Acknowledgments

I'd like to thank my parents, Jean Memken and Michael Rogers, for their financial and emotional support throughout this process, as well as my sister Sarah and brother Jonathan. I'd also like to put in a word for Chris Petkus, who convinced me that I could actually write this. This thesis would have also been impossible without the intervention of Mark Sump, my employer, and everyone at the Wakefield Campaign who helped collect the data.

Table of Contents

Background and Statement of the Problem

In the summer of 2013, I was hired as an account executive at Activate LLC, a company that focuses on the sale and support of phone surveying software and conducting surveys. Activate's clients include unions and non-profit organizations, but the majority of the company's purchases come from political campaigns. I also developed analytical methods for the accurate description of potential voter preferences as part of my duties as an account executive. These methods focused on the analysis of survey data.

When analyzing the survey data provided by campaigns, I used ordinary least squares and generalized linear models on the data, but these techniques had problems that interfered with modeling support. While these methods made it possible to estimate likelihood of voter support for a particular candidate or ballot initiative, they proved unsatisfactory during analysis due to the time and consideration needed for selecting relevant independent variables capable of predicting support in different environments, for different initiatives, and for different candidates. Most problematically, the regression analyses never provided a satisfying answer to what I still see as one of the most essential problems with the data. With the large number of independent variables, the logistic regression and ordinary least squares methods fall short when looking for relevant independent variables from a larger collection than an individual modeler could consider and sift through.

We face a problem of variable selection when working with a dataset that includes columns representing many potential independent variables and determining which variables should be included in a particular model. The variables whose inclusion campaign managers demanded were rarely sufficient for the accurate description of voter preference. Exacerbating the problem of selection, campaigns often provided hundreds of columns of data on surveyed voters, and the quantity of information provided made sorting through the variables to find those most capable of predicting voter support difficult. As time went on, my employer described my work as 'reinventing the wheel' for each modeling project performed, due to my need to consider the relevant variables of different data sets. While certain variables did come up frequently, there was no guarantee that the selections which effectively defined voter support in a municipal race in New York or a ballot initiative in Missouri could serve to describe a race for the House of Representatives in Kansas, and the question of finding information that predicted voter support in the data provided by state and local parties was of crucial importance.

The problem of finding relevant variables for predictive modeling came to a head in the summer of 2014, when I was working on a way of accurately describing the voting universe for Margie Wakefield's campaign for Kansas' Second House district. Collecting data for the Wakefield campaign, the company conducted a survey on voters across Kansas' second district. I had some input on the survey's design. After the phone interviews, the Kansas Democratic party provided data on the individuals surveyed. The data provided included 313 columns, many of which were likely irrelevant, but the party promised a collection of variables found to be relevant, which could be matched to a dataset and used to predict support of all potential voters in Kansas' Second district. The information available in the campaign dataset included multiple spreadsheets of data on the surveyed population, with 313 variables between the 1,540 respondents.

With the goal of creating a method to parse and select independent variables for a model that would designate who the campaign would try to reach and turn out, I created the BeSivA algorithm. I discuss the algorithm below and compare it to other methods of classifying observations which I have learned about since the campaign ended. I based the algorithm on a consideration of the demands made by the campaign manager and my own personal desire for an algorithm that selects by a criterion that differs from those that focus on the training data, or data used to create the model (Kuhn 2013). When compared to other developed methods, BeSiVa performs a more interpretable fit on the data it uses, and has a performance comparable to penalized regressions with when comparing on a set of validation data. It also pares down the independent variables and runs with data that contains missing values. For these reasons, the BeSiVa algorithm serves alongside other methods of predicting a dependent variable, serving as a tool that campaigns may use to better understand the universe of voters they are trying to reach.

While finding the right instrument to create a predictive model is a classic problem in the field of 'big data', it is also necessary to find relevant variables in provided voter data. In addition to a campaign's scope frequently prohibiting the use of an accurate surveying instrument on a majority of the electorate, the question of how to use the parties collected voter data becomes salient when analyzing a small survey for such a campaign. The Kansas Democratic Party had access to a large database of voter information, dealt out based on necessity to political campaigns in the state, including information on whether individuals voted in specific races, local census data including population density and median income, and voters' consumer choices. In attempting to provide the campaign with useful information, the party offered a vast collection of data to consider.

Finding a means of looking through the entirety of the data and finding the relevant variables was not only essential due to their quantity, but also due to the party's unwillingness to provide data on all possible voters. While generous with the data for individuals surveyed, the party was less interested in providing the entirety of the information for reasons of security and the time needed to get the information, preferring to keep the number of independent variables provided for all potential voters low. Thus, the client wanted a model that kept the number of relevant independent variables below a fraction of the total number of considered independent variables, as it was cheaper in terms of time and energy to collect less data from the party.

The data available to campaigns is expansive, but getting such a large variable collection for a small collection of voters comes with drawbacks. As a majority of these variables are irrelevant or likely to be collinear, with some offering fewer contributions to a model than other predictive independent variables, the question of parsing the large collection of data to find independent variables that relate to voter support remains challenging. This challenge is exacerbated by the understanding that some variables are highly likely to be relevant or preferred by campaign officials, based on preexisting understanding of how voters differ within the electorate, such as by political party, geographical population distribution such as county, and whether a voter turned out in elections similar to the one being modeled. Thus, any attempt to consider voter support quantitatively demands a means of building instruments designed to predict voter interest in candidate positions, as well as to sift through the vast quantity of information collected by the party to determine voter support for a candidate based on specific messaging.

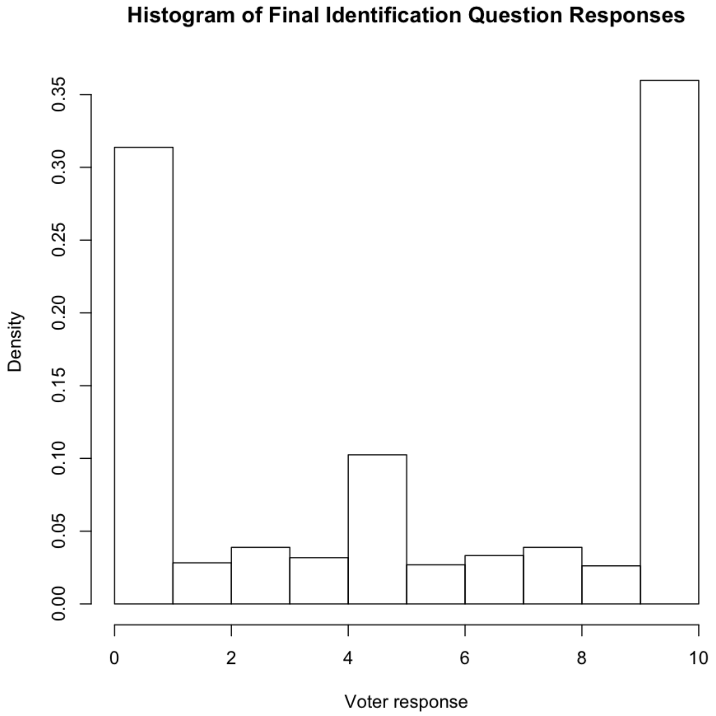
The Dependent Variable

The campaign collected survey data that included questions to address constituents' opinions on categorical issue positions, which were used to tailor messaging as well as to collect more detailed survey data used which served as the dependent variable. The survey consisted of six questions, designed to help the campaign fine tune its messaging and reveal a voter's overall affect for the candidates. The introductory survey question focused on issues and messaging relevant to the campaign, and this question changed between phases of the survey. The question was intended as a way of warming up the voter to taking the survey and was not used in the modeling. The survey workers also asked three questions to determine a voter's preference between the candidates' issue positions. For these questions, the survey worker conducting the survey first read a message associated with each candidate, both designed by the campaign, and then asked the voter to choose which candidate they agreed with on a scale of one to ten, with one representing complete support of the opposition based on the campaign's interpretation of the opponent's position, and ten representing complete support for the campaign's position. A sample issue question can be read in the appendix. Each of the messages followed the same format as the sample question, including brief descriptions of the positions held by the candidate and their opponent, and asking each voter to rate their support for the positions from one to ten. These questions were interspersed with the primary focus of the modeling, which the campaign referred to as an identification question.

The primary goal of the survey was determining overall voter support for the candidates and their positions, and the identification questions required voters to rate each candidate directly. Survey workers asked two identification questions about candidate support (one at the start, and one at the end of the survey). The identification questions named the candidates, their respective parties, and asked the voters to choose which candidate they would be more likely to support on a scale from one to ten in the 2014 Kansas Second District House Race. Selecting one meant the voter agreed or identified with Lynn Jenkins, the opposing candidate, and ten meant they agreed or identified the most with Margie Wakefield. Voter identification questions were placed deliberately. One identification question appeared immediately after warmup messaging question, while the other was the final question asked in the survey. This final question, which can also be seen in the appendix, served as the primary focus of the modeling for the campaign.

While I had persuaded the client that the underlying relationship needed a fine distinction to capture, the usage of the 1-10 scale became the next problem to consider. Although survey workers had been collecting the data in values from one to ten, the histogram of the data suggested that the 1-10 division was too fine a distinction to capture underlying relationships, or that the actual relationship was truncated, limiting the electorate when their actual beliefs might be more nuanced. Focusing on the histogram of the dependent variable, which can be seen below in figure 1, the pattern is almost trimodal, primarily concentrated on the edges of the distribution but with a small peak in the center. This shaping indicates a severe polarization in the electorate, with 65.9% of surveyed voters supporting either of the candidates wholeheartedly and 34.1% standing between the most polarized positions (with 10.1% choosing 5, the undecided option). While some regression options that used the scaling in the survey could be used to model voter identification, the client's concern with support led to a different conclusion, leading to the dichotomization of the dependent variable.

Figure 1: A histogram created in R of voter responses to the final identification question as a density estimate. Note the trimodal nature of the data, which lead to the decision to recode the dependent variable as dichotomous .



The scaling was reconsidered based on the client's preferences, as their main concern was the likelihood of voter support. Looking at whether voters supported Margie Wakefield on the position articulated during the survey, the data was recoded as a dichotomous variable. While the scaling may have truncated voters' beliefs, the options most appealing to the client led me to the conclusion that dichotomization was preferable to any truncated approach. Survey workers had been coached during that any voter response of five or six would be considered undecided, but the recoding reflected difficulty among survey workers (who were all campaign volunteers) in recalling that as part of the dichotomization of numeric responses. Thus any value below 6 was recoded as a zero, indicating that the person did not support Margie Wakefield on the issue or identification question, and any value above or including six was recoded as 1, a response that supported Margie Wakefield and would hopefully lead to support at the polls. Using this dichotomization with a logistic regressions, the results rose above the mode value percentage, but despite its ability to predict dichotomized dependent variables the base glm regression in R did not allow for an appropriate consideration of the independent variables and how the large quantity of provided data could be included in the development of a model of voter support.

When developing the support model, a few of the client's preferences came into play when selecting the instrument used. The client desired a model that predicted the support of individual voters based on the survey data collected, was highly interpretable, and had a smaller number of independent variables due to the state party's constraints on collection. While the primary goal, predicting voter support, was non-negotiable, interpretability and number of independent variables were, meaning that a model could have more independent variables if it improved interpretability or made better predictions, so long as the number of independent variables fell below a fraction of the total provided. As the number of independent variables increased, the difficulty in collecting that information grew, and this was something avoided by choosing the variables used for prediction rather than using all variables.

An Overview of the Independent Variables

The independent variables came from a preexisting database provided by the Kansas Democratic Party. This data included useful variables, including census data such as ethnicity, housing information, and level of education. In addition, the party collected data on individual voting records, and made available whether someone was a voter in any previous national or Democratic primary election up to 1996. The party also provided independent variables that included consumer data. While these variables seemed like they might have promise in predicting support, the data also came with many seemingly irrelevant consumer variables. The provided variables provided expansive information on the survey respondents, but this became problematic due to relevance and variables needed to be dealt with via removal.

The large quantity of potentially useful information came with a fraction of the data that could not be incorporated into modeling, and this was dealt with as it arose. Some variables were redundant, such as multiple gender, age, and party variables that had been included in several forms as binary categorizations, while other variables were missing most of their values. Two of the primary independent variables had zero variance, as none of the people who responded to the survey had voted in the last two democratic primaries, and the client requested that other variables be removed, including columns of predictions made by a different organization. Even after the ad hoc removal of these variables, however, 184 columns of independent variables remained, with each holding potential predictive power. There are several reasons why conventional regression techniques may cause problems with 184 independent variables as possible candidates for inclusion. Placing all independent variables into a single regression would make paring down the data difficult due to the certainty of multicollinearity, even with redundant variables removed. The client was also interested in using a minimum of needed variables, as any requested data would arrive from the party more quickly if fewer variables were used. The party's provision of a vast quantity of data came with relevant and irrelevant variables, cut down to create the best possible model from the remaining 184 independent variables.

Due to the large number of remaining independent variables, full inclusion was problematic. Kuhn and Johnson (2013) suggested the use of cross-validation for comparing the performance of different types of models, but had no suggestions for dealing with an excess of independent variables outside of adding them all into one model. Adding all variables into the model was still possible, but undesirable due to the effects of listwise deletion. With 184 columns of data, most of the independent variables had missing values, and using listwise deletion eliminated 1,355 observations, or 87% of the rows, from consideration. While the number of observations dropped to 185, meaning that all remaining independent variables could still be included in a regression, this was unclear during the removal process, and an alternative to the elimination of all data with missing observations was sought. This was due to the client's preference that all data be incorporated, but listwise deletion can also lead to biased parameter estimates, if the data is missing in a systemic manner (King et al. 2001). Thus, an alternative to total listwise deletion was needed, as it would have removed useful data and left biased coefficients.

In dealing with the missing data, two options have become standard, listwise deletion, which has problems with bias in estimation, and imputation. Imputation, the systemic creation of data based on the values in similar nonmissing observations is recommended. King recommends creating multiple imputed data sets, and combining them into one set of simulations (2001), but does not offer suggestions for each of variable selection techniques considered. With the exception of the random forest technique, which has incorporated another means of dealing with missing data in some implementations, the other techniques had issues with missing data that prevented their use without imputation. As no particular means of pooling the imputations exists for each method considered, a single set of imputed data was created to allow for the comparison of variable selection tools.

Preexisting Options for selection

When considering a problem that demands the removal of irrelevant independent variables in a dataset to create a new model, there are three options for types of methods that select independent variables. The first option involves systemized methods of independent variable choice based on specified criteria, such as best subset selection and the forwards and backwards stepwise selection methods. These methods are similar to the BeSiVa algorithm but prove unsuitable due to some conceptual concerns. The second option includes several methods that take all provided independent variables and constrain irrelevant variables' coefficients to zero, a subset of penalized regression. The third set of methods have only been recently created, and include such techniques as boosted regression trees and the random forest technique. Each of these method types have some problems, however, that made them less than desirable for determining voter support and led to the creation of the BeSiVa algorithm.

While thematically similar to the created algorithm and capable of dealing with missing data, the first type of method for choosing relevant variables, which includes two subtypes (1) subset selection and (2) stepwise methods, is problematic due to conceptual and computational limitations. In order to create highly interpretable models that do not require the use of imputed data, best subset selection focuses on building a model with every possible combination of independent variables and choosing the best model from all potential candidates. Best subset selection chooses the best model using a statistical criterion such as adjusted R^2, the Akaike Information Criterion, the Bayesian Information Criterion, or Mallow's C_p to determine which regression best predicts data outside that used for modeling while using only the data included in the model. While versatile in criterion and types of models used, and guaranteeing the best model for the specified criterion, best subset selection must run 2^p models, where p is the number of potential independent variables. This limitation means that best subset selection requires vast computational resources for any collection of independent variables greater than 40 (James et al. 2013), and the use of every combination of the 184 independent variables in the data results in 2.45×10^{55} different models, making the best-subset selection method unsuitable for this problem due to the inordinate length of time needed to create these models. Best subset selection is fully capable of finding the best model dictated by a criterion, but it requires excessive computational power due to its consideration of every possible combination of independent variables.

Avoiding the computational requirements of best subset selection is preferable, but its less computationally expensive counterparts methods remain problematic due to theoretical obstacles to good results. To approximate the results of best subset selection without its computational limitations, stepwise selection methods run multiple regressions with independent variables chosen by their impact on the fit. Stepwise selection seeks to find the predictors that create the best model by adding or removing predictors depending on whether forward or backward stepwise selection is chosen. With forward stepwise selection, the dependent variable is fit without independent variables, which are added one at a time based on which independent variable best improves the overall fit on the training data. The final number of independent variables is a tuning parameter that must be adjusted based on a model criterion such as the ones used in best subset selection, but forward stepwise selection is advantageous for several reasons. As described, forward stepwise selection fits a comparatively small fraction of the models seen, a maximum of 17,021 for the 184 predictors in the campaign's dataset, making it much more computationally feasible than best subset selection. Forward stepwise selection can always be used, no matter the number of predictors (Hastie et al. 2009), making it more practical than backward stepwise selection in this regard. Despite creating the same number of models, backwards stepwise selection can only be run if there are fewer independent variables than there are observations. If this condition is met, however, backward stepwise selection guarantees the consideration of all independent variables, placing them into a regression and removing one when the independent variable fits a specific statistical criterion such as highest p-value, allowing backward subset selection to find the single best model among those that can be created, using the same criteria as best subset selection (James et al. 2013).

Despite their ability to wander unaided through independent variables to select models that fit the data well, the stepwise and best subset selection methods have faced criticism for questionable statistical properties. On a frequently referenced webpage in the STATA support message boards, biostatician Frank Harrell discusses some of these problems, describing how stepwise regression R^2 values are biased to be high, due to the the method's proclivity for adding variables even they add very little information or prove theoretically irrelevant. Harrell continues by describing how F and Chi square tests for each variable do not have the claimed distribution. The confidence intervals and predicted values are falsely displayed as narrow, while the selection method's p values are difficult to fix. The methods are also described to have severe issues when collinearity is a factor. Harrell also describes how the issues raised are not helped by altering facets of the problem, such as adding data or working from best subset rather than stepwise methods, and condemns the methods by describing stepwise solutions as a substitute for thinking about the problem at hand (1998). Although sharing thematic similarities with the BeSiVa algorithm, the subset selection methods were not included in the comparison due to their problematic statistical properties.

Penalized regression analysis is a means of assigning weights to each included independent variable, substituting the unbiased regression used in the default form of generalized linear modeling for penalized regressions. A penalized regression decreases model variance by adding bias to each estimate in the form of a penalty. Using the form of OLS as an example, regressions are created by minimizing RSS in equation 1,

$$RSS(\beta) = \sum_{i=1}^{N} (y - f(x_i))^2 \tag{1}$$

where

$$f(x_i) = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j \tag{2}$$

Ordinary least squares then chooses estimates of β in equation 2 to minimize the difference between the dependent variable y and the value of $f(x_i)$. Instead of letting the coefficients vary freely, however, the penalized regressions each add a penalty to equation 1, changing it so that the minimization takes into account the size of the predicted coefficients using a penalty function multiplied by a tuning parameter (Kuhn and Johnson 2013), λ, as seen in equation 3

$$\sum_{i=1}^{N} (y - f(x_i))^2 + \lambda \sum_{j=1}^{p} penalty(\beta_j) = RSS(\beta) + \lambda \sum_{j=1}^{p} penalty(\beta_j) \tag{3}$$

By adding this penalty, the coefficient estimates are constrained. This penalization has similarities to the the way the adjusted R^2 shrinks the variance accounted for by the independent variables, preventing unnecessary independent variables from overstating the variance for which they account. The ability to shrink coefficient estimates to zero, invalidating the inclusion of unrelated independent variables in a model made the use of some of the penalized regression methods viable alternatives to the method created.

There are three kinds of penalties commonly in use in penalized regression: ridge regression, lasso, and the elastic net. The original method, ridge regression, adds a penalization based on the square term of the coefficients as seen in equation 4 below,

$$\lambda\sum_{j=1}^{p} penalty(\beta_j)=\lambda\sum_{j=1}^{p}\beta_j^2 \tag{4}$$

constraining overly large coefficient values. Although coefficient values can be driven to zero by a ridge regression, this is not guaranteed, and a test ridge regression used all 184 columns of provided data, compared to 28 columns used by the lasso for the same data. Of the three methods of penalized regression, only two guarantee shrinkage, the lasso and the elastic net (Hastie et al. 2009). The lasso adds a penalization based on the absolute value of the coefficients' sizes to the equation to minimize, as seen in equation 5

$$\lambda\sum_{j=1}^{p} penalty(\beta_j)=\lambda\sum_{j=1}^{p}|\beta_j| \tag{5}$$

which drives irrelevant coefficient estimates to zero for a large enough tuning parameter (James et al. 2013). The elastic net adds a different penalty to the value to be minimized. Its penalty is the sum of the lasso and ridge regression penalties and is displayed in equation 6

$$\lambda\sum_{j=1}^{p} penalty(\beta_j)=\lambda(\alpha\sum_{j=1}^{p}|\beta_j|+(1-\alpha)\sum_{j=1}^{p}\beta_j^2) \tag{6}$$

mirroring the form of the penalty calculated in the glmnet package. glmnet, created by Friedman, Hastie, and Tibshirani (2010), gives users of R the ability to perform penalized regression analysis, extending the penalized regressions for gaussian, logistic, and other link functions used by the glm command, allowing for options within these models including shrinkage In addition to the λ parameter, the glmnet package also has a version of the elastic net that includes a second tuning parameter, α. The value for α falls between 0 and 1, and varies the contribution of each penalty to the final coefficient value. When each penalty is chosen to minimize the RSS using cross-validation or bootstrapping, the elastic net's dual penalties can outperform the lasso for prediction while still allowing for coefficient shrinkage (Zou and Hastie 2005).

When working with basic regression techniques in all modern software, listwise deletion is the default for working with missing training data, eliminating all rows with missing data from consideration. The glmnet package is further limited in this respect, refusing to run any regression unless the training data does not include missing values, requiring a separate step for listwise deletion or imputation (James et al 2013). Despite the diverse regression options in glmnet, the package is limited due to its dependence on imputed data, meaning that the same data with missing values will provide different results due to the requirement of imputation or listwise deletion. While the use of a single imputed data set guarantees the same result for each use of the regression and was used for comparison, the usefulness of methods that cannot function unless input data has no missing values or is imputed remains questionable, especially for the campaign's data.

In addition to the penalized regressions, other methods of independent variable selection have been developed more recently. One such method, the random forest algorithm, classifies observations based on a random sampling of available predictors. From these predictions, the random forest proceeds as a bagging algorithm, determining which divisions in a bootstrapped set of data predict a dependent variable classification among the sampled predictors, and then classifying the dependent variable based on the designations the random forest algorithm decided were relevant (James et al. 2014, Hastie et al. 2009). This method then allows for the prediction of whether someone supports a candidate allows for the categorization of an observation while creating a highly interpretable regression tree (Liaw, A. and M. Wiener 2002), but its use is also problematic due to its use of all provided independent variables.

The implementation of the random forest technique used in the main comparison came from the randomForest package. This implementation was only able to run with the imputed set of data, rejecting any data set that included missing values. There are options for a random forest that include missing values within the implementation of random forest in the package known as party, which relies on surrogate splits for dealing with missing values. With surrogate splits, missing data is classified and considered separately by forming a list of surrogate independent variables that resemble the main interdependent variable used to split the data, so that missing values can still be used to predict outcomes (Hastie et al. 2009). The use of surrogate splits was problematic, however, as a test of the party package's implementation of dealing with missing data still had problems making predictions with an accuracy approaching the methods using imputed data, including the randomForest package implementation. In addition, both implementations of random forest used all included independent variables to make their predictions, something the client wished to avoid. While still capable of making predictions, even on missing data, the random forest technique was problematic for predicting voter support under the parameters the client was working with.

Similar to the random forest technique, the 'best off the shelf classifier', the Adaboost.M1 algorithm, serves as a means of dividing observations by classification, but issues with the algorithm made implementation for the campaign's data undesirable. To make predictions based on a training set, Adaboost.M1 generates an initial classifier for the training set, weighting each observation equally in its consideration. It then reconsiders its predictions, increasing the weights of observations where the classifier failed to correctly classify in its previous attempt. The algorithm then weights each classifier by their success, and adds each of the classifiers together multiplied by their weights to create a final classifying function, which is then used to make predictions (Hastie et al. 2009). While this method appeared to have results similar to the other methods for imputed data, the Adaboost.M1 algorithm was unsuitable for the problem of predicting voter support.

Despite its vaunted reputation, the Adaboost.M1 algorithm had issues with the predictions created making it difficult to implement for a campaign. While creating the best prediction remains the primary goal of trying these methods, there Is still some need for parsimony, as the Kansas Democratic party was unwilling to provide a dataset with all variables for the entire group of voters. In addition to using a majority of provided independent variables, the output of Adaboost.M1 is challenging to interpret compared to the penalized regressions, random Forest, and BeSiVa, returning a vector of relative importance that took the variable's Gini index gain and tree weight into consideration when ranking (Alfaro et al. 2013). The low interpretability and high number of independent variables that Adaboost.M1 requires to make predictions makes it difficult to justify using to classify voter support. In addition, Adaboost.M1 overfits the provided data dramatically, creating a set of perfect predictions of the imputed training dataset, and making predictions on the test set with a greater error rate than penalized regression. Finally, Adaboost.M1's relationship to missing data differed from the other methods, but it still failed if any of the dependent variable data was missing. While the algorithm can run if there are missing values in the independent variables, AdaBoost.M1 fails if the dependent variable contains missing values. Thus despite its reputation, Adaboost.M1 is unsuitable for modeling voter support, due to its low interpretability, need for the majority of the independent variables considered, overfit on the test set, difficulty in dealing with missing data.

A Description of the BeSiVa Algorithm

Independent of the methods already described, I created the BeSiVa algorithm to determine the likely support of voters in a race for Kansas' Second District, using data from a survey instrument collected in the state and data provided by the state party. Having preprocessed the data to remove irrelevant independent variables and dichotomize the dependent variable, I subsetted the data. The data was separated into two sets: training (X1) and pseudotest (X2) data, named because unlike test data, which is used to determine overall model performance (Kuhn 2013), X2 is incorporated into the algorithm as part of selecting independent variables. The pseudotest data was a randomly selected set of records constituting 10% of survey results, with the remaining 90% serving as the training data. A set of independent variables were identified as always included variables, or AIVs, which would always be included in any created models, due to potential theoretical relevance and the demand of the campaign manager. This combination of creating a validation set and designating AIVs shaped how the algorithm used the data set when operating.

After the independent variables were selected and organized, the algorithm could proceed. All models, regressed using X1, were built from the glm function in R, featuring a logistic regression described in the appendix. The first model created, designated M1, regressed the dependent variable against the AIVs, or if no AIVs were designated, the first independent variable in the list was used. Once M1 had been created, it was used to analyze X2. These predictions were compared to X2's dependent variable data, calculating the percent of correct predictions (hereafter the PCP). The variables in M1 and their PCP were stored, and the first loop created individual models M2i, where i = 1...p. M2i included each potential independent variable and the AIVs. The algorithm calculated the PCP of each M2, and one M2i's calculated PCP was greater than M1, all independent variables in that M2i and its PCP were stored. This was repeated for all remaining independent variables, and the independent variable in the M2i that featured the highest PCP was added to the AIVs, allowing it to remain for the second loop, which controlled the number of times independent variables could be added to the AIVs. This second loop contained ten iterations, but its length was arbitrary, as the addition of independent variables eventually failed to improve the PCP, and after an iteration of the second loop without improvement, the algorithm ended. Thus, by allowing each independent variable to become part of the model, the algorithm allowed both for the selection of the most appropriate independent variables among a large data set such as the one provided to the campaign.

A Summary of the BeSiVa Algorithm

1. Manually Delete Irrelevant Columns
2. Subset the training data into two sets
    1. Training set X1, used in creating models
    2. Pseudotest set X2, used to test models
3. Designate Client Identified Always Independent Variables (AIVs)
4. Create M1
    1. Regress Dependent variable against AIVs using X1
    2. If no AIVs are designated, M1 uses the first independent variable in the set
5. Calculate PCP(M1) on X2
6. Create M2is
    1. Add each possible independent IVi variable to M1

```
7. Calculate PCP(M2i) on X2
8. if PCP(M2i) > PCP(M1),
    1. Add IVi to AIVs
    2. Save PCP(M2i)
    3. Go to step 4
9. if PCP(M1) > PCP(M2i) or steps 4-8 have iterated 10 times, save M1 and end
```

Limitations of BeSiVa's initial formulation

Despite its promise in making predictions and selecting the most relevant independent variables, the formulation of BeSiVa is limited due to a number of potential issues arising from the data and the algorithm. These issues come from the fact that BeSiVa treats every regression as a fresh problem. While the algorithm allows for continuity between selected variables, keeping independent variables that maximize PCP in the model, it also calculates an entirely new regression for each combination of independent variables considered and new data used. This formulation of BeSiVa provides no continuity between predictor coefficients, allowing their values to fall freely. While this means that coefficient values are consistent with a logistic model, guaranteeing that the results are easily interpretable, the lack of continuity is problematic both for computational and algorithmic reasons. When using the algorithm, the free assignment of variables proves computationally expensive, slowing the algorithm unnecessarily. Conceptually, there is a preference for continuity among coefficients, as the results demonstrably affect the overall fit created by the algorithm.

Giving the algorithm free choice of coefficient values proves computationally expensive, and the conceptual aspect of treating each regression as a new problem makes BeSiVa a variant of the stepwise selection methods. BeSiVa faces the same challenges of all stepwise and best subset selection methods, including multicollinearity. A value may have no theoretical relationship with the dependent variable but can predict the dependent variable due to collinearity with a relevant independent variable. If there is a relationship between two independent variables, one that relates both theoretically and has a statistically significant relationship with the dependent variable and one that does not, their inclusion creates a situation where a variable with no theoretical basis for inclusion is placed in the model. This issue with collinearity causes ordering to be an issue for the algorithm, suggesting that if an Irrelevant independent variable that predicts the dependent variable well due to correlation with another theoretically viable independent variable, then the irrelevant variable may take over some of its viable correlate's value, and possibly prevent the correct variable from appearing in the model. This however, would require perfect correlation, as any improvement in the PCP is sufficient to include a variable in the final model. Like other subset selection methods, BeSiVa never substitutes for thinking about the problem, an issue Harrell raises with stepwise and subset selection methods (1998), but it can allow a researcher to input data and build models that can make predictions on a new dataset with the same independent variables. Using BeSiVa to enhance theory, however, runs afoul of its original purpose, predicting responses given a small number of observations with an included response.

When looking at the the formulation of BeSiVa, a possibility arises that variables are included that do not improve the final model, especially in the AIVs. When running the data for the Wakefield campaign, the campaign manager demanded a categorization of age that didn't necessarily improve the fits. This was mitigated, however, as included variables that weren't part of the AIV set needed to improve the performance of the designated pseudotest set, which was deliberately separated from training data to test around irrelevant independent variables. Despite the potential inclusion of irrelevant variables in the final model, BeSiVa mitigates their effects by design, due to the pseudotest set, providing the best fit of the data based on the pseudotest PCP.

In addition to the problem of irrelevant AIVs, the BeSiVa algorithm can run into trouble due to its treatment of missing values and reliance on the way an independent variable interacts with the PCP. While functions like glmnet fail when introduced to data with missing values and must rely on imputation (Hastie et al. 2009), BeSiVa handles the values by listwise deletion of included independent variables, making it at least as tolerant of missing values as the other methods, but this tolerance represents a low bar. BeSiVa can still run into issues due to the lack of a consideration of how listwise deletion can affect data.

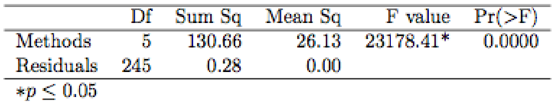
Listwise deletion is problematic due to the potential for biased coefficients (King et al. 2001), but BeSiVa's formulation creates a unique potential problem for the wrong combination of independent variables. Suppose the data run through the algorithm included a variable with a high number of missing values, but the values remaining on included variables after listwise deletion predicted the data well. BeSiVa would run, but its current incarnation risks using a small subset of the data if that subset's PCP is higher than any other used to create a model. In practice, the missing data wasn't a major factor due to preprocessing the independent variables and the use of imputed data during analysis, but it presents a potential effect on larger datasets with greater counts of missing variables. This problem, however, must be considered in context. Without exception, all comparable algorithms require parts of the data or the entire set to be complete or experience massive drops in predictive power, as was the case with random forest and Adaboost.M1 algorithms or total failure as seen with the implementation of penalized regressions, and all comparisons have attempted to meet this minimum requirement through the mice package's imputation by chained equations (van Buuren and Groothius-Oudshoorn 2011), meaning that while a column of missing data can be problematic for the algorithm, its handling of missing values is still an improvement on the other methods used for comparison.

Comparison and Results

To determine the quality of BeSiVa's fitting using a single dataset, the method was compared against several others. This was done by writing code that served as an evaluative workbench for each method. The code uses a bootstrapping iterative process and a single set of imputed data, so that every method could function and would use the same data. The program split the imputed data using a random seed to determine the data subsets that would be used for modeling and testing, and calculated a final percent correctly predicted on a separate test set for BeSiVa, the elastic net and lasso penalized regression methods, the Adaboost.M1 algorithm, and the random forest. Gathering 50 PCP values on the bootstrapped test sets from each method and storing them in a matrix, I then took the values of these PCPs and converted them into a single column of data, regressing the models' numeric PCPs against a categorical variable containing the name of the regression used for each PCP to determine whether BeSiVa's performance differed from the other methods. If BeSiVa could make predictions with comparable PCPs to the other methods, then there should be no statistically significant difference between BeSiVa and them. Thus, a null hypothesis that the average of all values were the same is considered using ANOVA to make the comparison.

The results of the analysis of variance for the bootstrapped PCPS and the linear regression used to create it appears in Tables 1 and 2, demonstrating the performance of BeSiVa compared the other methods. In Table 1 below, the ANOVA data is displayed, showing that there is a difference in population means, demonstrated by the statistically significant value of the F test.

Table 1: Anova Results on the Mean of each method. This table, created using the xtable package (Dahl 2014), displays the result of ANOVA. The F tests has a large enough result that the p value rounds to zero This indicates that the means are different, allowing for the rejection of the Null Hypothesis.

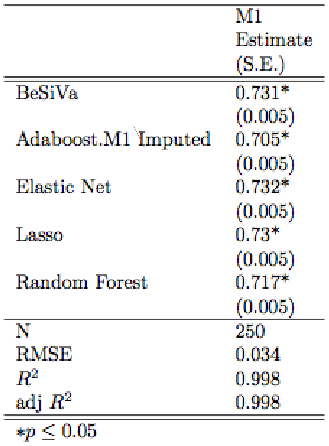


	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Methods	5	130.66	26.13	23178.41*	0.0000
Residuals	245	0.28	0.00		

$*p \leq 0.05$

Tables 2 and 3, however, demonstrate that while this is the case, it stems from the underperformance of the Adaboost.M1, and random forest algorithms. In table 2, the Adaboost.M1 and random forest have PCPs that both have a statistically significant but smaller estimate than the BeSiVa algorithm, Elastic Net, and Lasso regressions, indicating that the BeSiVa outperforms Adaboost.M1 and random forest on the data with 95% confidence. This is reinforced in table 3, which shows that the elastic net and lasso techniques also have comparable coefficients to BeSiVa. Their p values do not approach statistical significance in table 3, and, Figure 2 shows that their mode falls below that of BeSiVa, suggesting that the overall performance of the penalized regressions and BeSiVa are comparable, but that the Adaboost.M1 and random forest algorithms perform worse for the imputed data. Thus, using the same data, the BeSiVa algorithm creates a set of predictions on the test set that when compared to other methods using bootstrapping, indicates that the BeSiVa algorithm creates a competitive fit with preexisting methods for predicting outcomes.

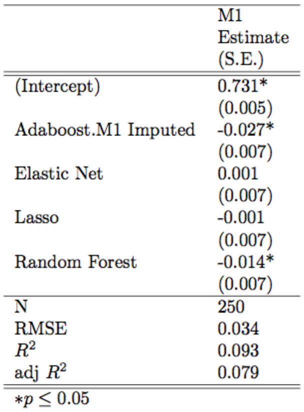
Table 2: A comparison of means. This table, created using the rockchalk package (Johnson 2013), displays the result of the linear regression of the comparison behind ANOVA. While it shows the different means, it appears that the failure of the Null results from the decreased values of Adaboost.M1 and the Random Forest.



	M1 Estimate (S.E.)
BeSiVa	0.731*
	(0.005)
Adaboost.M1 Imputed	0.705*
	(0.005)
Elastic Net	0.732*
	(0.005)
Lasso	0.73*
	(0.005)
Random Forest	0.717*
	(0.005)
N	250
RMSE	0.034
R^2	0.998
adj R^2	0.998

$*p \leq 0.05$

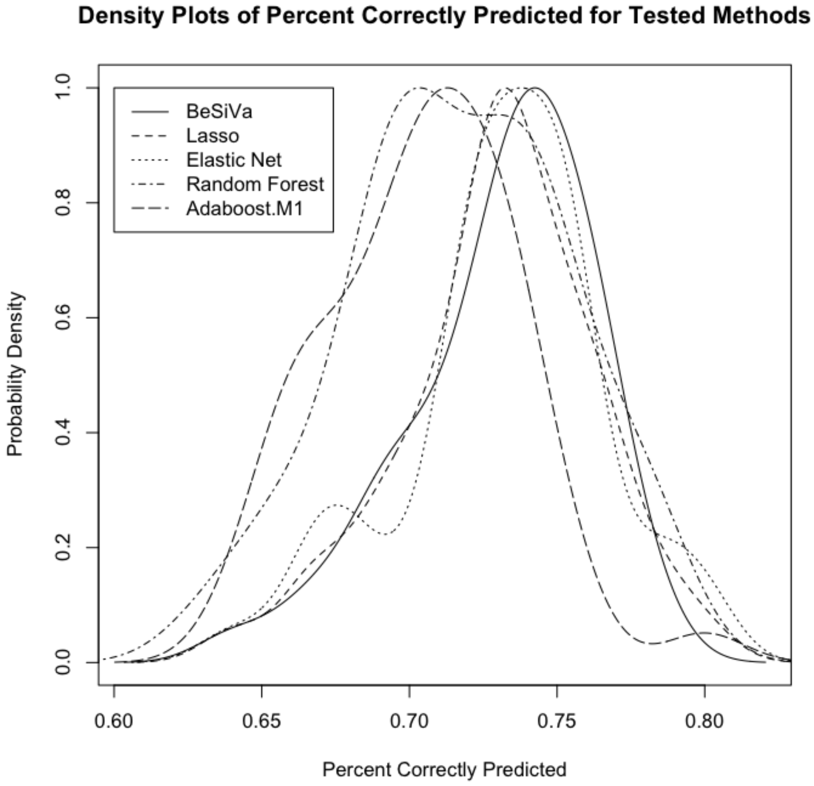
Table 3: The treatment contrasts of means. The failure of the null results directly from the decreased values of Adaboost.M1 and the Random Forest, while Lasso and Elastic Net display comparable estimates to the BeSiVa algorithm.



	M1 Estimate (S.E.)
(Intercept)	0.731*
	(0.005)
Adaboost.M1 Imputed	-0.027*
	(0.007)
Elastic Net	0.001
	(0.007)
Lasso	-0.001
	(0.007)
Random Forest	-0.014*
	(0.007)
N	250
RMSE	0.034
R^2	0.093
adj R^2	0.079

$*p \leq 0.05$

Figure 2: Overlaid density plots of the bootstrapped sample predictions. Each of the methods was given a randomly subdivided set of training data, which was used to predict test data. The resulting superimposed plots can be seen below (R Core Team 2014).



Future Considerations

The BeSiVa algorithm is capable of providing a fit that competes with other methods considered, but its current incarnation needs improvement before working with other data. To simplify data analysis with BeSivA, the algorithm needs to be transformed into a function, moving the focus away from the code that constitutes the algorithm and towards an easier comparison of the algorithm with similar methods. Such a function would use BeSiVa's initial formulation as a default, but would also have options that allow for different evaluations of the data provided. The BeSiVa function would include arguments to shape the pseudo-test set, setting its size as a variable subset of the data with a default size of ten percent. The use of the pseudotest set would also be extended by a cross validation approach, creating an arbitrary multiple pseudotest sets, getting the PCP from each set, and including the variables whose cross-validated PCPs held properties decided by the user, such as the highest mean or median value. Such variation in the pseudotest set enables a more rigorous consideration of multiple independent variables using methods recommended by Kuhn and Johnson for choosing between models (2013). In addition to making the PCP selection more robust via cross-validation, the algorithm could also allow users to choose a more sophisticated means of determining which variables should always be included in the model, instead of the list of AIVs. Such options might include a shuffled variable ordering based on a random seed so that the included variables cannot be affected by initial column order. Another possibility would be to create a priority ranked ordering of variables, where independent variables would be ordered according to the modelers beliefs on relevance, allowing the variables most capable of predicting the model to be included while still considering a set of preferred independent variables more heavily. With these improvements, BeSiVa can be considered against not only a greater collection of methods, but can be used with a more versatile collection of data to further understanding of when it is preferable to the other methods.

In addition to improving and adding onto the BeSivA algorithm, future work demands the further consideration of the other methods for predicting voter choice and the appropriateness of each method. While the bootstrapped resampling demonstrates that BeSiVa creates a competitive fit with other methods, it fails to explain the underlying reason for the variation in the PCP values, included in the appendix as figure 2. Using simulated data rather, future work will also concentrate on understanding exactly when each of the described methods are most appropriate for prediction, giving modelers and practitioners a better sense of when to consider and use each of the methods.

Conclusion

While numerous methods exist for the classification of a choice based on survey data, none were readily apparent to me when I was working to analyze the data collected for the Wakefield campaign. In order to solve the problem of predicting voter support, I created the BeSiVa algorithm, which divides included data into a training and a pseudo-test set. It then creates individual models with a preselected set of independent variables and one independent variable from a set of potential candidates. The algorithm then decides whether to include or disregard a specific independent variable based on whether its inclusion increases the percent of observations correctly predicted in the pseudo-test set. If an independent variable increases this value more than any other in the collection of independent variables, it is placed into the list of independent variables to be included the final model. This process is repeated until no independent variable's inclusion increases the PCP. The method was compared against four other potential classifying methods: elastic net and lasso penalized regressions, random forests, and the Adaboost.M1 algorithm. Comparisons, drawn using a bootstrapping method allowed for the determination that BeSiVa competed with the penalized regressions, and outperformed the random forest and AdaBoost.M1 methods for classifying observations in the specified dataset.

Appendix

Figure 3: The division of counties from which each sample was drawn. Leavenworth, Shawnee, and Douglas Counties each had a random sample proportionate to the size of the county population drawn, while the Southeast and Northeast Regions each had a sample drawn proportionate to their entire population. This map was created using the maps and ggplot2 packages in R (Becker et al. 2014, Wickham 2009)

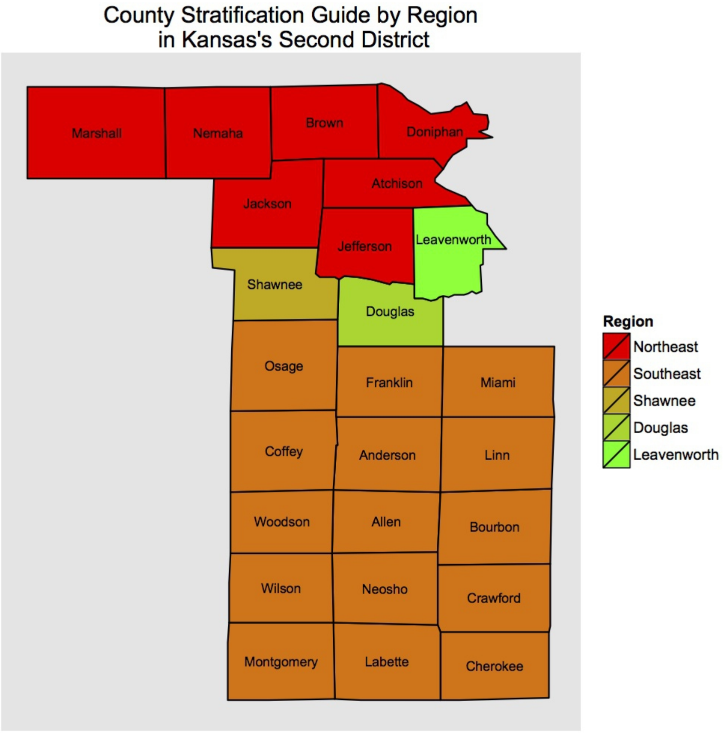
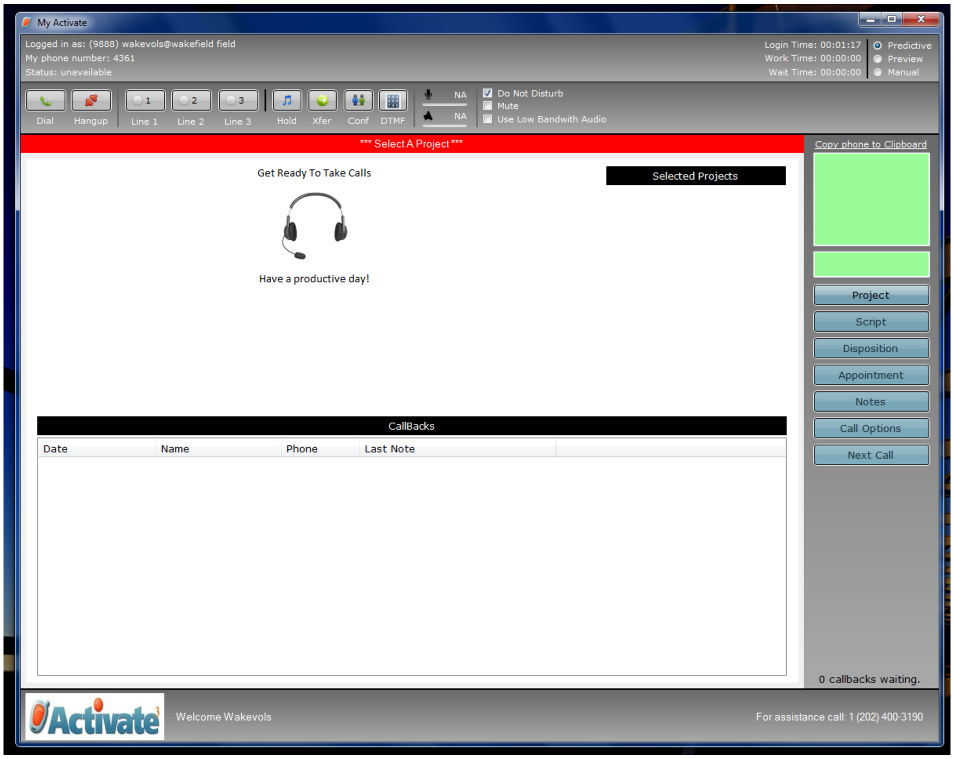


Figure 4: A picture of Activate's predictive dialer, used by volunteers for collecting surveys. The white space in the Screen displays the survey when a voter picks up the phone is dialed, and the software allows volunteers to record responses via menus added to the survey.



Logged in as: (9888) wakevols@wakefield field
My phone number: 4361
Status: unavailable

Login Time: 00:01:17
Work Time: 00:00:00
Wait Time: 00:00:00

● Predictive
○ Preview
○ Manual

NA
NA

☑ Do Not Disturb
□ Mute
□ Use Low Bandwith Audio

*** Select A Project ***

Get Ready To Take Calls

Have a productive day!

0 callbacks waiting.

Welcome Wakevols

For assistance call: 1 (202) 400-3190

Surveying Methodology

The dependent variable was a response to a survey question, collected in the 2014 House race in Kansas' Second District by survey workers associated with the political campaign of the Democratic candidate, Margie Wakefield. The survey was collected through the use of Activate LLC's predictive and preview dialing software, for cell and landline phones. Due to the illegality of using predictive dialing software to contact cell phone records, survey workers used preview dialing to collect surveys from voters who could only be reached at their cell phones.

Software for predictive and preview dialing requires three things to make calls: a valid phone number (which people being called must see for legal reasons), a survey or script for survey workers to read when a call is made, and a list of people to call. The list for the calling portion came from a stratified random sample of voters, pre-selected from the full list of voters in the Second District proportionally based on the number of people in five regions of the district based on population and political import. These regions included Leavenworth, Shawnee, Douglas, and Counties in the northeast and southeast sections of the Second District, with the regions pictured above as Figure 3. I drew this data from a state level voter database curated by the Kansas Democratic party, selecting records based on the proportions of the population in each region. Information on these individuals including name, phone number, and the voter database's identifier of a unique record were loaded into the predictive dialer to allow the voter to be called.

When a volunteer uses Activate's predictive dialer to collect surveys, the software, pictured in figure 4, presents an interface which lets the volunteer make calls, starting by selecting a calling project. Once a volunteer selects the project, they uncheck the box labeled 'Do Not Disturb' to start making calls. Unchecking 'Do Not Disturb' causes the predictive dialer to begin calling numbers on land line phones, automatically screening out answering machines, numbers for which service was unavailable, and phones that were allowed to ring but had no answering machine attached. Due to the desire for efficiency in calling, multiple calls are made at once, attempting to find a number of calls that maximizes the amount of time survey workers spend talking to voters, while also preventing too many voters from being called at once. If too many voters are called, then there will be more voters picking up the phone than there are survey workers to take their calls, leading to dropped or abandoned calls, calls where voters are called and hang up before being connected to a volunteer. Once a potential voter picked up the phone, the volunteer saw the survey display in the dialer software, as well as the voter's name and gender for the purpose of verifying identity. Having confirmed the identity of the voter, survey workers read the script, coached specifically to use only the wording that showed up on the screen and not to deviate from it, which was necessary due to their work on persuasive calling projects. The survey workers used drop down boxes included in the script to record each voters' response to a question from a set of options designated by the campaign. For cell phones, the predictive dialer interface did not change, but parts of the interface specific to cell phones were used, altering how the volunteer used the software. To comply with the law, the software called cell phones in preview mode. Using Activate's software in preview mode requires a volunteer to hit Next Call to bring up a voter's record and the survey, and then press the Dial button to call that voter. Pressing Dial causes the software to dial the voter's number, and the volunteer then waits for the voter to answer the call, requiring survey workers to deal with answering machines, unanswered phones, and service problems. After a call had finished in both predictive dialing and preview dialing modes, the volunteer needed to leave a brief summary of how each call went, called a disposition, before moving on. Dispositions allow the campaign staff managing calling to get a better sense of how calls are going, and they direct the software on whether to call a voter's number again. Once a record is dispositioned, the software allows the volunteer to continue calling, either by starting to dial numbers automatically in predictive mode or by hitting Next Call to bring up the next record in preview mode. Once calls were completed, the volunteer checked the Do Not Disturb box again to make themselves unavailable, and exited the program. This software was effective in collecting dependent variable data, the support of voters for a particular position or candidate.

Logistic Regression

The client's primary focus, predicting whether people were likely to support Margie

Wakefield in the 2014 midterm elections lead me to decide that logistic regression was the most appropriate way to model voter support. Logistic regression uses a linear combination of a set of predictors, akin to those seen in ordinary least squares and displayed in equation 7 as η.

$$\eta = \beta_0 + \beta_1 x1_i + \ldots + \beta_p XP_i \tag{7}$$

These predictors are then placed within the logistic link function *m* in equation 8

$$m(\eta) = \frac{e^{\eta}}{1 + e^{\eta}} \tag{8}$$

to determine the probability π_i that a dichotomous observation has a value of 1, as seen in equation 9

$$\pi_i \;=\; m(\beta_0 + \beta_1 x1_i + \ldots + \beta_p XP_i) \tag{9}$$

which simplifies to equation 10

$$\pi_i = \frac{e^{(\beta_0 + \beta_1 X1_i + \ldots + \beta_p XP_i)}}{1 + e^{(\beta_0 + \beta_1 X1_i + \ldots + \beta_p XP_i)}} \tag{10}$$

Probabilities predicted by the logistic regression fall between 0 and 1 (Verzani 2005), and their values were used throughout the project to determine likely voter support and independent variable inclusion.

In addition to helping the client determine likely supporters, the probabilities created by the logistic regression were incorporated into the the BeSiVa Algorithm's percent correctly predicted calculation. Due to logistic regression's prediction of probabilities between 0 and 1, the PCP is defined as the percent of predictions less than 0.50 for observations of nonsupport, coded as 0 and predictions greater than or equal to 0.50 for observations of support coded as 1. This count of observations was divided by total number of responses to create the PCP. The calculated PCP for all observations in X2, the data held out for prediction in the algorithm, was the determining factor on whether an independent variable was included in the BeSiVa's final model.

Sample Questions

Issue Question

Lynn Jenkins believes that we need to continue to explore and expand oil, gas and coal to meet our nation's energy needs and strengthen our energy security. She believes that innovations in fracking technology have vastly improved our ability to extract oil and gas and will lead to greater energy independence and jobs. She favors deregulation of fracking, opening national parks to further oil and gas exploration and expansion of pipelines including the Keystone XL pipeline. She does not believe that expansion of alternative energy like wind energy is a viable option to our energy needs.

Margie Wakefield believes that an "all of the above" energy policy is preferable and that we should be promoting investment in a national energy infrastructure to facilitate the rapid growth of the Kansas wind energy industry. During just the past few years, this industry has generated five billion dollars in revenues and over 4,000 high-wage jobs in areas of the state that badly need economic growth. She believes that fracking technology needs to be closely monitored for environmental reasons as well as concern about an alarming increase in the frequency of earthquakes in Northern Oklahoma and South-Central Kansas where fracking has recently increased dramatically.

Knowing that everything is not always black and white and that you might agree with some parts of either statement, can you tell me on a scale of 1 to 10, 1 meaning you agree strongly with Lynn Jenkins position and 10 you agree strongly with Margie Wakefield's position, which statement is closer to your own belief?

Final Identification Question:

Sometimes in a survey like this people change their minds about who they might support. If the election were held today for U. S Congress, can you tell me on a scale of 1 to 10, 1 being more likely to support Lynn Jenkins and 10 being more likely to support Margie Wakefield, who you would be more likely to support?

References

Alfaro, Estefan, Matias Gamez, and Noelia Garcia. 2013. adabag: An R Package for Classification with Boosting and Bagging. Journal of Statistical Software, 54(2), 1-35. http://www.jstatsoft.org/v54/i02/.

Becker, Richard, Allan R. Wilks, Ray Brownrigg, and Thomas P. Minka. 2014. maps: Draw Geographical Maps. R package version 2.3-7. http://CRAN.R-project.org/package=maps

Dahl, David B. 2014. “xtable: Export tables to LaTeX or HTML”. R package version 1.7-3. http://CRAN.R-project.org/package=xtable

Friedman, Jerome, Trevor Hastie, Robert Tibshirani. 2010. Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1-22. http://www.jstatsoft.org/v33/i01/.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning with Applications in R.* New York: Springer.

King, Gary, James Honaker, Anne Joseph, and Kenneth Scheve. 2001. “Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation.” American Political Science Review. 95(March): 49-69.

Kuhn, Max and Kjell Johnson. 2013. *Applied Predictive Modeling.* New York: Springer.

Harrell, Frank. 1998. “What are some of the problems with stepwise regression?” May 1998. http://www.stata.com/support/faqs/statistics/stepwise-regression-problems/ (November 10, 2014)

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements Of Statistical Learning: Data Mining, Inference, and Prediction.* 2nd ed. New York: Springer.

Johnson, Paul E. 2013. rockchalk: Regression Estimation and Presentation. R package version 1.8.0. http://CRAN.R-project.org/package=rockchalk

Liaw, A. and M. Wiener. 2002. Classification and Regression by randomForest. R News 2(3), 18-22.

R Core Team. 2014. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

van Buuren, Stef, Karin Groothuis-Oudshoorn. 2011. mice: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software, 45(3), 1-67. http://www.jstatsoft.org/v45/i03/.

Verzani, John. 2005. *Using R for Introductory Statistics.* New York: Chapman & Hall/CRC

Wickham, H. 2009. *ggplot2: elegant graphics for data analysis.* New York: Springer.

Zou, Hui and Trevor Hastie. 2005. “Regularization and variable selection via the elastic net”
Journal of the Royal Statistical Society B. 67:301–320.