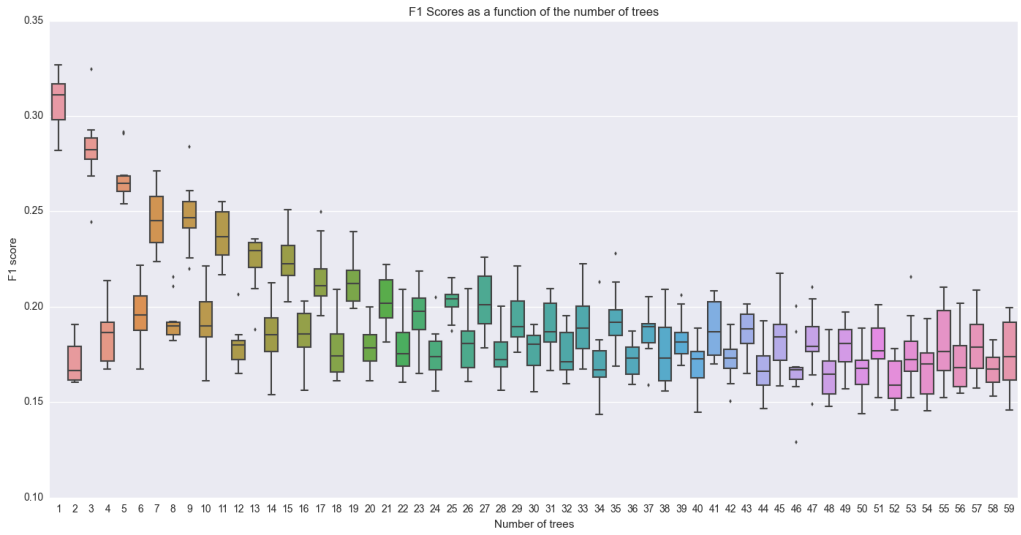

Figure 5.2: *We constructed random forests of classification trees with the number of estimators ranging from 1 to 59 and used 10-fold cross validation for each ensemble to generate 10 F-measures. Boxplots were used to visualize the distribution of the 10 classification scores at each random forest. The F-measure appears to again stabilize around 50 trees; therefore, the tuning parameter of the number of trees was chosen to be 50.*

We considered tuning the maximum depth parameter of the individual trees within the forest but found that, "using full-grown trees seldom costs much, and results in one less tuning parameter." [5, p. 596] Therefore, we do not prune the trees in the forest and instead only consider random forests of full-grown trees.

The next tuning parameter we considered is known as the "cutoff" for the prediction probabilities. A useful feature of the random forest method is that it can return the prediction probabilities for a given input vector x_0 [9]. For example, given an input x_0, among the trees in the forest, there could be a 25% probability that the observation belongs to class 1 and a 75% probability that it belongs to class 0. In this case, the predicted class for x_0 would be class 0 because under the typical random forest prediction scheme, the observation will be classified as class 1 if the probability of belonging to class 1 is greater than 0.50 and class 0 otherwise. We can alter this rule in the prediction step to favor the minority class in order to reduce the number of false negatives. For example, setting the cutoff to 0.20 rather than 0.50 will result in a prediction of class 1 for input x_0.

Figure 5.3 illustrates the F-measures at cutoff points ranging from 0.10 to 0.90 for a random forest of 50 trees. The F-measure is maximized at a cutoff of 0.20, while the median F measure at a cutoff of 0.30 is above 0.40, which is a significant improvement over previous methods.

Table 5.2 summarizes the performance measures for the random forest models. It appears that

the models without a specified cutoff value do nothing more than predict the majority class in order to minimize the overall classification error rate. Interestingly, applying a cutoff value less than 0.50 led to an increase in the F measure but at the expense of accuracy. The random forest with a cutoff of 0.20 had the highest F measure, but resulted in an accuracy of less than 0.50, which is undesirable. However, the cutoff of 0.30 resulted in a reasonable F measure without decreasing the accuracy too much.

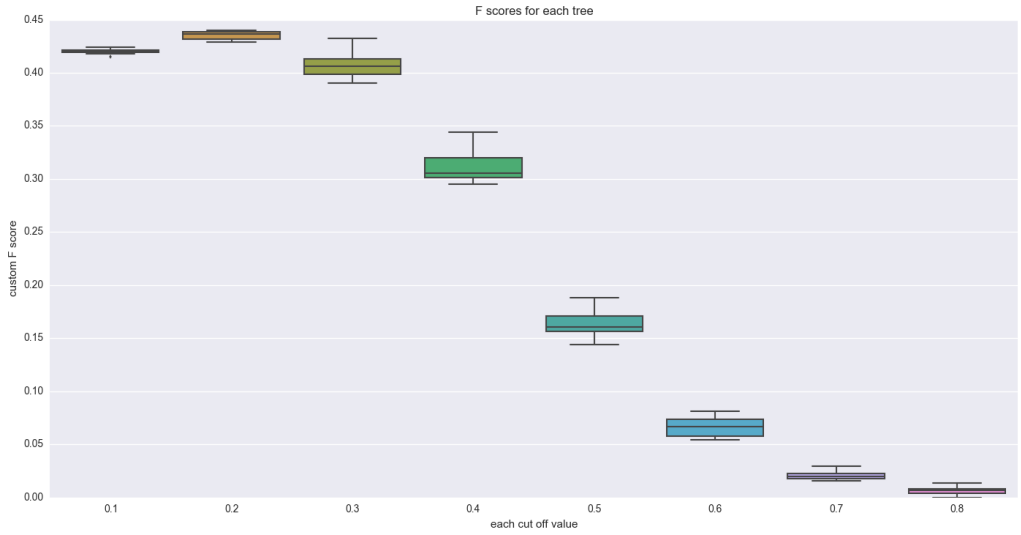

Figure 5.3: *Random forest methods can return the prediction probabilities for a given input vector x_0. The "cutoff" value refers to the rule at the prediction step of the algorithm, which states that if the probability that x_0 belongs to class 1 is greater than 0.5, then it will be predicted as class 1. We consider different values of the cutoff to favor the minority class in order to reduce the number of false negatives. The F-measure is greatest at the cutoff points 0.1-0.3. We will consider the cutoff points 0.2 and 0.3 in our random forest model.*

5.2.2 Variable Importance Measures

Random forest methods improve prediction accuracy at the expense of interpretability [6]. However, we can record the amount that the Gini index (or cross-entropy) is decreased by splits over a given predictor, averaged over all *B* trees. Thus, important predictors will have a large value of *variable importance*. Figure 5.4 is a graphical representation of the variable importances resulting from the random forest of 50 classification trees. The variables with the largest mean decrease in the Gini index are MED_ZIP_STATE_INCOME, TIMETOSURVEY, AGE, FOOD_ENVIRONMENT_INDEX, NUMERIC_VALUE and DIVERSITY_INDEX. Only two survey questions, B2 "How long did you have to wait to see your doctor after your scheduled time?" and B3 "Did you have any trouble getting a referral?", appear to have measurable importance.

Model Label	Criterion	Number of Estimators	Weighted Accuracy	F-measure
RF_E50	Cross-entropy	50	0.74 (+/- 0.01)	0.16 (+/- 0.03)
RF_G50	Gini	50	0.74 (+/- 0.00)	0.17 (+/- 0.02)
RF_G50 (cutoff=0.2)	Gini	50	0.49 (+/- 0.01)	0.43 (+/- 0.01)
RF_G50 (cutoff=0.3)	Gini	50	0.65 (+/- 0.01)	0.41 (+/- 0.02)

Table 5.2: *We obtained the weighted accuracy and F-measures of the random forests with the number of estimators chosen by 10-fold cross validation, based on the F-measure. There was no significant difference in the performance measures of the trees split by Gini or Cross-entropy. The random forest methods with the default cutoff point of 0.5 (RF_E50 and RF_G50) yielded very F-measures and likely predicted each of the observations as the majority class. However, the models with specified cutoffs led to a significant improvement in the F-measure. The random forest with cutoff=0.3 resulted in a higher F-measure without decreasing the accuracy as much as the model did with a 0.2 cutoff. Therefore, the cutoff tuning parameter was selected to be 0.3.*

This suggests that demographic variables, such as income, play a more important role in predicting how a patient perceives her experience with her doctor. Additionally, the amount of time in days between the doctor visit and the survey appears to matter significantly more relative to the survey questions. Despite this being a population of Medicare-aged members (typically over 65), differences in age appear to have measurable importance as well.

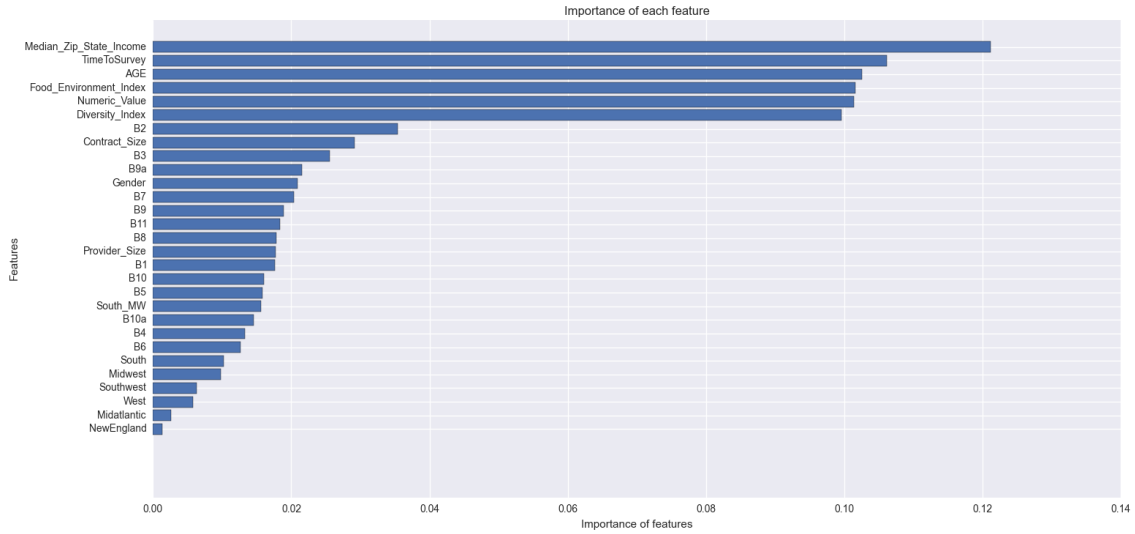

Figure 5.4: *The amount that the Gini index (or cross-entropy) is decreased by splits over a given predictor, averaged over all B trees in the random forest is known as the Variable Importance Measure. Higher values of variable importance correspond to stronger predictors. The variable importance plot illustrates that demographic variables and TIMETOSURVEY are more predictive relative to the survey questions. Interestingly, this suggests that how a patient perceives her experience with her doctors depends greatly on her demographic characteristics.*

Chapter 6

Answers to the Original Research Questions

In this section, we will summarize the results of the three research questions of interest. It is important to reiterate that these results are inherently biased and can therefore only be applied to the fraction of the population who opted to complete the entire survey, rather than the entire Medicare population of the health plan.

What distinguishes populations who score a low NPS from a high NPS?

In section 3.1, we showed that the two classes of patients were not separable due to the large overlap in their distributions of various predictors. Thus, there is no intuitive distinguishing classification of patients with a low NPS response versus a high NPS response with the given predictors. However, we showed that our proxies for income (*Median_Zip_State_Income*, *Food_Environment_Index*, and *Numeric_Value*) and ethnicity (*Diversity_Index*) were most predictive of NPS response. Therefore, a more comprehensive study using the actual income and ethnicity values may produce better classification results.

Which survey questions are most predictive of NPS?

The most predictive survey questions were B2 "How long did you have to wait to see your doctor after your scheduled time?" and B3 "Did you have any trouble getting a referral?". These were determined by the variable importance measures that were measured from the random forest model with 50 trees.

How well can we predict NPS using only the responses to the survey and demographic factors?

We determined the best model to be the random forest with 50 trees and a cutoff value of 0.30. This model was able to produce a reasonable F measure relative to the other models while still maintaining an adequate classification accuracy.

Therefore, given these predictors, along with the fact that the class distributions greatly overlap, we were able to achieve a prediction accuracy of 0.65. The highest F-measure we were able to

reach is 0.42. Additionally, the AUC of this method on the testing set (see Figure 6.1) was 0.63, which is moderately better than random guessing.

Model Label	Precision	Recall	F-measure	Weighted Accuracy
RF_G50 (cutoff=0.30)	0.37	0.48	0.42	0.65

Table 6.1: *The best model selected was the Random Forest with 50 estimators and a cutoff value of 0.30. This model yielded an F-measure of 0.42 and an accuracy of* 0.65 *on the testing set.*

	Predicted Positive Class	Predicted Negative Class
Actual Positive class	19,640	21,660
Actual Negative class	33,130	82,278

Table 6.2: *The confusion matrix of the predictions of the random forest model highlights the fact that over half of the Positive class was predicted as the majority positive class.*

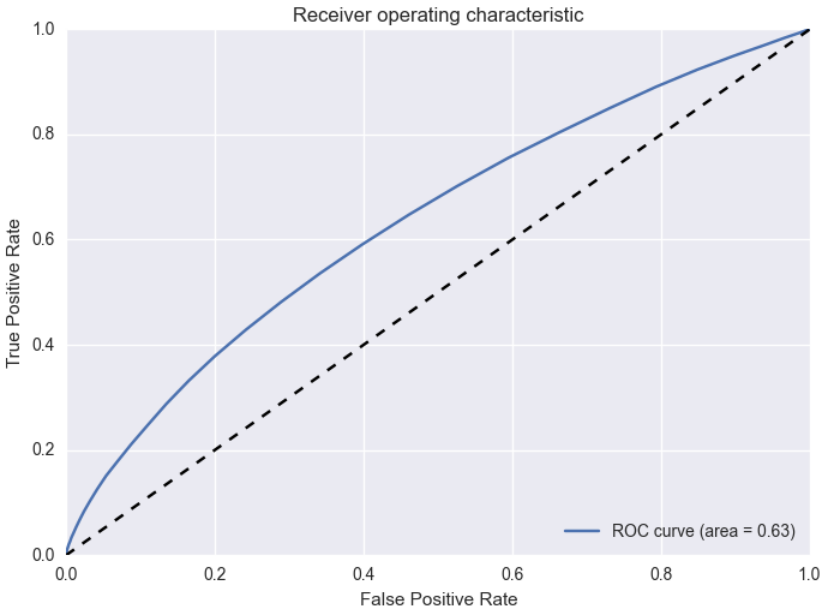

Figure 6.1: *This is the ROC curve for the random forest model with a 0.3 cutoff value. The AUC value of 0.63 is closer to a coin toss (0.5) than the ideal value of 1.*

Bibliography

[1] Alan, Smedley, and Stith. "Free Executive Summary of Unequal Treatment: Confronting Racial and Ethnic Disparities in Health Care". National Academies. http://coe.stanford.edu/courses/ethmedreadings06/em0601garcia2.pdf

[2] "CAHPS ©: Assessing Health Care Quality From the Patient's Perspective". U.S. Department of Health & Human Services. https://cahps.ahrq.gov/about-cahps/cahps-program/cahps_brief.html

[3] Chen, Liaw, Breiman, "Using Random Forest to Learn Imbalanced Data," University of California, Berkeley Statistics Department, http://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf

[4] Evans, Hund, Varley, Wang. "Predictive Power of a Generalized Care Segmentation Model". 2014 WPI REU Program in Industrial Mathematics, Worcester Polytechnic Insitute, Worcester, MA. July 25, 2014.

[5] Hastie, Tibshirani, Friedman, 2009, Springer, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* 2nd Edition, pp. 305-310;587-596

[6] James, Witten, Hastie, Tibshirani, 2013, Springer *An Intorduction to Statistical Learning with Applications in R*, pp. 37-39, 231, 303-321

[7] Jeevan, Manu. "Dealing with Unbalanced Classes ,Svm, Random Forests And Decision Trees In Python." Data Science Blog. Big Data Examiner. February 20, 2015 published. March 1, 2015 accessed. http://www.bigdataexaminer.com/dealing-with-unbalanced-classes-svm-random-forests-and-decision-trees-in-python/

[8] Paffenroth, Randy. "Trees and Some SVM (probably)." MA 543: Class 19. WPI, Worcester, MA. Mar.-Apr. 2015. Lecture.

[9] Scikit-learn: Machine Learning in Python, Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, Duchesnay, *Journal of Machine Learning Research*, Vol. 21, pp. 2825-2930, 2011

Appendix A

Appendix

A.1 Balanced Random Forests

In learning extremely imbalanced data, "there is a significant probability that a bootstrap sample contains few or even none of the minority class, resulting in a tree with poor performance for predicting the minority class." [3] For the tree classifier, artificially making class priors equal either by down-sampling the majority class or over-sampling the minority class is usually more effective with respect to a given performance measurement [3]. However, down-sampling the majority class may result in loss of information, as a large part of the majority class is not used. The Balanced Random Forest is an ensemble of trees induced from balanced, down-sampled data [3]. Its algorithm is shown below:

1. For each iteration in a random forest, draw a bootstrap sample from the minority class. Randomly draw the same number of cases, with replacement, from the majority class.

2. Induce a classification tree from the data to maximum size, without pruning. The tree is induced with the CART algorithm, with the following modification: At each node, instead of searching through all variables for the optimal split, only search through a set of m_{try} randomly selected variables.

3. Repeat the two steps above for the number of times desired. Aggregate the predictions of the ensemble and make the final prediction.

Unfortunately, we were not able to implement this algorithm properly, and so the results we obtained were not reliable. As future work, we will attempt to properly implement the balanced random forest and compare the model's performance to our other models.

A.2 Results Using the 1-5 NPS Scale

Using decision tree classification, several trees were constructed on a training set and tested on a validation set. Below is the confusion matrix for the full tree. Here, the NPS responses were scaled

from 1 to 5, and the original predictor data set was used. The diagonals of this matrix correspond to the proportion of the testing set that was correctly predicted. Overall, the full tree accurately predicted only 33.89% of the testing set. With well over 100 terminal nodes, the full tree highly over-fit the data, and so tree pruning was desired.

Table A.1: Confusion Matrix for Full Decision Tree

	Predicted Values				
Actual Values	1	2	3	4	5
1	.2130	.0992	.0942	.1422	.4515
2	.2035	.1082	.1076	.1445	.4363
3	.1738	.1102	.1216	.1518	.4426
4	.1755	.0917	.1001	.1530	.4797
5	.1670	.0820	.0873	.1438	.5199

The first tree-pruning strategy was to alter the stopping criteria on the full tree by setting a minimum number of observations in each leaf node. Specifically, the first pruned tree required that each leaf contain at least 1,000 observations. Below is the confusion matrix for the predictions made with the pruned tree. While this tree correctly predicted 50.0% of the testing set overall, it failed to adequately model the responses ranging from 2 to 4, which is evidence of high bias.

Table A.2: Confusion Matrix for Pruned Decision Tree

	Predicted Values				
Actual Values	1	2	3	4	5
1	.0405	0	0	0	.9595
2	.0254	0	0	0	.9746
3	.0172	0	0	0	.9828
4	.0135	0	0	0	.9865
5	.0113	0	0	0	.9887

The next strategy was to perform bagging on the classification trees in order to reduce variance without increasing the bias of model. Below is the confusion matrix of the bagging classification tree with 10 trees in the ensemble. This method outperformed the single full-tree, making a total of 42.73% correct predictions. While this is a measurable improvement, the test error rate is undesirably high.

Next, a random forest was constructed in an effort to further improve prediction accuracy by de-correlating the trees in the ensemble. The random forest slightly outperformed the bagging classification trees with an overall 49.79% correct predictions on the testing set. Although the overall percent of correct predictions is higher, the random forest appears to do a worse job predicting NPS responses less than 5.

Table A.3: Confusion Matrix for Bagging Tree

	Predicted Values				
Actual Values	1	2	3	4	5
1	.2212	.0492	.0440	.0616	.6240
2	.2031	.0661	.0491	.0797	.6019
3	.1632	.0595	.0617	.0796	.6360
4	.1532	.0405	.0517	.0855	.6692
5	.1303	.0326	.0327	.0712	.7331

Table A.4: Confusion Matrix for Random Forest

	Predicted Values				
Actual Values	1	2	3	4	5
1	.1004	.0070	.0066	.0057	.8803
2	.0902	.0096	.0063	.0053	.8886
3	.0574	.0072	.0091	.0099	.9163
4	.0365	.0030	.0045	.0085	.9475
5	.0291	.0028	.0018	.0048	.9615

Table A.5: Random Forest Predictions using 1,000 Decision Tree Classifiers

Training Data	NPS Scale	Criterion	Tree Specifications	%Test Correct
Simple random sample from 20% of Population	1-5	Gini	Full Tree	49.81
			Max Depth = 5	TBD
			Max Depth = 7	TBD
			Max Depth = 9	TBD
Stratified Random Sample on NPS score	1-5	Gini	Full Tree	TBD
			Max Depth = 5	TBD
			Max Depth = 7	TBD
			Max Depth = 9	TBD
Simple random sample from 20% of Population	0-2	Gini	Full Tree	TBD
			Max Depth = 5	53.58
			Max Depth = 7	54.01
			Max Depth = 9	54.22
			Max Depth = 11	54.34
Stratified Random Sample on NPS score	0-2	Gini	Full Tree	61.14
			Max Depth = 5	TBD
			Max Depth = 7	62.00
			Max Depth = 9	62.17
Simple random sample from 20% of Population	0-1	Gini	Full Tree	TBD
			Max Depth = 5	62.28
			Max Depth = 7	62.79
			Max Depth = 9	63.04
			Max Depth = 11	TBD
Stratified Random Sample on NPS score	0-1	Gini	Full Tree	61.14
			Max Depth = 5	TBD
			Max Depth = 7	62.00
			Max Depth = 9	62.17
			Max Depth = 11	TBD