Performance of 2-class Classifiers on Data for which Labels are Missing by a Non-Random Mechanism

Amy Crawford

Spring 2017

Abstract

When companies implement algorithms to expedite existing job functions it often leaves them with newly available employees, time, and money to investigate other unexplored parts of their market. To be as efficient as possible in an expansion such as this, the company can again turn to some machine learning algorithm to point to where the newly available resources should be focused in the market. Unfortunately, the only available data to train an algorithm on would then be data from the portion of the market that has already been explored by the company. In other words, whether a case in the data set would have a label or not is due to a specific non-random mechanism. I evaluate the performance of logistic regression, **s**upport **v**ector **m**achine (SVM), and random forest models under this framework. Additionally, I implement one classification method called **a**ctive **s**et **s**election **c**lassification (ASSC) that involves weighted bootstrap and emsemble learning.

Introduction

Overview of the Product

A large company manufactures small and large equipment for both personal and commercial use. I will call it Company X. Founded by one individual in the 1800s, this Fortune 100 company has grown to serve over 100 countries. The company has a strong distribution channel and relies heavily on its many dealerships to connect to consumers.

A financial division of Company X provides financing options for equipment and industry inputs such as fuel, seeds, and fertilizers. Financing options are provided to all types of consumers including individuals, companies, and government entities. The consumer can choose to finance with an installment loan, a lease, or a revolving credit product. My project with the company focused on the revolving credit product. This product functions like a credit card and is used exclusively for industry inputs. Account information is held at the consumer's home dealership and the consumer can puchase industry inputs using the revolving credit product financed by Company X directly through the dealership.

Unfortunately, there are consumers and dealers who misuse the revolving credit product. The financial division of the company is responsible for detecting and handling these dishonest individuals. One type of the product misuse comes in the form of the consumer attempting to obtain a cash advance. In this scenario, the consumer and dealer (usually) work together to submit a transaction that overstates the price or amount of industry input purchased. For example, the consumer's equipment might only hold 200 gallons of fuel but the dealer submits a transaction for 500 gallons of fuel on the consumer's revolving credit product account. This is particularly unsavory for Company X because the payment plans for the revolving credit product are necessarily very flexible. The consumers who use this financing option must often spend large sums of money on inputs to produce their product, but don't have the cash flow to repay Company X until their product is mature enough to sell. Because of the nature of this industry the revolving credit product has a payment schedule that requires large payments from the consumer during the season of high cash flow and much smaller payments from the consumer during seasons of low cash flow. Thus, when the consumer misuses the revolving credit product with the intention of receiving a cash advance, the company takes on a sizeable amount of unnecessary risk, usually during a season of low cash flow.

Another type of misuse occurs when the consumer is indebted to the dealer and the dealer shifts the debt (and risk) to Company X by way of submitting a transaction on the revolving credit product of the consumer. When a transaction is placed with the intent of misusing the revolving credit product, it is labeled as a fraudulent transaction. Since account information for this revolving product is held within a dealership, Company X experiences little of what the reader might immediately think of as traditional credit card fraud. Nevertheless, any misuse of this revolving credit product is considered to be an attempt to defraud the company and is treated as such.

Overview of the Problem

When I joined Company X as an intern in May 2016, the financial division had resources to audit a small subset of transactions made in the United States on the revolving credit product. The group in charge of this task, which we will refer to as the audit group, was auditing all transactions that they considered to be "high risk". Any transaction that was audited by the group was subsequently labeled according to whether it was found to be fraudulent or not.

While all transactions under consideration had large purchase amounts (> \$3,000), a transaction was considered to be of "high risk" to the company if it was made on an account with a high existing balance (> \$10,000) that was in a state of non-payment (> 60 days past payment due date). These account thresholds were chosen strategically because it was beleived that a higher rate of fraud existed among these "high risk" transactions. Additionally, the company stood to incur significant losses if transactions of this nature were found to be fraudulent. My first task as an intern was to use data available from past audits and build a model to rank incoming sets of "high risk" transactions in order of probability of fraud. Consequently, the audit group would be able to audit transactions in the order of the ranked list, and catch almost all of the fraudulent transactions in a shorter period.

With implementation of the model built to rank the "high risk" transactions, the audit group anticipated having newly available time and resources. With these resources, the audit team wanted to expand to auditing a wider range of transactions. Thus, my second task was to rank all incoming transactions with large purchase amounts ($>$ \$3,000), not just those that were considered "high risk", in order of probability of fraud. This more general pool of transactions were charged on accounts with any amount of existing balance that were not necessarily in a state of non-payment. This second task poses many challenges and leads to the problem of interest for this project. That is, how can we execute a 2-class classification or estimate fraud probability for transactions in the entire sample space if we only have labels for the "high risk" transactions, since those were the only transactions audited?

When trying to build a classifier for the rest of the transactions we find ourselves in a situation where data cases are missing labels not at random, but by a very specific covariate-dependent mechanism. While some may simply view this as a severe case of extrapolation, it is a real scenario that arises in business settings when companies aquire additional resources to expand their operations. The ability to identify the next set of transactions to spend resources on is extremely valueable to Company X. Even a classifier that we know will perform relatively poorly can save a great deal of money in a situation such as this.

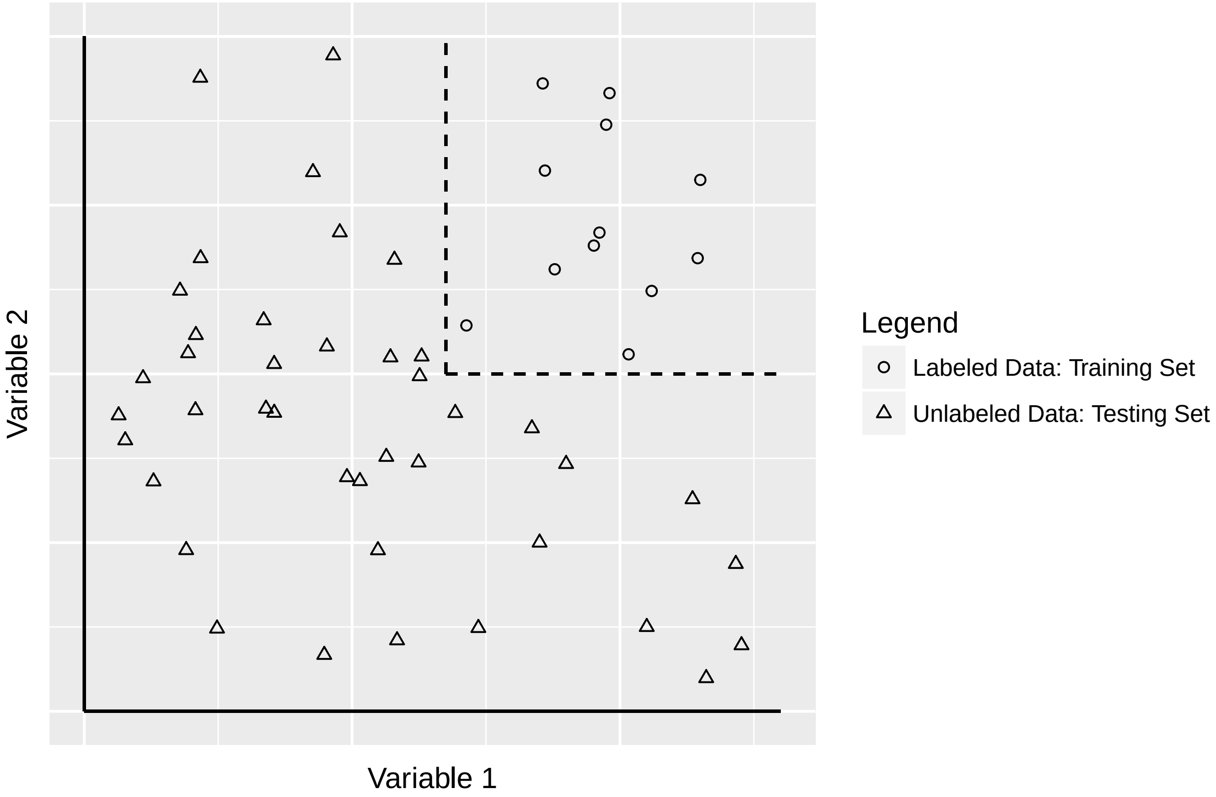
Data

Company X Data

From this point forward refer to data cases that are labeled as belonging to the training set. Similarly, data cases that need to be classified (those that are not labeled) will belong to the testing set. Since the mechanism dictating which data cases are labeled enforces strict thresholds on two variables (account balance > \$10,000 and state of non-payment > 60 days past due) we can think of the cases in the training set to be confined to a "corner" of the feature space. See Figure 1 for a simplified graphical representation of this in 2-dimensions.

The goal of this project is to investigate the best way to conduct 2-class classification of data cases that lie outside of a corner of a feature space where a classifier is built. The Company X dataset poses two major concerns. The first is that since fraud occurs at such a low rate, the data labels (in the entire data set) are not at all balanced. The second concern comes with the assumption that a higher rate of fraud exists in the training set ("high risk" transactions) than the testing set. This suggests that the distribution of fraud in the training set differs from that in the testing set.

Figure 1



Introducing the Water Pump Data

In the interest of complete demonstration of the methods in the remainder of the paper, the data used will not be the proprietary data from Company X that motivated the project. To allow for proper evaluation of the prediction methods, I proceed using data from a data science competition (DrivenData Inc. 2017). The competition is a 3-class classification problem designed as an excercise for learning and exploration for intermediate level participants. Data were collected on 59,400 water pumps and their corresponding water points, in Tanzania. The pumps are used to draw water from community water points. The primary task of the posted competition is to predict which pumps are functional, which need minor repairs, and which don't work at all. Each pump in the data set is labeled according to these categories.

The water pump data can be configured to be analogous to the motivating problem from

Company X. The water pump data includes information on the year in which the water point was manufactured and the population density around the water point. These are two variables that I will build a "corner" on. Just as the employees at Company X were able to spend resources to label only "high risk" transactions on the revolving credit product, we can consider a similar scenario with the water pump data.

Hypothetically, consider that until very recently the organization responsible for monitoring the status of the water points and pumps had restricted resources of money, time, and employees and could only monitor old wells (built before 1985) that have a relatively large surrounding population (> 200 people). These water points are analogous to the "high risk" set of transactions from Company X. So, the organization has data on features of all pumps, but only pumps on these currently monitored water points are labeled as `functional`, `functional needs repair`, or `non functional`.

Now, suppose the organization has acquired resources to hire a few new employees who will devote their time to monitoring and managing water points with less densely populated surrounding areas. The organization needs to use these few employees in a very strategic way. There are many, many wells that will fall under this group's jurisdiction and they need to target pumps that are likely broken, or need repairs. The organization doesn't want to waste time sending these employees all over the country to check on pumps that are functional. Using what we know about the status of the pumps that have been monitored (the "high risk" pumps), we need to predict which of the relatively new pumps that are located in less densely populated areas will require the attention of the new employees (i.e. the pumps that need repairs, or are broken).

Structural Changes to the Data

The water pump data has 59,400 rows and 41 columns and contains information on various features of the pumps as well as the water points where they reside. Water pumps were originally labeled as `functional`, `functional needs repair`, and `non functional`. First, to parallel the motivating problem from Company X, we will turn this 3-class classification problem into a 2-class classification problem. To do this, consider the pumps labeled as `functional needs repair` and `non functional` together in one group labeled `requires attention`.

There is a variable in the data set, `population`, that provides information about the size of the population surrounding the water point. Nearly 36% of the values in this column are zero. It is clear that many of the zeros simply indicate a missing value, but we acknowledge the possibility that there truly are not any residents residing in the vacinity of a water point. If the latter is the case, we can eliminate the water point from the data set since we do not wish to expend effort classifying a water point that is abandoned. If the former is true, then the zero represents a missing value. After looking for patterns that might help us understand these many missing values, we find that the location of the water point seems relevant.

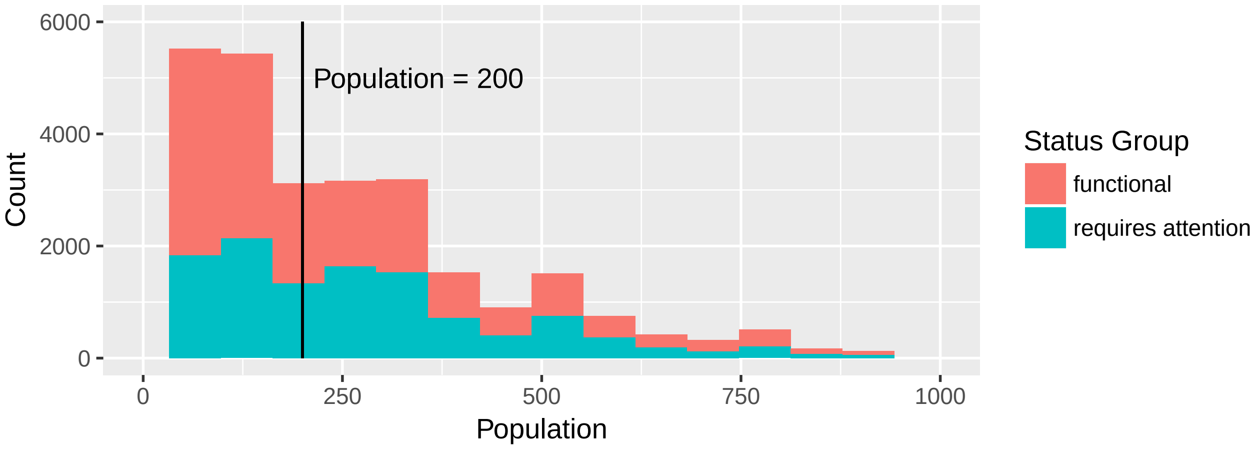
Tanzania is divided into regions and most zero values of the `population` variable are for water points in six of these regions (DrivenData Inc. 2017). Since the focus of this problem is not a spatial analysis and it would not be particularly helpful to classify abandoned water points, we can proceed only considering water points with non-zero values of the `population` variable. Similarly, we will elimate any wells that are missing information about construction year. This effectively reduces the number of rows from 59,400 to 37,344. The map in left panel of Figure 2 displays all 59,400 water points in the original data set. The map in the right panel of Figure 2 includes only the water points with valid `population` and `construction_year` data. Both maps are colored by the classification of the water pump status. Maps were generated using shape files from the GADM spatial database (Hijmans 2017).

Building the "Corner"

Recall that in the motivating problem from Company X, there was assumed to be a higher rate of fraud among the training cases. In order to be consistent with this motivating problem consider choosing water points to be members of the "cornered" training set based upon threshold values of the variables `population` and `construction_year`. The `population` variable discussed above ranges from 1 to 30,500. Water points with larger surrounding populations have a higher proportion of `requires attention` pumps than water points located in areas with fewer people. Figure 3 shows this graphically with a vertical line placed at the population level of 200 people. There are a relatively small number of wells with a surrounding population of more than 1000 people, so the histogram is restricted to population values of 900 or less for presentation purposes. Now, the first requirement for membership in the training data set is a `population` larger than 200 people.

Figure 3

Truncated Histogram of the Population Variable

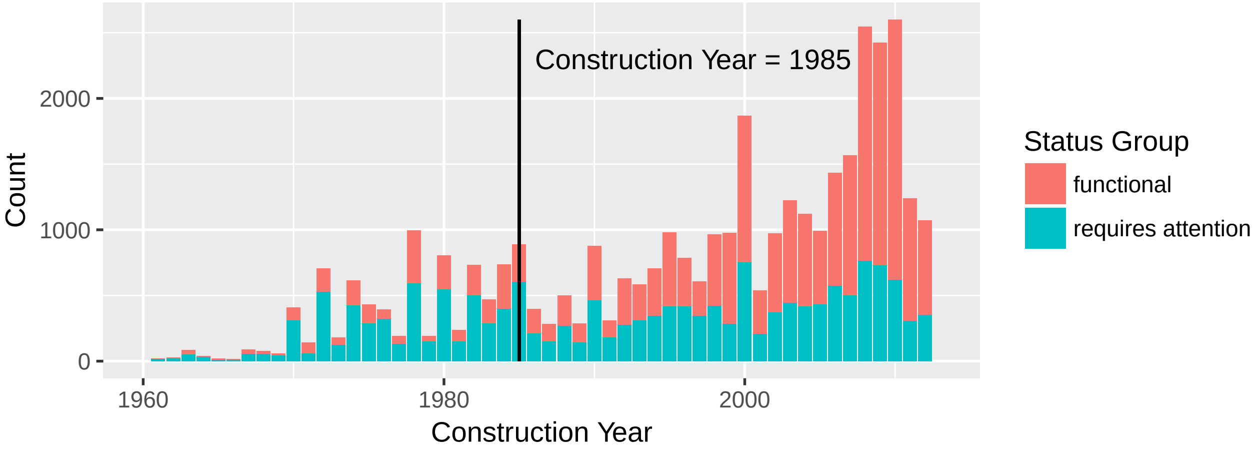


The variable `construction_year` is simply the year in which water point was constructed. The variable ranges from 1960 to 2013. Visually, it is clear from Figure 4 that water points built before 1985 have a higher ratio of pumps that `require attention` to those that are `functional` than pumps built after 1985 (and certainly higher than those built after 1990).

Thus, the second requirement for membership in the training data set is a `construction_year` before 1985.

Figure 4

Histogram of the Construction Year Variable



Now, the data have been divided into the training and testing sets based on the thresholds for the `population` and `construction_year` variables prescribed above. The training data has 2,886 rows and the testing data has 34,458 rows. The distribution of the target variable `status_group` in each of these data sets is summarized in Table 1.

Table 1

Distribution of the Status Group Variable in Training and Testing Data Sets

	Status Group (row %)		
Data Set	"functional"	"requires attention"	Total
Training Data	939 (32.54%)	1,947 (67.46%)	2,886 (100%)
Testing Data	19,770 (57.37%)	14,688 (42.63%)	34,458 (100%)

Because broken pumps are not quite such a rare event as fraud on a revolving credit product, the Tanzania water pump data are not unbalanced like the data from Company X.

We have, however, constructed a training set including only water points with `population` larger than 200 and `construction_year` before 1985 effectively forcing the training data into a "corner" of the feature space. The Company X data was believed to have a higher rate of fraud among the "cornered" training data cases than the testing data cases. Using the chosen thresholds to split the Tanzania water point data, we are able to mirror this phenomenon. Now, labels on the status of the water pumpts are removed from the testing dataset and we proceed to select features and model as if we never had them. They will be used only to evaluate model performance.

Feature Selection

There are many groups of variables in the data that are meant to give the same information at different levels of detail. For example, the variables `source`, `source_type`, and `source_class` all provide information about the water source feeding the water point. These variables are all coded as factors. The `source_class` variable gives very general, high level information while `source_type` is a bit less general, giving a moderate amount of detail. The `source` variable is the most detailed of the three. Table 2 shows the unique values each of these variables can take. These variables are highly correlated and will clearly introduce multicollinearity during the modeling process. For variable groups like these, the one providing what we judge to be the most appropriate level of detail is chosen using visual comparison and numerical summaries. For the three variables discussed here, `source_type` was chosen as the most appropriate to inform the subsequent models about the water source feeding the water point. A bar plot of the `source_type` variable, colored by the target variable, `status_group`, in the training data can be found in Figure 5.

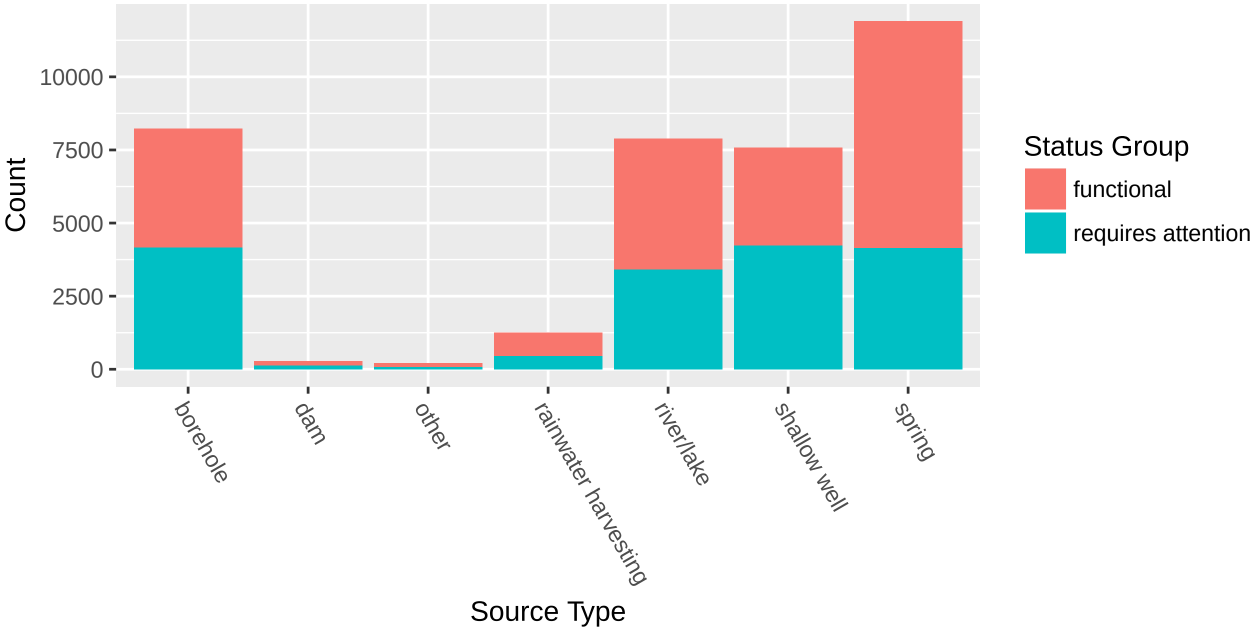
Table 2

Unique Values in each of the Variables that Provide Water Source Information

source	source_type	source_class
dam	borehole	groundwater
hand dtw	dam	surface
lake	other	unknown
machine dbh	rainwater harvesting	
other	river/lake	
rainwater harvesting	shallow well	
river	spring	
shallow well		
spring		
unknown		

Figure 5

Bar Plot of the Source Type Variable Colored by Status Group



There are variables in the data set that appear to be manually maintained and have too many unique values to be managable. These variables were eliminated as well as any variables with only one unique value. All variables, except basin, providing information on the location of the water point were eliminated and few other variables were excluded for lack of predictive power. Basin was kept because it gives information on the geographic location of the water points with regard to the water basin they pull from. The data began with 40 feature columns (excluding the target variable, status_group). After feature selection, the resulting data set has 15 feature columns and the target variable. Since the population and construction_year variables were used to create the split between training and testing data sets it will be interesting to fit models with and without these variables to compare performance. Table 3 is a compilation of the final 15 features, their descriptions, and values that each can take.

Table 3

Final Set of Predictor Variables

Variable Name	Values	Description
*population**	Numeric, (1, 30500)	Population around the well
*construction_year**	Numeric, (1960, 2013)	Year the waterpoint was constructed
amount_tsh	Numeric, (0, 200000)	Total static head amount (amount of water available to the water point)
gps_height	Numeric, (-90.0, 2232.0)	Altitude of the well
basin	Factor, 9 levels	Geographic water basin the water point draws from
public_meeting	Factor, True/False/Unknown	Was there a public meeting held?
scheme_management	Factor, 11 levels	Who operates the water point
permit	Factor, True/False/Unknown	If the water point is permitted
extraction_type_class	Factor, 7 levels	The type of extraction the water point uses
management_group	Factor, 5 levels	The type of entity that manages the water point
payment	Factor, 7 levels	When do users have to pay to use the water point?
quality_group	Factor, 6 levels	A measure of water quality at the water point
quantity	Factor, 5 levels	A measure of whether the quantity of water available from the water point is sufficient
source_type	Factor, 7 levels	The type of source the water comes from
waterpoint_type	Factor, 7 levels	The type of water point used

**Variables used to create split between training and testing sets.*

Analysis

Four prediction methods are detailed in the following portion of the paper. Each model was built on the “cornered” training data set excluding the `population` and `construction_year` variables (13 predictors) and again with the variables included (15 predictors).

Logistic Regression

Logistic regression has a long history in the field of statistics and is one of the most widely used source of classifiers (James et al. 2013). In general, it is a method of estimating conditional probabilities as a function of explanatory variables and regression coefficients, and is often used to classify binary response data which is how we will apply it here. Two logistic regression models were fit to the training data using the `glm` function in `R`. For logistic regression, categorical variables (stored as `factors` in `R`) are expanded into binary indicator variables. The full equations for the 13 predictor, and 15 predictor models can both be found in the appendix.

Using the 13 predictor and 15 predictor models to classify water pumps in the testing set yields the confusion matrices and statistics in Tables 4 and 5 respectively. The largest advantages of implementing a logistic regression model are interpretability and computational convenience. Since we are extrapolating, the relatively low values of Accuracy, Sensitivity, and Specificity are not surprising. Kuhn and Johnson (2013) write that Kappa values within 0.30 to 0.50 indicate reasonable agreement, so although they are on the low end, we are not so dissatisfied with the Kappa statistics of 0.3703 and 0.3748. 10-Fold cross validation results are reported in Table 10.

Table 4

Confusion Matrix and Statistics for Classification by Logistic Regression with 13 Predictor Variables

		Reference	
		functional	requires attention
Prediction	functional	13138	4206
	requires attention	6632	10482

Kappa :	0.3703
Accuracy :	0.6855
Sensitivity :	0.6645
Specificity :	0.7136

Table 5

Confusion Matrix and Statistics for Classification by Logistic Regression with 15 Predictor Variables

		Reference	
		functional	requires attention
Prediction	functional	13302	4271
	requires attention	6468	10417

Kappa :	0.3748
Accuracy :	0.6883
Sensitivity :	0.6728
Specificity :	0.7092

Support Vector Machine (SVM)

Support **V**ector **M**achines (SVMs) are a more flexible procedure that enlarges the feature space using basis expansions. Linear classifiers generally achieve better training class separation in this enlarged space. These classifiers translate to nonlinear classification boundaries in the original space. The dimension of the enlarged feature space is allowed to get very large (Hastie, Tibshirani, and Friedman 2009).

Two SVM models were fit using the `svm` function with a radial kernal in the `e1071` package in `R`. Cost paramaters for each model were tuned using 10-fold cross validation. The model fit with 13 predictors was fit with a cost parameter of 40 and the model fit with 15 predictors with a cost parameter of 90. Confusion matrices and prediction statistics are included in Tables 6 and 7. We notice that classification using these SVM models yields slightly lower values of Accuracy and Kappa statistics, but the Kappa values both remain above 0.3. 10-Fold cross validation results are in Table 10.

Table 6

Confusion Matrix and Statistics for Classification by Support Vector Machine (SVM) with 13 Predictor Variables

		Reference	
		functional	requires attention
Prediction	functional	12683	4097
	requires attention	7087	10591

Kappa:	0.3533
Accuracy :	0.6754
Sensitivity :	0.6415
Specificity :	0.7211

Table 7

Confusion Matrix and Statistics for Classification by Support Vector Machine (SVM) with 15 Predictor Variables

		Reference	
		functional	requires attention
Prediction	functional	12764	4161
	requires attention	7006	10527

Kappa:	0.3535
Accuracy :	0.6759
Sensitivity :	0.6456
Specificity :	0.7167

Random Forest

Random forest is a tree-based method that can be used for classification. This algorithm builds a number of decision trees on bootstrapped training samples. During building, at each node (or split) of each tree, the algorithm radomly selects a subset of the input features and finds an optimal single split. For each tree, we repeat splitting in this fashion up to a fixed depth or until no single split improvement is possible without creating a split resulting in a small number of cases. Many trees are built in this fashion, then aggregated (James et al. 2013).

Two random forest models were trained using the `randomForest` function from the randomForest package in R. The parameter mtry determines the number of features randomly sampled as candidates at each split. Again using 10-fold cross validation, the model fit with 13 predictor variables was optimized at an mtry value of 3 and the model fit with 15 predictor variables at 4. The resulting confusion matrices and statistics for these two models are in Tables 8 and 9. 10-Fold cross validation results are in Table 10.

Table 8

Confusion Matrix and Statistics for Classification by Random Forest 13 Predictor Variables

		Reference	
		functional	requires attention
Prediction	functional	12533	4097
	requires attention	7237	10591

Kappa:	0.3455
Accuracy :	0.6711
Sensitivity :	0.6339
Specificity :	0.7211

Table 9

Confusion Matrix and Statistics for Classification by Random Forest 15 Predictor Variables

		Reference	
		functional	requires attention
Prediction	functional	13083	4373
	requires attention	6687	10315

Kappa:	0.3568
Accuracy :	0.6790
Sensitivity :	0.6618
Specificity :	0.7023

Active Set Selection Classification (ASSM)

Dataset shift is a common problem in predictive analytics that presents itself in many practical applications. This phenomenon occurs when the joint distribution of inputs and outputs differs between training and testing stages of predictive modeling. It may be present for a variety of reasons such as the inability to reproduce testing conditions during training, or because experimental design introduces bias. In the real world, it is often the case that the conditions under which we use the models that we develop will differ from the conditions in which the model was developed. Environments are generally nonstationary and while "textbook" predictive machine learning algorithms are powerful, they generally ignore these differences (Quiñonero-Candela et al. 2009).

Active **S**et **S**election **C**lassification (ASSM) is the next classification method we will consider. ASSM is introduced by Wen Zhou in Chapter 4 of his doctoral dissertation (Zhou 2014). The chapter discusses the covariate shift problem, which is a particular type of dataset shift where the distribution of feature vectors are possibly different between training and testing sets. By training existing classifiers using subsets of training data that are similar (in some sense) to the training cases, one is able to mitigate a large amount of the effect brought on by the covariate shift problem. ASSM is felixible and can be used with existing methods of classification. Zhou details the ASSM algorithm in Algorithm 4.2.2 of his paper. Here we implement ASSC with an SVM classifier (ASSM-SVM) in the same fashion, substituting our own functions and parameters in the algorithm as follows.

First, lay the notational groundwork used in Algorithm 4.2.2 of Zhou (2014). Consider the training set $\mathcal{T}$ with $n = 2886$ (labeled) observations $\{(Y_i, \mathbf{X}_i)\}_{i=1}^{n}$ where Y_i are the corresponding class labels and $\mathbf{X}_i$ are p-dimensional feature vectors. Recall that for the well data we consider both the cases of $p = 13$ and $p = 15$ (training data excluding and including `population` and `construction_year` variables respectively). Also consider the testing (or predicting) set $\mathcal{P}$ with m (unlabeled) observations $\{\mathbf{U}_j\}_{j=1}^{m}$ which are p-dimensional feature vectors to be classified. The algorithm is extremely computationally expensive, so for demonstrative purposes I will simply consider a random subset of size $m = 300$ from $\mathcal{T}$ to evaluate both .

Step 1.

Compute an **association matrix** $W_{m\times n} := [w_{ji}]_{j=1;i=1}^{m,n}$ that for $X_i = (X_{i1},\ldots,X_{ip})' \in \mathcal{T}$ and $U_j = (U_{j1},\ldots,U_{jp})' \in \mathcal{P}$,

$$w_{ji} = \exp\left\{-\frac{\sum_{k=1}^{p} d_k^2(U_{jk}, X_{ik})}{\lambda}\right\},$$

where λ can be thought of as a tuning parameter, and $d_k(u,x)$ denotes a distance (or dissimilarity) function of u and x on the k^{th} coordinate of the feature vector. For the k_n standardized numeric features use Euclidean distance

$$d_k(u,x) = \sqrt{(u_{jk} - x_{ik})^2}.$$

For the remaining k_c other categorical features use the following weighted degree-of-difference measure to assess dissimilarity for these features (Hastie, Tibshirani, and Friedman 2009)

$$d_k(u,x) = \tau\ \mathrm{I}[u_{jk} \neq x_{ik}] = \tau \begin{cases} 1 & u_{jk} \neq x_{ik} \\ 0 & u_{jk} = x_{ik} \end{cases},$$

where τ is the weight given to a pair of dissimilar categorical features at every coordinate of a categorical variables k.

Let k_n be the number of numeric features and k_c be the number of categorical features in the data set. Then, we compute each $[w_{ji}]_{j=1;i=1}^{m,n}$ as

$$w_{ji} = \exp\left\{-\frac{\sum_{k=1}^{k_n}(u_{jk} - x_{ik})^2 + \tau^2 \sum_{k=k_n+1}^{k_c} \mathrm{I}[u_{jk} \neq x_{ik}]}{\lambda}\right\}.$$

Step 2.

Compute the **selection probability matrix** $P_{m\times n} := [p_{ji}]_{j=1;i=1}^{m,n}$ with

$$p_{ji} = \frac{w_{ji}}{\sum_{i=1}^{n} w_{ji}}.$$

Step 3.

Fix B > 0, and for the b^{th} iteration where $b = 1, \ldots, B$, do the following.

(a.) For each $U_j \in \mathcal{P}(j = 1, \ldots, m)$, sample $q < n$ observations from $\mathcal{T}$ with replacement based on the selection probability that

$$\left(Y_{j,l}^{b}, X_{j,l}^{b}\right) := (Y_i^*, X_i^*)|\mathcal{T} \overset{iid}{\sim} \{p_{ji}\}_{i=1}^{n}$$

for $l = 1, \ldots, q$ (X^* denotes a sampling version of X), so that we obtain a **selected active set** (i.e. a b^{th} training set of size mq for predictions on $\mathcal{P}$)

$$\mathcal{S}_b^q = \left\{\left(Y_{j,l}^{b}, X_{j,l}^{b}\right)\right\}_{j=1,l=1}^{m,q}.$$

(b.) Train an SVM classifier with parameter cost = 10 on the selected active set $\mathcal{S}_b^q$ by which we make predictions $\hat{V}_{j,b}$ for observations in $\mathcal{P}$.

Step 4.

Obtain the predictions for observations in $\mathcal{P}$ by the majority voting that

$$\hat{V}_j^{ASSC} = \arg\max \sum_{b=1}^{B} I\left[\hat{V}_{j,b} = c\right]$$

for $j = 1, \ldots, m$. The collection $\mathcal{S}_B(m, n, q) = \cup_{b=1}^{B} \mathcal{S}_b^q$ is called the **active set** for $\mathcal{T}$ with respect to $\mathcal{P}$.

First, we implement this algorithm using a random sample of size $m = 300$ from $\mathcal{P}$ with 13 predictor variables using parameters $B = 201$, $q = 150$, $\tau = 1.8$, $\lambda = 4$. This process is repeated three times and the averaged Accuracy and Kappa statistics can be found near the bottom of Table 10. Then, we implement this algorithm using a random sample of size $m = 300$ from $\mathcal{P}$ with 15 predictor variables using parameters $B = 201$, $q = 150$, $\tau = 1.5$, $\lambda = 3$. This process is also repeated three times and the averaged Accuracy and Kappa statistics are summarized in the last row of Table 10. Cross validation was not performed on ASSC-SVM models.

Because subsets of the testing cases were used in runs of this algorithm, the reference distribution of `functional` and `requires attention` wells will likely differ from the overall distribution for the entire set $\mathcal{P}$ (i.e. the no information rates will differ for different subsets of $\mathcal{P}$). While the Accuracy values are certainly informative, they are not as reliable as Kappa statistics to compare ASSC-SVM to other classification methods. Because the Kappa statistic takes into account the accuracy that would be generated simply by chance (Kuhn and Johnson 2013) it will be best served to compare ASSC-SVM models to each other as well as to all other models presented here.

Table 10

Actual and 10−Fold Cross Validation Accuracies and Kappa Statistics for Comparison of All Models Presented

		Actual		10-Fold CV	
	Model	Accuracy	Kappa	CV-Accuracy	CV-Kappa
Logistic Regression	13 Predictors	0.6855	0.3703	0.7861	0.5148
	15 Predictors	0.6883	0.3748	0.7911	0.5251
SVM	13 Predictors	0.6754	0.3533	0.8147	0.5785
	15 Predictors	0.6759	0.3535	0.8192	0.5909
Random Forest	13 Predictors	0.6711	0.3455	0.8413	0.6352
	15 Predictors	0.6790	0.3568	0.8450	0.6464
ASSC-SVM	13 Predictors	?0.70(1.5more)	?0.40(1.5more)	-	-
	15 Predictors	0.7255	0.4486	-	-

Discussion

It is quite interesting to note that of the fundamental classifiers in Table 10 (all but ASSC-SVM), the logistic regression models produce the lowest cross validation Accuracy and Kappa statistics but the highest actual Accuracy and Kappa statistics. This method of classifying wells in the training set is the least flexible of the three methods considered here, but that seems to serve it well. The more flexible classification methods of random forest and SVM certainly have higher cross validation Kappa statistics, but fall short when they are used to predict on testing cases outside of the corner. The random forest model that includes `population` and `construction_year` (15 predictor variables) outperforms the random forest model that excludes those variables, but only by a relatively small amount. For the SVM models though, it is not clear that including the `population` and `construction_year` variables improves the model at all.

The ASSC-SVM classifiers clearly outperform all of the fundamental models considered here. By fitting SVM classifiers to $B = 201$ weighted bootstrap samples of (only) size $q = 150$ and letting each classifier vote to make a final prediction for each test case, we are able to outperform logistic regression, as well as SVM and random forest which were both optimally tuned. Particularly interesting is that the ASSC-SVM outperformed the optimally tuned SVM which implies that ASSC does alter the decision boundary based on the predicting (testing) set (Zhou 2014).

The ASSC-SVM model with 15 predictor variables has a much larger (average) Kappa statistic than ACCS-SVM with 13 predictor variables. This makes sense because we take weighted bootstrap samples of training cases based on which training cases are "most near" the test case we consider. Using the `population` and `construction_year` variables in the dissimilarity function in Step 1 of the algorithm outlined by Zhou (2014) will give much better selection probabilities in Step 3. Using information about how far the test cases are with regard to the variables that specified the split for the corner will be very informative of which training cases in the corner are nearest the test case.

When it comes to choosing a single classifier to implement in a business application, such as the motivating problem from Company X, the results from ASSC algorithm are very attractive. It has many benefits and can be used in combination with simple fundamental methods. Unfortunately, since this algorithm is so computationally expensive it would not serve a company that needs to classify hundreds of thousands of transactions in a single month very well. If time and resources aren't an issue ASSC or something similar seems to be the best course of action. Otherwise logistic regression does an okay job as far as extrapolating with parametric methods goes.

Future Work

Optimizing the ASSC algorithm so that larger subsets of the testing set $\mathcal{P}$ (or the entire set) can be classified in a reasonable amount of time would be quite useful and yield more precise results. This may require using a lower level programming language or parallel computing in `R`. Then, using the ASSC algorithm with other simple funcdamental classifiers would be a great next step. It would also be quite interesting to implement different dissimilarity functions in Step 1 of the ASSC algorithm.

Additionally, I would like to explore the idea of cross validation within ASSC. First, the ability to tune parameters of the fundamental classifier used within the algorithm might significantly improve results. Next, finding some way to "tune" the dissimilarity function might prove quite useful. It would allow us to get a better grip on which training cases are most like the testing case at hand, then assign those training cases higher probabilities of being chosen during bootstrap sampling by way of a dissimilarity (similarity) index. This would allow a model to be built on a set of training cases that are very similar to the testing case. This idea is at the heart of the ASSC algorithm and is extrememly valuable, so setting a reasonable dissimilarity function is very important.

Appendix

Logistic Regression Equations

13 Predictor Variables

$$
\begin{aligned}
logit(status\_group) = {} & \beta_0 + \beta_{1,1} x_1 + \beta_{2,1} x_2 \\
& + \beta_{3,1} I_{\{x_3 = LakeNyasa\}} + \beta_{3,2} I_{\{x_3 = LakeRukwa\}} \\
& + \beta_{3,3} I_{\{x_3 = LakeTanganyika\}} + \beta_{3,4} I_{\{x_3 = LakeVictoria\}} + \beta_{3,5} I_{\{x_3 = Pangani\}} \\
& + \beta_{3,6} I_{\{x_3 = Rufiji\}} + \beta_{3,7} I_{\{x_3 = Ruvuma/S.Coast\}} + \beta_{3,8} I_{\{x_3 = Wami/Ruvu\}} \\
& + \beta_{4,1} I_{\{x_4 = True\}} + \beta_{4,2} I_{\{x_4 = Unknown\}} \\
& + \beta_{5,1} I_{\{x_5 = Other\}} + \beta_{5,2} I_{\{x_5 = Parastatal\}} + \beta_{5,3} I_{\{x_5 = PrivateOp.\}} + \beta_{5,4} I_{\{x_5 = SWC\}} \\
& + \beta_{5,5} I_{\{x_5 = Trust\}} + \beta_{5,6} I_{\{x_5 = VWC\}} + \beta_{5,7} I_{\{x_5 = WaterAuth.\}} \\
& + \beta_{5,8} I_{\{x_5 = WaterBoard\}} + \beta_{5,9} I_{\{x_5 = WUA\}} + \beta_{5,10} I_{\{x_5 = WUG\}} \\
& + \beta_{6,1} I_{\{x_6 = True\}} + \beta_{6,2} I_{\{x_6 = Unknown\}} \\
& + \beta_{7,1} I_{\{x_7 = handpump\}} + \beta_{7,2} I_{\{x_7 = motorpump\}} + \beta_{7,3} I_{\{x_7 = other\}} \\
& + \beta_{7,4} I_{\{x_7 = submersible\}} + \beta_{7,5} I_{\{x_7 = wind-powered\}} \\
& + \beta_{8,1} I_{\{x_8 = other\}} + \beta_{8,2} I_{\{x_8 = parastatal\}} + \beta_{8,3} I_{\{x_8 = unkown\}} + \beta_{8,4} I_{\{x_8 = user-group\}} \\
& + \beta_{9,1} I_{\{x_9 = other\}} + \beta_{9,2} I_{\{x_9 = payannually\}} + \beta_{9,3} I_{\{x_9 = paymonthly\}} \\
& + \beta_{9,4} I_{\{x_9 = payperbucket\}} + \beta_{9,5} I_{\{x_9 = paywhenschemefails\}} + \beta_{9,6} I_{\{x_9 = unknown\}} \\
& + \beta_{10,1} I_{\{x_{10} = flouride\}} + \beta_{10,2} I_{\{x_{10} = good\}} + \beta_{10,3} I_{\{x_{10} = milky\}} + \beta_{10,4} I_{\{x_{10} = salty\}} + \beta_{10,5} I_{\{x_{10} = unkown\}} \\
& + \beta_{11,1} I_{\{x_{11} = enough\}} + \beta_{11,2} I_{\{x_{11} = insufficient\}} + \beta_{11,3} I_{\{x_{11} = seasonal\}} + \beta_{11,4} I_{\{x_{11} = unkown\}} \\
& + \beta 12, 1 I_{\{x_{12} = surface\}} + \beta 12, 1 I_{\{x_{12} = unknown\}} \\
& + \beta_{13,1} I_{\{x_{13} = communalstandpipe\}} + \beta_{13,2} I_{\{x_{13} = communalstandpipemultiple\}} + \beta_{13,3} I_{\{x_{13} = handpipe\}} \\
& + \beta_{13,4} I_{\{x_{13} = handpipe\}} + \beta_{13,5} I_{\{x_{13} = improvedspring\}} + \beta_{13,1} I_{\{x_{13} = other\}}
\end{aligned}
$$

where,
x_1 = amount_tsh
x_2 = gps_height
x_3 = basin
x_4 = public_meeting
x_5 = scheme_management
x_6 = permit
x_7 = extraction_type_class
x_8 = management_group
x_9 = payment
x_{10} = quality_group
x_{11} = quantity
x_{12} = source_class
x_{13} = waterpoint_type

15 Predictor Variables

$$\begin{aligned}
logit(status\_group) = {} & \beta_0 + \beta_{1,1}x_1 + \beta_{2,1}x_2 \\
& + \beta_{3,1}I_{\{x_3=LakeNyasa\}} + \beta_{3,2}I_{\{x_3=LakeRukwa\}} \\
& + \beta_{3,3}I_{\{x_3=LakeTanganyika\}} + \beta_{3,4}I_{\{x_3=LakeVictoria\}} + \beta_{3,5}I_{\{x_3=Pangani\}} \\
& + \beta_{3,6}I_{\{x_3=Rufiji\}} + \beta_{3,7}I_{\{x_3=Ruvuma/S.Coast\}} + \beta_{3,8}I_{\{x_3=Wami/Ruvu\}} \\
& + \beta_{4,1}I_{\{x_4=True\}} + \beta_{4,2}I_{\{x_4=Unknown\}} \\
& + \beta_{5,1}I_{\{x_5=Other\}} + \beta_{5,2}I_{\{x_5=Parastatal\}} + \beta_{5,3}I_{\{x_5=PrivateOp.\}} + \beta_{5,4}I_{\{x_5=SWC\}} \\
& + \beta_{5,5}I_{\{x_5=Trust\}} + \beta_{5,6}I_{\{x_5=VWC\}} + \beta_{5,7}I_{\{x_5=WaterAuth.\}} \\
& + \beta_{5,8}I_{\{x_5=WaterBoard\}} + \beta_{5,9}I_{\{x_5=WUA\}} + \beta_{5,10}I_{\{x_5=WUG\}} \\
& + \beta_{6,1}I_{\{x_6=True\}} + \beta_{6,2}I_{\{x_6=Unknown\}} \\
& + \beta_{7,1}I_{\{x_7=handpump\}} + \beta_{7,2}I_{\{x_7=motorpump\}} + \beta_{7,3}I_{\{x_7=other\}} \\
& + \beta_{7,4}I_{\{x_7=submersible\}} + \beta_{7,5}I_{\{x_7=wind-powered\}} \\
& + \beta_{8,1}I_{\{x_8=other\}} + \beta_{8,2}I_{\{x_8=parastatal\}} + \beta_{8,3}I_{\{x_8=unkown\}} + \beta_{8,4}I_{\{x_8=user-group\}} \\
& + \beta_{9,1}I_{\{x_9=other\}} + \beta_{9,2}I_{\{x_9=payannually\}} + \beta_{9,3}I_{\{x_9=paymonthly\}} \\
& + \beta_{9,4}I_{\{x_9=payperbucket\}} + \beta_{9,5}I_{\{x_9=paywhenschemefails\}} + \beta_{9,6}I_{\{x_9=unknown\}} \\
& + \beta_{10,1}I_{\{x_{10}=flouride\}} + \beta_{10,2}I_{\{x_{10}=good\}} + \beta_{10,3}I_{\{x_{10}=milky\}} + \beta_{10,4}I_{\{x_{10}=salty\}} + \beta_{10,5}I_{\{x_{10}=unkown\}} \\
& + \beta_{11,1}I_{\{x_{11}=enough\}} + \beta_{11,2}I_{\{x_{11}=insufficient\}} + \beta_{11,3}I_{\{x_{11}=seasonal\}} + \beta_{11,4}I_{\{x_{11}=unkown\}} \\
& + \beta 12, 1I_{\{x_{12}=surface\}} + \beta 12, 1I_{\{x_{12}=unknown\}} \\
& + \beta_{13,1}I_{\{x_{13}=communalstandpipe\}} + \beta_{13,2}I_{\{x_{13}=communalstandpipemultiple\}} + \beta_{13,3}I_{\{x_{13}=handpipe\}} \\
& + \beta_{13,4}I_{\{x_{13}=handpipe\}} + \beta_{13,5}I_{\{x_{13}=improvedspring\}} + \beta_{13,1}I_{\{x_{13}=other\}} + \beta_{14,1}x_{14} + \beta_{15,1}x_{15}
\end{aligned}$$

where,
x_1 = amount_tsh
x_2 = gps_height
x_3 = basin
x_4 = public_meeting
x_5 = scheme_management
x_6 = permit
x_7 = extraction_type_class
x_8 = management_group
x_9 = payment
x_{10} = quality_group
x_{11} = quantity
x_{12} = source_class
x_{13} = waterpoint_type
x_{14} = population
x_{15} = construction_year

References

DrivenData Inc. 2017. *Pump It up: Data Mining the Water Table (Webpage).* URL: https://www.drivendata.org/competitions.

Hastie, Trevor, Robert Tibshirani, and Jerome H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer.

Hijmans, Robert et al. 2017. *GADM Database of Global Administrative Areas (Webpage).* URL: http://www.gadm.org/country.

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

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

Quiñonero-Candela, Joaquin, Masashi Sugiyama, Anotn Schwaighofer, and Niel D. Lawrence, eds. 2009. *Dataset Shift in Machine Learning.* The MIT Press.

Zhou, Wen. 2014. *Some Bayesian and Multivariate Analysis Methods in Statistical Machine Learning and Applications.* Graduate Theses and Dissertations. Paper 13816.