A Feature-Selection Based Approach for the Detection of Diabetes in Electronic Health Record Data

Likhitha Devireddy (likhithareddy@gmail.com)
David Dunn (jamesdaviddunn@gmail.com)
Michael Sherman (michaelwsherman@gmail.com)

7 May 2014

1 Abstract

Type II diabetes is a serious health problem, with over 25 million Americans diagnosed. Thus, diabetes prediction is a major focus of interest, within both the medical community and the general public. In 2012, Kaggle.com, a competitive data-analysis website, released a set of electronic health record data and challenged its users to predict which patients had diabetes. Although the competition is over, we used this rich dataset to build models predicting diabetes from a patient's electronic medical record.

Traditional approaches to a modeling problem like this are often based on domain knowledge, which is used to guide the selection and creation of relevant features for predictive modeling. Instead of this domain-based approach, we considered a feature selection-based approach. Our approach began with the generation of a very large feature table, which we then subsetted using various feature reduction techniques (of both "wrapper" and "filter" types). We then induced a variety of models, and assembled these models into ensembles in attempts to maximize lift. Our final models were more than three times as effective as random selection, suggesting this feature generation and selection approach is competitive with a domain knowledge-based approach. Our feature selection-based approach also allowed for the discovery of unexpectedly relevant features, something a domain-based procedure does not allow for. Although feature selection can be more labor-intensive than a domain based approach, for this diabetes problem feature selection led to both powerful predictive models and interesting insights.

2 Introduction & Background

Type II diabetes is a serious health problem in the United States and across the world. Over 25 million Americans currently suffer from diabetes, and monitoring and treating diabetes requires

regular blood tests, sticking to a restricted diet, and regular insulin shots [Kaggle.com, 2014]. Diabetes is also associated with a range of complications, including heart disease, stroke, kidney disease, blindness, and loss of limbs [Kaggle.com, 2014]. Early treatment of diabetes can help minimize complications, and learning more about the co-morbidities of diabetes can help researchers better understand the risks and warning signs associated with diabetes.

For these reasons, Kaggle.com partnered with Practice Fusion (a vendor of electronic health record software) [Practice Fusion, 2014] to create a contest where data analysts worked to predict diabetes in patients based on other medical information about those patients.

Predicting diabetes from medical information is a difficult task, since medical information for a single patient is often stored in multiple locations. Furthermore, the storage schema of electronic medical records makes getting "one set of predictors" per patient tricky, as patients can have multiple records of medications, diagnoses, allergies, etc. for a single visit, and most patients visit a healthcare provider multiple times. The Kaggle competition forced participants to meet these challenges head-on.

Although the contest ended on September 10, 2012, we decided to attempt the challenge to learn more about working with healthcare data. We did not look into the winners' solutions until after we had created and tested most of our own models. When we finally read the winner and runners-up documentation, it was apparent that they used domain knowledge to inform their feature choice and model construction. We possessed significantly less domain knowledge than of many contest participants who work in the healthcare industry, but our research suggests that statistical feature selection methods often perform as-well or better than domain-knowledge-based approaches when selecting features to construct models [Cheng et al., 2006]. Because of this research, we tried several different feature selection methods, and created models based on 25 different datasets, which we winnowed down after further analysis of model and dataset predictive power. Eventually we settled on two models maximizing lift at the population and dataset diabetes incidence rates, which were ensembles of 25 and 93 models, respectively.

Lastly, we wish to note that the challenge was not about predicting future diabetes, but about predicting diabetes in patients who already had a diagnosis–diagnoses of diabetes (as well as diabetes-related medications and lab results) were masked from the data. While this task is certainly more manageable than predicting diabetes in patients who have not yet received a medical diagnosis, it is also ultimately less useful in the real world (if still a valid learning activity).

3 Feature Creation and Table Merging

3.1 Introduction

The data was provided by Kaggle, and consisted of 17 tables filled with various aspects of patient healthcare information (see Figure 1 for a full breakdown with tables of fields). A total of 9948 patients were in the dataset, with 1904 patients having diabetes, a rate of 19.1%. Several tables, such as `SyncPatientSmokingStatus` and `SyncTranscriptDiagnosis`, only functioned as linkage tables between other tables. Many tables, however, contained multiple entries for a single patient, and these multiple entries had to be reduced to a single entry per patient before modeling could occur. Furthermore, some tables only had information for a subset of patients (including `SyncImmunization`, which contained information on only 6 patients and was not considered).

We spent significant time determining how to best reduce the data to "one row per patient" while still preserving as much of the information as possible. We also had to deal with general issues of data cleaning, including contradictory information, junk fields, empty fields, and redundant features. Precise information about the exact fields in each table can be seen in Figure 1. For the sake of brevity, we will not list all the fields as we discuss each table.

3.2 The `SyncPatient` Table

The `SyncPatient` table contained a single row for each of the 9948 patients. Each row contained information about the patient's gender, year of birth, and state. Gender and year of birth were kept as fields, and state was replaced with the percent prevalence of diabetes in that state, as provided by the Centers for Disease Control [Centers for Disease Control and Prevention, 2012]. Age was calculated and added as a feature.

3.3 The `SyncAllergy` Table

The `SyncAllergy` table contained 2798 records detailing the allergy information of 1725 different patients. The table contained information about the allergen (including a medication name, if relevant), the year the allergy was diagnosed, and the symptoms and severity of the allergic reaction.

We split the allergen field into 18 binary fields representing groupings of allergens. We similarly split the allergic reaction descriptive field into 25 binary fields representing groupings of reactions. We then transformed the qualitative severities ("very mild" to "severe") into a numerical field ranging from 1 to 4. We multiplied the binary reaction and allergen fields by severity, to give a number from 1 to 4 instead of a binary value for allergen and allergic reaction. This transformation allowed information about allergic reaction and severity to be contained within the same feature, and information about allergen and severity to be contained within the same feature. Finally, we calculated the length of time every allergy had been diagnosed for.

Additional features were then calculated on a per-patient basis: the average time since diagnosis of all the allergies a patient had, a "max" field with the highest severity of all a patient's allergies, a count of the number of "severe" allergies a patient had, and mean of a patient's allergy severities. Finally, we "rolled up" some of the 18 groupings of allergens into some alternative smaller bins by medication families, and the 25 groupings of reactions into smaller bins based on the biological system (skin, circulatory, respiratory, etc.) the reaction occurred in.

3.4 The `SyncTranscript` Table

The `SyncTranscript` table contained 131,032 records of patient-provider interactions, and all 9948 patients had at least one record. The table contained vitals at the time of visit (height, weight, BMI, blood pressure, temperature, respirations), visit year, and the specialty of the provider. Missing values were found throughout the table, and the combination of missing values with multiple records per patient made the SyncTranscript table especially cumbersome to deal with.

We created features of the minimum, maximum, and range of the following fields: height, weight, BMI, systolic BP, diastolic BP, respirations, and temperature. We also created a feature of the time since last the last visit to a provider. Lastly, we created a feature with a count of the total number of patient visits, and a set of features with visit counts by the specialty of the provider.

3.5 The `SyncDiagnosis` Table

The `SyncDiagnosis` table contained only a few fields: diagnosis (ICD-9) codes, start and stop years of the diagnoses, if the diagnosis was acute and/or current, and information about the doctor and patient linking the diagnoses to other tables. But with 94,831 records across all 9948 patients, `SyncDiagnosis` held a wealth of information that translated to hundreds of features in our complete feature set.

The diagnosis descriptions matched numeric ICD-9 codes, which is where the bulk of the information lay [Centers for Disease Control and Prevention, 2009]. ICD-9 codes are organized in a hierarchical manner, with nearby numbers related to each other, and decimal places used to provide extremely specific information extending the meaning of the primary 3-digit code. Thus, truncating the ICD-9 codes to the whole number or tens digit provides an easy way to group diagnoses without losing much information. `SyncDiagnosis` contained 3134 unique ICD-9 codes, but truncating to the whole number left only 617 unique codes. We kept many of these 617 truncated ICD-9 code bins as features with a count of times a patient was assigned a code, but many of these bins were eliminated due to low variance. Finally, we added more features by rolling up the ICD9 codes into 62 separate bins based on families of conditions (corresponding to headers in the ICD-9 manual), with more granular bins related to endocrine-system issues.

Capturing the information contained in the start year, stop year, and acute fields was espe-

cially difficult. For each of the 62 bins of diagnosis families, we applied the following formula in order to incorporate the start year, stop year, and acute condition fields:

$$1 + \ln\left(endYear - startYear + 1\right) \times \frac{acute}{current}$$

where:
$endYear$ is final year of the diagnosis
$startYear$ is first year of the diagnosis
$acute = 2$ if the diagnosis is acute, otherwise $acute = 1$
$current = 1$ if the diagnosis is current, otherwise $current = 2$

Most of the diagnosis were current, not acute, and were fairly new. A few, however, had records of significant duration, and the log-transform of the diagnosis duration gives credit to the duration of the diagnosis. Further, the acute multiplier and current divider both add information about the severity of the diagnosis. The result of this formula can be stored in one cell per diagnosis, while representing several cells of data.

3.6 The `SyncConditions` and `SyncPatientConditions` Tables

`SyncPatientConditions` contained flags of patients with no known allergies and no known medications (a total of 2836 statuses from 2824 patients), with the status descriptions in the `SyncConditions` lookup table. Two Binary features marking patients with these two flags.

3.7 The `SyncSmokingStatus` and `SyncPatientSmokingStatus` Tables

`SyncPatientSmokingStatus` contained 4940 entries corresponding to 4427 patients. The different smoking statuses recorded by Practice Fusion roughly overlapped with the National Institute of Standards and Technology (NIST) smoking status codes [Office of the National Coordinator for Health Information Technology, 2010]. NIST smoking codes are primarily concerned with how often a person smokes and if they have smoked in the past. However, some of the statuses available to medical providers in Practice Fusion offered information about quantity of cigarettes per day, and we made an attempt to capture this as well. Issues arose when we discovered that some patients had multiple smoking statuses (corresponding to multiple visits) with contradictory information. We attribute these errors (which occurred in about 2% of patients) to Practice Fusion's software, which has two contradictory fields beginning with the same text: "0 cigarettes per day (previous smoker)", and "0 cigarettes per day (non-smoker or less than 100 in lifetime)". These type of contradictory errors were cleaned on a record-by-record basis.

Next, we considered a handful of different transformations of multiple smoking statuses, with ranging granularity of information. One transformation captured non-smokers, ex-smokers, and

current smokers separately, with binary features capturing quantity of current smoking (ranging from, "Few cigarettes per day", to, "2 or more packs per day"), as well as a binary field marking if the patient reduced their level of smoking over time period in the dataset. Another transformation merged this specific information into three groups: current smokers, ex-smokers, and never smoked. A final transformation simply grouped patients into "smoking" and "non-smoking".

3.8 The `SyncPrescription` Table

`SyncPrescription` contained 78,863 prescription records for 8953 patients, with fields capturing the medication, the year the prescription was written, the quantity, refill information, and if generic substitution was allowed. We calculated the following 10 features from this data on a per-patient basis: total count of prescriptions, mean prescriptions per year, total count of prescribed refills, mean refills per prescription, total count of "refill as needed" prescriptions, ratio of "refill as needed" prescriptions to total prescriptions, total count of generic prescriptions, and ratio of generic prescriptions to total prescriptions.

3.9 The `SyncMedication` Table

`SyncMedication` contained 44,520 medication history records for 9846 patients. The 2553 unique medications in `SyncMedication` were re-classified into 109 groups to reduce sparsity while retaining information, using medical RxList [2014] and WebMD [2014]. These 109 groups each became feature, with a per-patient count of prescribed medications in each group.

3.10 The `SyncLabResult` `SyncLabObservation` `SyncLabPanel` Table

`SyncLabResult`, `SyncLabObservation`, and `SyncLabPanel` contained information about laboratory work in a nested structure. `SyncLabObservation`, contained individual lab tests ("Vitamin B", "LDL Cholesterol", "Hemoglobin", etc.) with a test result, which were part of a panel listed in `SyncLabPanel`. The lab panels were linked to `SyncLabResult`, which were then linked to patients. Because of this nested structure separating individual patients from the actual test results, we considered all of these tables as a single set of information. In all, there were 29,014 individual lab tests performed on 791 patients.

After combining the tables, we focused our attention on a field marking individual lab test results as abnormal and used this to construct meaningful features. We grouped individual lab results into 101 panels, and then grouped these 101 panels into 31 sets of panels based on the diseases lab panels were attempting to identify (using information from WebMD [2014]). Each of the 31 panel sets became a feature, with a per-patient count of of abnormal lab results from each set. Finally, a count of total lab results (normal or abnormal) and a mean number of lab results per year were added on a per-patient basis.

3.11 Joining the Individual Tables into a Single Features Table

All of these transformed tables were joined together into a single features table, for a total of 1043 features on each of 9948 patients. During the joining, not every patient was captured in every table of transformations. For example, only 791 patients had lab results, leaving 9157 patients without any lab data. One approach we considered would have been to leave these fields with a missing value marker, indicating data was not available. However, because of R's difficulty in dealing with missing values, we decided we had to replace these missing values. For some of the transformations, this made sense–for example, in a feature capturing allergy severity where 1 was the lowest result, setting a value of 0 for patients with no allergy made sense. For other transformations, however, less obvious solutions were necessary. To address this issue, we added more features indicating a patient's presence or absence in other fields in the table. For example, from lab data, a new binary feature was created where a 1 indicated they had received lab results, and a 0 indicated they had not. After eliminating features with a variance less than .01, our table had 980 features.

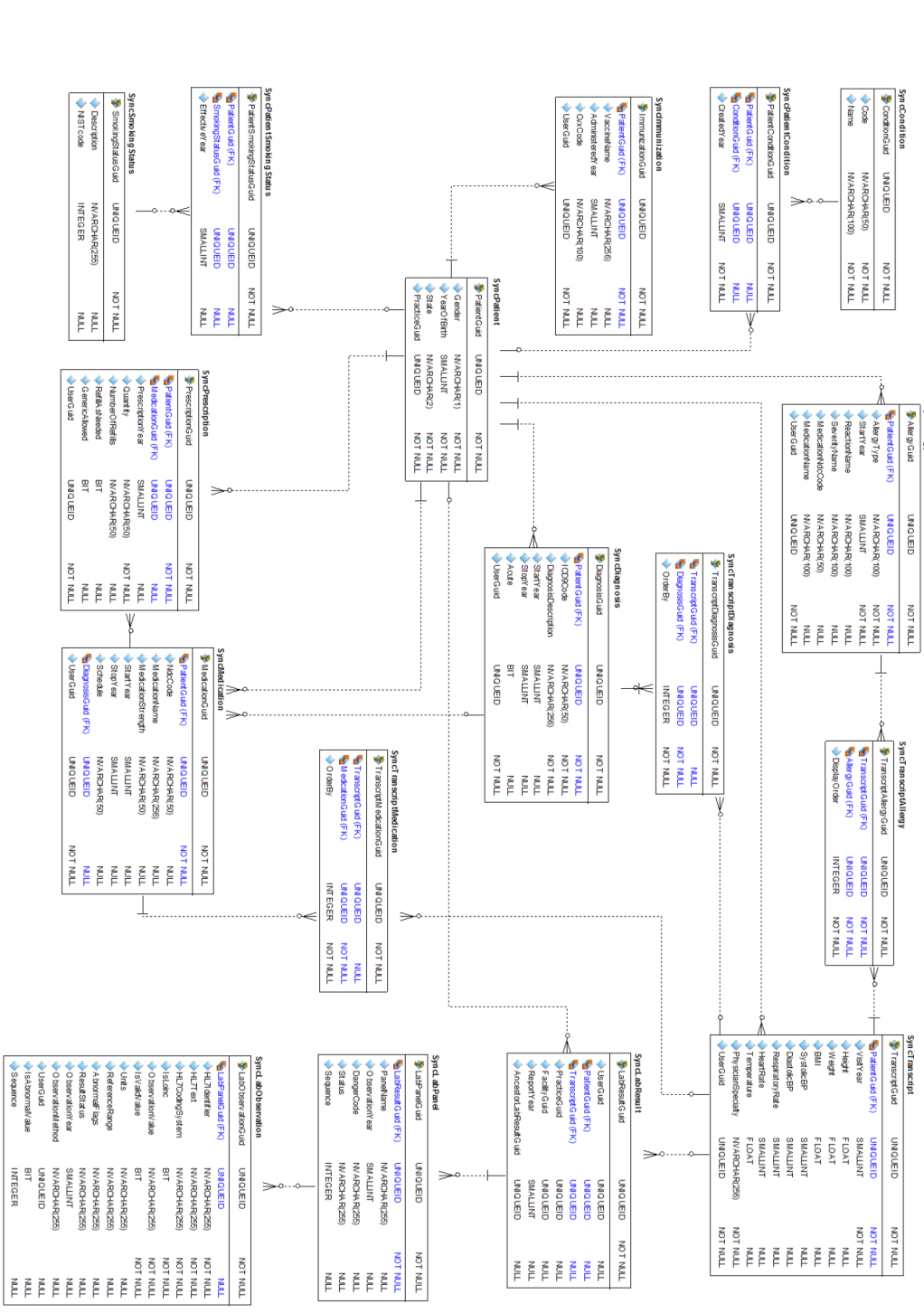


Figure 1: Database schema of the Practice Fusion diabetes dataset. This figure should zoom-in well on an original copy of the pdf.

4 Numerical Pre-Processing

We created three numerical transformations of our 980 features. A binary transformation made every data point either 0 or 1 (1 meaning the original data had any value besides 0). A standardization transformation mean centered all features at 0, with 1 being one standard deviation above the mean and -1 being one standard deviation below the mean. And a logarithmic transformation made every data value x equal to $ln(x + 0.2)$.

The complete summary of the numerical transformations as combined with different feature selection techniques is in Table 1. Note that numerical transformations were performed after feature selection was complete on the untransformed data, primarily for reasons of time. For some of our feature selection techniques, different numerical transformations would have given different feature subsets.

No.	Feature Selection	Transformation	#Features
1	CFS	Binary	13
2	CFS	Log	13
3	CFS	None	13
4	CFS	Standardize	13
5	High Correlation($\rho = 0.8$)	Binary	679
6	High Correlation($\rho = 0.8$)	Log	679
7	High Correlation($\rho = 0.8$)	None	679
8	High Correlation($\rho = 0.8$)	Standardize	679
9	Lasso	Binary	134
10	Lasso	Log	134
11	Lasso	None	134
12	Lasso	Standardize	134
13	Low Correlation($\rho = 0.25$)	Binary	421
14	Low Correlation($\rho = 0.25$)	Log	421
15	Low Correlation($\rho = 0.25$)	None	421
16	Low Correlation($\rho = 0.25$)	Standardize	421
17	None	Binary	980
18	None	Log	980
19	None	None	980
20	None	Standardize	980
21	Principal Components	Principal Components	275
22	Random Forest	Binary	50
23	Random Forest	Log	50
24	Random Forest	None	50
25	Random Forest	Standardize	50

Table 1: 25 Datasets From Numeric Transformation and Feature Selection Combinations

5 Feature Selection

5.1 Introduction

After we aggregated the data from all the individual tables into a single feature set, we had 9,948 patients with 980 (partially redundant) features. While some modeling techniques can handle a large number of highly correlated features, others cannot. Additionally, we lacked the computing power to run more sophisticated, computationally intensive modeling techniques (like SVMs and GBDTs) on such a large dataset. Because of these issues, we turned to feature selection in an attempt to reduce the feature size (p) of our dataset.

We used six different feature selection processes. Four of the processes are "filter" approaches done prior to modeling, and two of the processes are "wrapper" approaches done as part of modeling. Technically, our "wrapper" approaches (using Random Forests and Lasso to select features) become "filter" approaches when the feature sets selected by a model's "wrapper" are then used in other models, but this is merely a semantic issue.

The results of the feature selection (as combined with the numerical transformations) are seen in Table 1.

5.2 Principal Components Analysis

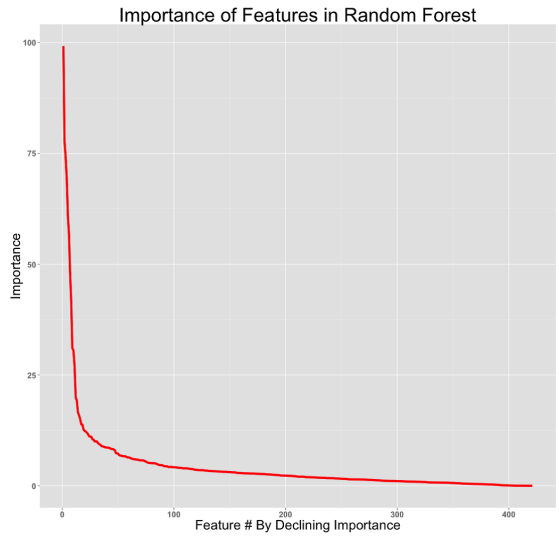
While principal components is technically not a feature selection technique, since it was an attempt to reduce the number of features we describe it here. Principal components analysis [Kuhn and Johnson, 2013] was applied to the 980 features. All features with an eigenvalue (λ) greater than 1 were kept, for a total of 275 features post-PCA (see Figure 2c)

5.3 Pair-Wise Correlation Reduction

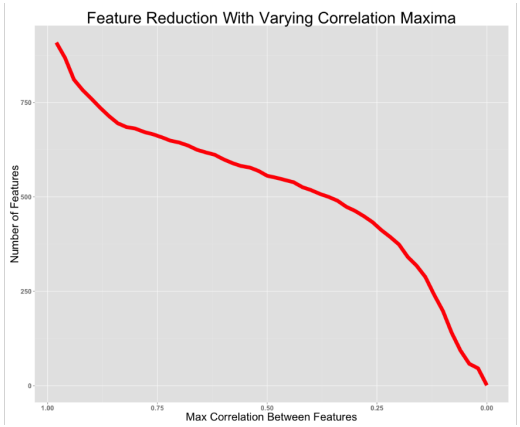
Using the `findCorrelation` function in the `caret` R package [Kuhn et al., 2014], we eliminated features having the highest pair-wise correlations. `FindCorrelation` works by searching for the highest absolute pair-wise correlation, determining the mean absolute correlation of each pair member with the other features, and then eliminating the feature with the greater mean absolute correlation from the dataset. This process is repeated iteratively until there are no remaining pair-wise correlations above a user-chosen threshold.

To determine the cutoff threshold, we applied the `findCorrelation` process across the 0 to 1 range of possible absolute correlations. This gave us a plot of absolute correlation thresholds vs. number of features (Figure 2b).

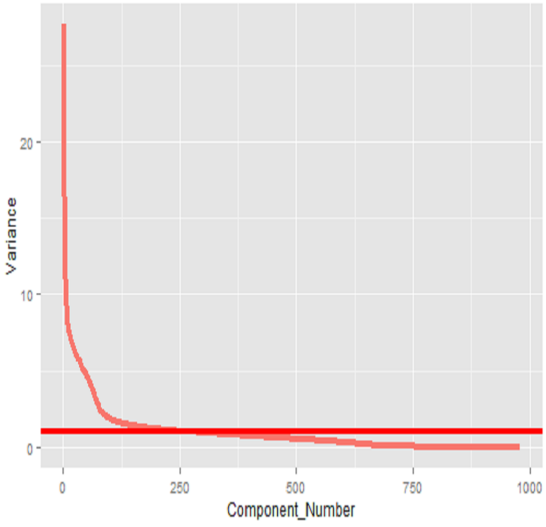
In the graph, we noticed that at pair-wise correlation thresholds of approximately $\rho = 0.8$ and approximately $\rho = 0.25$ were the greatest changes in slope. We used both these values as thresholds for the `findCorrelation` function, creating two feature subsets. The high correlation



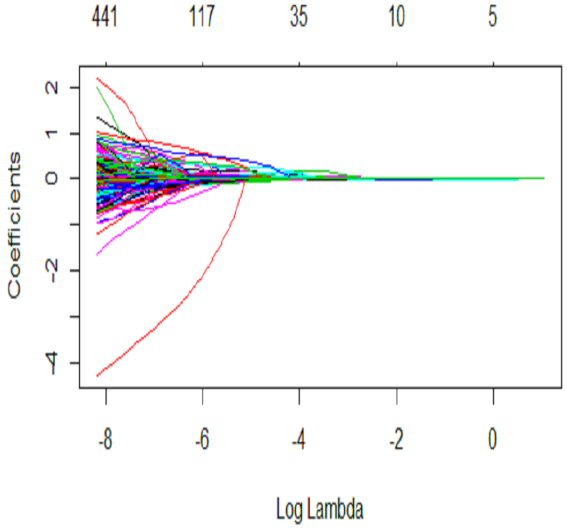
(a) Reduction in Gini Coefficient vs. Features, Ranked by Importance



(b) Number of Features vs. `findCorrelation` Threshold



(c) Lowest Eigenvalue vs. Principal Components Retained. The red line is $\lambda = 1$.



(d) Values of Coefficients vs. Changing Lambda in a Lasso model

Figure 2: Figures Related to Feature Selection

($\rho = 0.8$) feature subset had 679 features, and the low correlation ($\rho = 0.25$) feature subset had 421 features.

5.4 Correlation Feature Selection

We applied Correlation Feature Selection (CFS) as described in Hall and Smith [1997]. Descriptively, CFS works by selecting the features that are most correlated with the target feature but

not correlated with each other. CFS produced the smallest feature set of the entire project, with only 13 features retained (see Table 2). Many of these features can be seen as proxy for age, which indicates that older people are more likely to get diabetes, or a proxy for poor digestive health, which is also strongly associated with diabetes.

5.5 Random Forests Importance

Using the Random Forests modeling technique in R's `randomForests` [Liaw and Wiener, 2002] package, we induced a predictive model. As part of inducing a model, the `randomForests` package provides relative importance measures of features based on how much that feature reduces the impurity (measured by gini coefficient) of child nodes.

The first attempt at feature selection using Random Forests on the full set of 980 features had highly correlated variables grouped together when features were sorted by their mean reduction in impurity. For example, of the top 5 features, two were maximum BMI and minimum BMI, and another two were age and date of birth. Inducing a Random Forests model on the feature set with all absolute pair-wise correlations above 0.80 removed (p=679) still had correlated features grouped together. Finally, inducing a Random Forests model on the feature set with all absolute pair-wise correlations above 0.25 removed (p=421) and number of trees=1000 (so Random Forest's random selection of features at each spilt gave equal consideration to all features) left few correlated features.

Plotting the gini coefficient of features against their rank in importance showed a Pareto distribution, with a clear inflection point at approximately 50 features. These 50 features became a feature set (see Figure 2a). The top 15 features as selected by Random Forests can be seen in Table 2.

5.6 Lasso Feature Selection

Finally, we induced a Lasso [Tibshirani, 1996] model with R's `glmnet` [Friedman et al., 2010] package. Lasso is a form of regularized regression, which imposes an additional modeling penalty (λ) equal to the summed absolute value of all regression coefficients that is added to the squared errors of the model (see Figure 2d). On the raw feature set (p=980), Lasso reduced the regression coefficients to 0 for 844 features, leaving 136 features. The top 15 features as selected by Lasso can be seen in Table 2.

5.7 Evaluation of Different Feature Selection Approaches

To evaluate the 25 datasets (see Table 1) generated by combinations of feature selection and numeric transformation, we looked at summary statistics of lift values at rates of positive prediction of 8.3% and 19.67% (Table 3). Among the numeric transformations, binary performed the worst with an

average lift value of 2.18 and 2.54 at a rate of positive prediction of 19.67% and 8.3%, respectively. Among feature selection techniques, Low Correlation ($\rho = 0.25$) and Random Forests performed worst, followed by No Transformation. It is important to note that High Correlation ($\rho = 0.8$) (679 predictors), CFS (13 predictors) and Lasso Feature Selection (134 predictors) perform significantly better than No Transformation (980 predictors).

No.	CFS Features	Random Forest Features	Lasso Features
1	Taking cholesterol medication?	Taking cholesterol medication?	Count of diagnoses indicating Lipid Metabolism Disorders
2	Birth year	Minimum Diastolic Blood Pressure from the last year the patient came in	Count of diagnoses of Lipoid Metabolism Disorders
3	Patient's max BMI in the last year they visited the doctor	Highest temperature in the last year the patient came in	Whether the patient has ever been diagnosed with a circulatory system disease
4	Maximum systolic blood pressure	Maximum height recorded from the last year	Count of diagnoses indicating renal disease
5	Score of diagnoses indicating lipid metabolism disorders	Difference between highest and lowest weights recorded in last year	Lab Panel CBC Abnormal
6	Count of diagnoses indicating circulatory system diseases	Has any medication been prescribed?	Taking cholesterol medication?
7	Count of diagnoses indicating lipid metabolism disorders	Chronic Renal Failure	Max single visit score of an external Health Hazard diagnoses
8	Secondary diabetes diagnoses?	Minimum Respiratory Rate recorded in the last year the patient came in	Diagnoses formula score for Lipoid Metabolism Disorders
9	Count of diagnoses indicating blood diseases	How long since the patient last visited the doctor?	Score for genital disorder diagnoses
10	Score of diagnoses indicating multiple circulatory diseases	Range of respiratory rates observed in the last year	Count of chronic Lipoid Metabolism Disorder diagnoses
11	Count of diagnoses indicating hypertension	Diabetes complications	No. of times a patient visited a podiatrist
12	Count of diagnoses indicating renal disease	How many years a patient has had a diagnoses	Count of total diagnoses
13	Has any medication been prescribed?	Maximum Single Visit Score of a Skin Disease diagnosis	Gender
14		Count of unspecified back disorder diagnoses	Count of chronic Digestive Diseases diagnoses
15		Count of unspecified anemias diagnoses	How many allergies a patient has medications for

Table 2: Best Features Remaining after Various Feature Selection Techniques

	Transformation			
Lift 19.67%	**None**	**Standardize**	**Log**	**Binary**
Mean	2.25	2.235	2.207	2.187
SD	0.231	0.234	0.174	0.118

	Feature Selection					
Lift 19.67%	**None**	**High Correlation**	**Low Correlation**	**Lasso**	**Random Forest**	**CFS**
Mean	2.196	2.363	1.959	2.35	2.013	2.305
SD	0.216	0.082	0.078	0.104	0.074	0.123

	Transformation			
Lift 8.3%	**None**	**Standardize**	**Log**	**Binary**
Mean	2.687	2.585	2.65	2.545
SD	0.293	0.244	0.21	0.203

	Feature Selection					
Lift 8.3%	**None**	**High Correlation**	**Low Correlation**	**Lasso**	**Random Forest**	**CFS**
Mean	2.583	2.709	2.457	2.756	2.453	2.619
SD	0.261	0.17	0.242	0.25	0.199	0.245

Table 3: Summary Statistics of Lifts at Rates of Positive Prediction of 8.3% and 19.67%, Organized by Feature Selection Techniques and Numeric Transformations.

6 Modeling

6.1 Metrics and Rationale

Our goal in modeling was to correctly classify patients as having or not having diabetes. Lift was used to evaluate the models, where

$$Lift = \frac{TruePositive}{TruePositive + FalsePositive}$$

In addition to creating lift plots with R's `ROCR` package [Sing et al., 2005], we manually computed lifts. Predictions were ordered from most likely to least likely to have diabetes, and lifts were calculated at specific points in the ordered predictions (sometimes called "point lifts" or "instantaneous lifts"). We measured lift at 8.3% of ordered predictions (the rate of diabetes in the USA [CB Online Staff, 2012]), and 19.7% of ordered predictions (the rate of diabetes in our test set). These instantaneous lifts allowed for the comparison of far more models than could fit on a single lift plot.

When choosing modeling techniques, we looked for techniques which both classified data and had an obvious way to rank predictions (for lift calculations). We chose Logistic Lasso Regression and Logistic Ridge Regression because they are wrapper models suited to dealing with large sets of data, and because they return probabilities rather than merely a prediction. We chose unpenalized Logistic Regression because it is the traditional baseline for classification, and because it returns probabilities. We chose Random Forest because it is a wrapper model suited to dealing with large sets of data. Finally, we chose Support Vectors Machines (SVM) and Gradient Boosted Decision Trees (GBDT) because they tend to perform well in classification tasks. All of our chosen techniques (except logistic regression) were suitable for use with sparse, near-multicollinear datasets.

Calculating lifts from Random Forests, SVMs, and GBDTs required an extra computational step. These modeling techniques do not return probabilities, orderings, or rankings, but merely predictions. To get an ordering of predictions to calculate lifts, we combined multiple predictions from our different datasets(see Table 1). Patient by patient, we took the arithmetic mean of these predictions (where 0=no diabetes and 1=yes diabetes). This allowed an (admittedly coarse) ordering of predictions by likelihood of having diabetes.

6.2 Splitting the Data: Train, Test, and Holdout Sets

To evaluate our model, we subsetted our 9948 samples into three sets. A training dataset of 5200 samples was used to induce models. A test dataset of 1800 samples was used to evaluate the performance of our models and inform decision making. A final "holdout" set of 2948 samples was left untouched to provide a final evaluation of the model on data that was not used to inform model construction.

Raw accuracies of all of our models can be seen in Table 4. Lifts (after ranking predictions by confidence of a patient having diabetes) of the top 25 models at 8.3% of predictions are in Table 5, and lifts of the top 25 models at 19.67% of predictions are in Table 6.

6.3 Logistic Regression

Binomial Logistic Regression models were induced as a baseline for evaluating other modeling techniques. Models for all 25 datasets in Table 1 were induced using the `glmnet` R package [Kuhn et al., 2014]. On our test sets, Logistic Regression never performed better than the nave 80.3%-correct "assume no patient has diabetes" method, and for many test sets the Logistic Regression models had accuracies more than 10% below the other modeling techniques (see Table 4). Logistic Regression is especially ill-suited for sparse (low variance), high-dimensional data. The highest Logistic Regression accuracy was on the PCA dataset, where every feature had a minimal amount of variance. The worst Logistic Regression accuracies were on the data selected by Random Forests- a modeling technique highly indifferent to sparsity or variance.

Because Logistic Regression performed significantly below 80% correct on test data, we decided to abandon it and use the nave 80.3%-correct "assume no patient has diabetes" method as the baseline to evaluate modeling techniques against.

6.4 Ridge Regression

Ridge Regression is a form of regularized regression which assigns a cost penalty to the sum of squared coefficient values. This penalty helps Ridge Regression deal with poor features and multicollinearity in a way Logistic Regression cannot. Our penalty (λ) was tuned using 5-fold cross validation, with tested λs ranging from 113.02-0.01.

Ridge Regression achieved very high test set accuracies of 85.09% and 84.23% on the standardized dataset with all features and the PCA datasets, respectively (Table 4). However, Ridge Regression models did not perform as well as other modeling techniques when evaluating lifts (#19 in Table 5 and #15 in Table 6).

6.5 Lasso

Lasso Regression, which imposes an absolute value penalty similar to Ridge Regression's squared penalty, was used for both feature selection and modeling. To select the best penalty (λ), we used 5-fold cross validation while testing penalties ranging from 2.83-0.0002. We observed that the Lasso model generally had the best lifts with the Lasso-generated datasets, which is unsurprising. We were able to achieve lifts of 3.03 (on top 8.3%: #7 in Table 5) and 2.49 (on top 8.3%: #7 in Table 5) with the test data.

6.6 Random Forests

In addition to regression-based modeling techniques, we used a handful of classification-based modeling techniques, including Random Forests, which we used both to induce models and for feature selection. We determined the number of trees (300) manually, by noting the increase in accuracy on evaluation data as tree count increased. The number of features to consider per split was left at the package default of $\sqrt{p}$, to deal with the differing counts of features across our different datasets.

Random Forests is especially well suited for large feature sets. It was run on all 25 datasets listed in Table 1, and different combinations of the predictions from these 25 datasets (see Table 7) were averaged on a patient-by-sample basis to order the predictions so lifts could be calculated. Random Forests models generally performed very well when lift of the top 8.3% of diabetes likelihood was considered, but not as well at 19.7% of ranked predictions.

6.7 Support Vector Machines (SVMs)

SVMs are powerful classification-based modeling techniques, but they require considerable computing resources to build. One of the reasons to use feature selection on a dataset is so a more computationally-intensive modeling technique like SVM can be used. SVMs were built on the 9 datasets with the fewest features, as detailed in Table 7, and tuned using 5-fold cross-validation repeated twice (aided by the `caret` R package). Slack penalties were tuned using a grid search on a dataset-by-dataset basis. Only a linear kernel was considered.

Like Random Forests, SVMs were created on multiple feature sets and arithmetic means of the 0 and 1 predictions were calculated on a per-patient basis to generate rankings of predictions (Table 7. SVMs did not perform as well as we expected at (#15 in Table 5). Furthermore, the SVMs had a tendency to under-predict diabetes, and less than 19% of the samples had a positive prediction, which meant we could not evaluate the lift of the top-ordered 19.7% of ranked predictions.

6.8 Gradient-Based Decision Trees (GBDTs)

Like SVMs, GBDTs require considerable computing resources. And like SVMs, we induced GBDTs on the 9 datasets with the fewest features, as detailed in Table **??**. The GBDTs were tuned with a grid search using 5-fold cross-validation repeated twice (with the **??** R package). The tuned parameters for the GBDTs were number of trees (100, 150, 200, 500, 750, 1000, 3000, 5000), interaction depth (1, 2, 3, 5, 10, where $2^{(interactiondepth)}$ is the number of terminal nodes), and learning rate (.1, .01, .001). Restrictions on the relationship between number of trees and learning rate were imposed to save computation time (for example, on larger number of trees only smaller learning rates were considered, and on smaller number of trees only greater learning rates were considered).

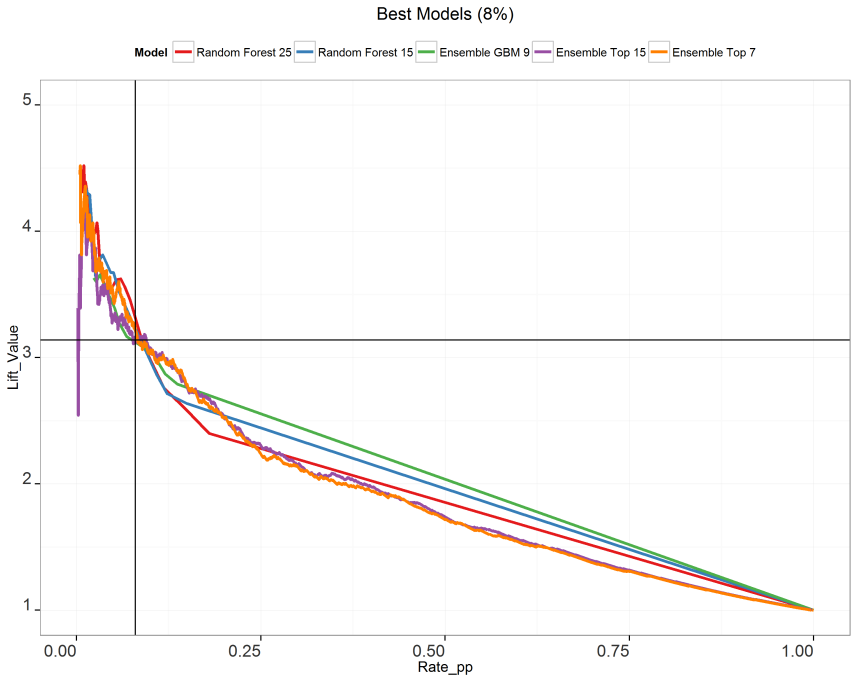
Again, like Random Forests and SVMs, multiple GDBTs from different feature sets had to be combined to generate ranked predictions (Table 7). GBDTs performed very well when the top-ranked 8.3% of positive diabetes classifications were considered (# 5 in Table 5). However, like SVMs, the GBDTs had a tendency to under-predict diabetes, and the lift from the 19.7% highest probability predictions could not be considered since the GBDTs in aggregate delivered positive predictions for fewer than 19.7% of the patients in the dataset.

6.9 Ensembles

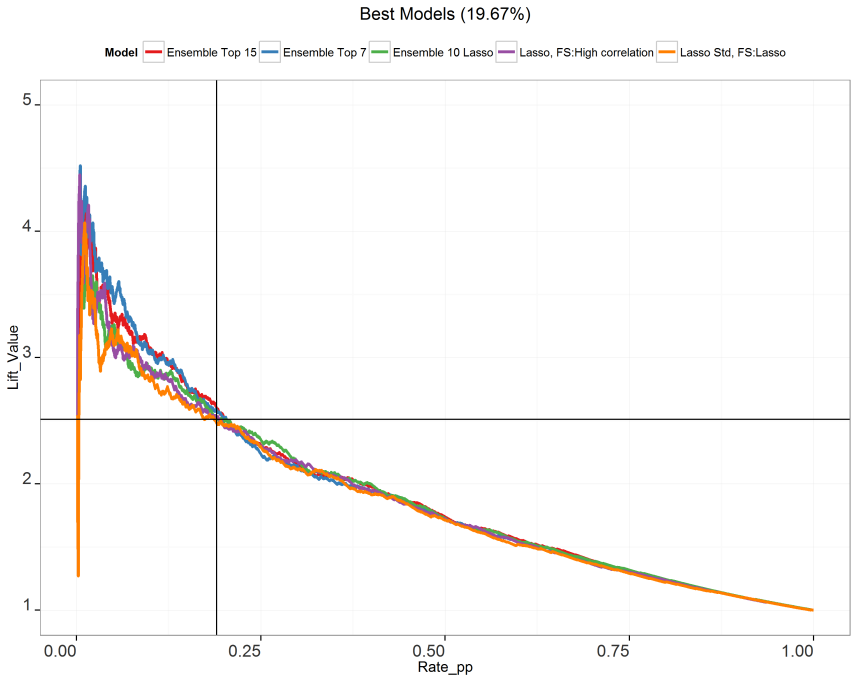
In addition to the "ensembles" of Random Forests, SVMs, and GBDTs used to generate ranked predictions, we induced ensembles combining multiple modeling techniques. These ensembles were created by noting models that had high lifts and combining these high-performing models together. We created three final ensembles: "Ensemble 7" and "Ensemble 15" were the models (or groups of models in the case of Random Forests, SVMs, and GBDTs) with the highest lifts (7 highest and 15 highest, respectively) at the 8.3% cutoff, and "Ensemble 10 Lasso" was the model with the highest lift at the 19.67% cutoff (which happened to all be Lasso). See Table 7 for more details about the individual models in the ensembles. These ensembles performed very well, and had the added benefit of allowing SVMs and GBMs to be a part of models that could be evaluated at the 19.67% cutoff.

6.10 Final Model

Lift charts were made of the best 5 models at the 8.3% cutoff (Figure 3a), and the 19.67% cutoff (Figure 3b). After evaluating all of our models (both qualitatively and quantitatively), including the ensembles, we chose two best models. One model, the Random Forest of all 25 datasets, had the highest lift (3.17) at the 8.3% cutoff (Table 5). The other model, "Ensemble 15", had the highest lift (2.55) at the 19.67% cutoff.



(a) Lift Plots of the 5 Models with the Highest Lift at the 8.3% Cutoff



(b) Lift Plots of the 5 Models with the Highest Lift at the 19.67% Cutoff

Figure 3: Lift Plots of the Best Models on Test Data. The horizontal line and vertical line intersect at the mean lift for all models on the plot at the target cutoff percentage.

Feature Selection	Transformation	Logistic Regression	Lasso Logistic Regression	Ridge Logistic Regression	Random Forests
PCA	**PCA**	0.793	0.804	0.809	0.802
None	**None**	0.697	0.811	0.808	0.821
	Standardized	0.697	0.812	0.803	0.818
	Log	0.504	0.817	0.803	0.814
	Binary	0.657	0.816	0.798	0.814
High Correlation ($\rho = 0.8$)	**None**	0.487	0.813	0.809	0.818
	Standardized	0.487	0.811	0.812	0.817
	Log	0.601	0.813	0.799	0.819
	Binary	0.632	0.811	0.799	0.816
Low Correlation ($\rho = 0.25$)	**None**	0.420	0.817	0.803	0.820
	Standardized	0.420	0.806	0.797	0.820
	Log	0.276	0.813	0.803	0.822
	Binary	0.329	0.812	0.804	0.818
CFS	**None**	0.551	0.817	0.801	0.811
	Standardized	0.551	0.814	0.803	0.809
	Log	0.721	0.812	0.803	0.811
	Binary	0.741	0.813	0.806	0.813
Random Forest	**None**	0.197	0.814	0.807	0.819
	Standardized	0.197	0.812	0.812	0.819
	Log	0.197	0.811	0.806	0.818
	Binary	0.311	0.813	0.804	0.817
Lasso	**None**	0.758	0.815	0.807	0.822
	Standardized	0.758	0.815	0.811	0.818
	Log	0.233	0.817	0.807	0.821
	Binary	0.639	0.812	0.793	0.817

Table 4: Prediction Accuracies (Test Sets) of 4 Modeling Techniques on all Datasets in Table 1.

No.	Model	Transformation	Feature Selection	Lift(8.3%)
1	Ensemble Random Forest 25			3.174
2	Ensemble 15			3.14
3	Ensemble 7			3.14
4	Ensemble Random Forest 15			3.14
5	Ensemble GBM 9			3.14
6	Ensemble Random Forest 12			3.071
7	Lasso	None	High Correlation	3.037
8	Lasso	None	Lasso	3.037
9	Lasso	Standardize	Lasso	3.003
10	Lasso	Log		2.969
11	Ensemble Random Forest 9			2.969
12	Lasso	None	CFS	2.935
13	Ensemble GBM 6			2.935
14	Lasso	Log	Lasso	2.901
15	Ensemble SVM 9			2.901
16	Ensemble 10 Lasso			2.867
17	Lasso	Standardize	CFS	2.832
18	Lasso	None		2.832
19	Ridge	None		2.832
20	Lasso	Binary	High Correlation	2.832
21	Lasso	Log	CFS	2.832
22	Lasso	Binary	Lasso	2.832
23	Lasso	Binary		2.832
24	Ensemble SVM 6			2.832
25	Ridge	PCA	PCA	2.798

Table 5: Top 25 Lifts (on Test Data) at 8.3% of Predictions (After Ranking Predictions by Confidence of Patient Having Diabetes). Ensemble composition detailed in Table 7. Additional Results in `completeLifts.xlsx`.

No.	Model	Transformation	Feature Selection	Lift (19.67%)
1	Ensemble 15			2.557
2	Ensemble 7			2.542
3	Ensemble 10 Lasso			2.514
4	Lasso	None	High Correlation	2.499
5	Lasso	Standardized	Lasso	2.485
6	Lasso	None	CFS	2.471
7	Lasso	Standardized	High Correlation	2.456
8	Lasso	None	Lasso	2.456
9	Lasso	Standardized	CFS	2.442
10	Lasso	Log		2.427
11	Lasso	Log	Lasso	2.427
12	Lasso	None		2.399
13	Lasso	Log	High Correlation	2.399
14	Lasso	Standardized		2.384
15	Ridge	None		2.384
16	Ridge	Standardized	CFS	2.37
17	Ridge	None	Lasso	2.37
18	Lasso	Binary	High Correlation	2.341
19	Ridge	PCA	Principal Components	2.341
20	Ridge	None	High Correlation	2.327
21	Ridge	Binary	High Correlation	2.327
22	Ridge	Standardized	Lasso	2.327
23	Lasso	Log	CFS	2.313
24	Lasso	Binary	Lasso	2.313
25	Ridge	Standardized		2.313

Table 6: Top 25 Lifts (on Test Data) at 19.67% of Predictions (After Ranking Predictions by Confidence of Patient Having Diabetes). Ensemble composition detailed in Table 7. Additional Results in `completeLifts.xlsx`.

No.	Ensemble	Components
1	Ensemble Random Forest 25	All 25 datasets in Table 1
2	Ensemble Random Forest 15	Same as above excluding PCA, all FS:Low Correlation ($\rho = 0.25$), all T:Binary
3	Ensemble Random Forest 12	Same as above row, excluding all full-featured (p=980) datasets
4	Ensemble Random Forest 9	Same as above row, excluding all FS:High Correlation ($\rho = 0.8$) datasets
5	Ensemble SVM 9	Same as above row
6	Ensemble GBM 9	Same as above row
7	Ensemble SVM 6	Same as above row, excluding all FS:Random Forests
8	Ensemble GBM 6	Same as above row
9	Ensemble Top 7	Ensemble Random Forest 25, Ensemble Random Forest 15, Ensemble GBM 9, Ensemble Random Forest 12, Lasso T:None FS:High Correlation, Lasso T:None FS:Lasso, Lasso T:Standardize FS:Lasso
10	Ensemble Top 15	Same as above + Lasso T:None FS:Log, Ensemble Random Forest 9, Lasso T:None FS:CFS, Ensemble GBM 6, Lasso T:Log FS:Lasso, Ensemble SVM 9, Lasso T:Standardize FS:CFS, Lasso T:None FS:None
11	Top 10 Lasso	Lasso T:None FS:High Correlation ($\rho = 0.8$), Lasso T:Standardize FS:Lasso, Lasso T:None FS:CFS, Lasso T:None FS:Lasso, Lasso T:Standardize FS:High Correlation ($\rho = 0.8$), Lasso T:Standardize FS:CFS, Lasso T:Log FS:None, Lasso T:Log FS:Lasso, Lasso T:None FS:None, Lasso T:Log FS:High Correlation ($\rho = 0.8$)

Table 7: Ensemble Models Considered. FS: means feature set and T: means Numeric Transformation

7 Final Evaluation of Models on Holdout Data

Having selected our favored models (Random Forests of all 25 datasets for the 8.3% cutoff, and "Ensemble 15" for the 19.67% cutoff), we rebuilt them using both the training and test patients (a combined total of 7000 patients). The Lasso models were then tuned again tuned using 5-fold cross validation to select the optimal λ, but the SVM and GBM were not retuned due to time constraints.

We then evaluated these final models. The 25 Random Forests had a lift of 3.63 at a cutoff of 8.3% of ranked predictions. "Ensemble 15", which was evaluated at a cutoff of 17.97% of predictions to match the prevalence of diabetes in the holdout data, had a lift of 3.01. Both of these lifts were higher than in the test data, likely because they had more samples to train on. Lift plots for these models are in Figure 4.

We believe the model performance is easily explained by the characteristics of the models. Both Lasso and Random Forests are techniques suited to sorting through large sets of features. The Random Forest model performed well, but likely under-predicted at the higher cutoff percentages (19.67% and 17.97%), which in turn made it more accurate at the lower cutoff percentage (8.3%). Lasso (which made up most of "Ensemble 15") did not under-predict at the higher cutoff percentage, but this likely led to more inaccurate predictions at the lower cutoff percentage. Both models benefitted from the variation provided by feature selection, and "Ensemble 15" likely benefitted from the variation of being composed of multiple modeling techniques.

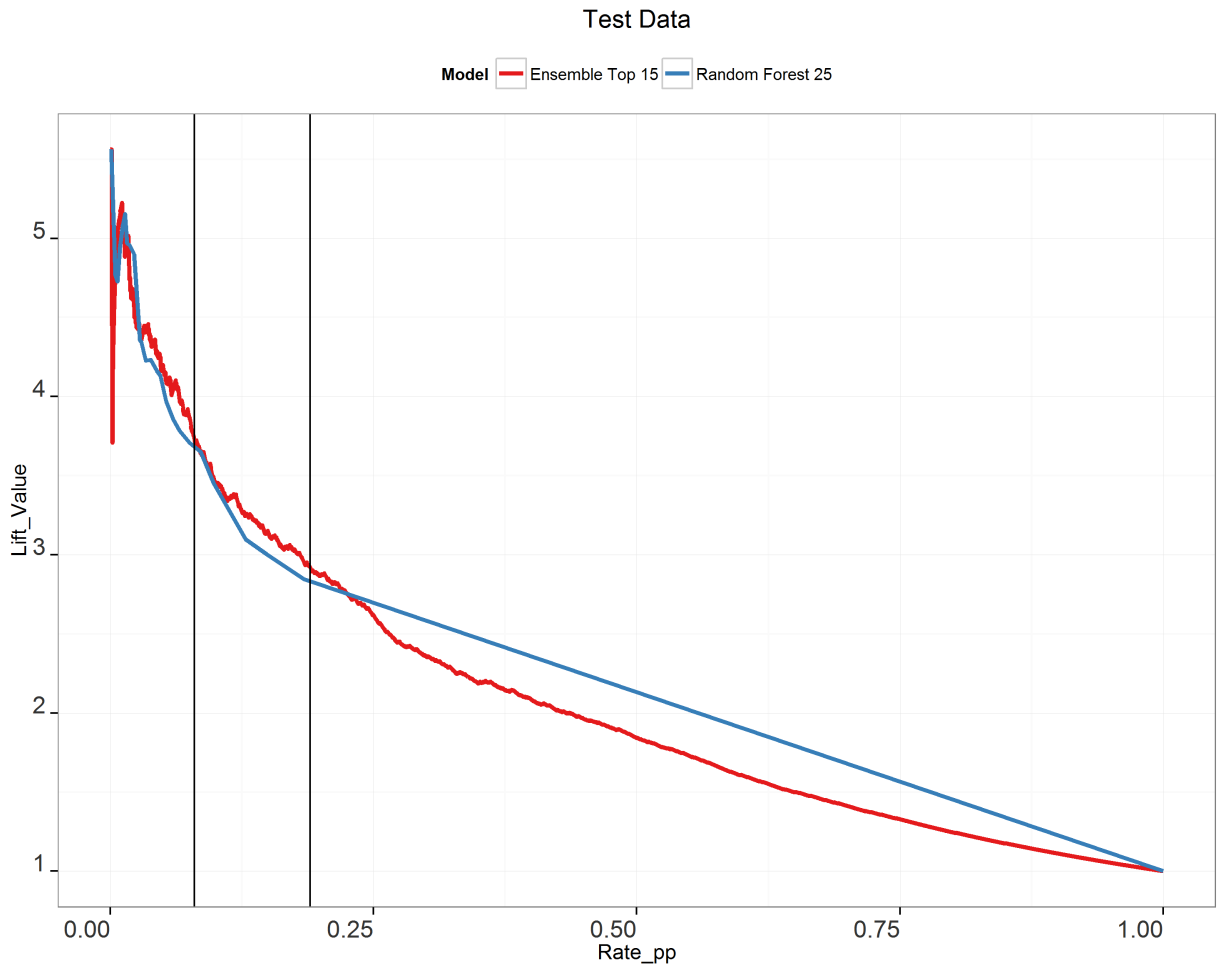


Figure 4: Lift Plots of "Random Forest 25" (attempt to maximize lift at rate of positive predictions = 8.3%) and "Ensemble 15" (attempt to maximize lift at rate of positive predictions = 17.97%) on Evaluation (Holdout) Data.

8 Key Findings

We initially started with a total of 980 features, but using different feature selection techniques we were able to reduce this count without losing (and sometimes gaining) predictive power. With Correlation Feature Selection we reduced 980 features to 13 (1.3% of the original feature count) without losing much predictor power. Despite having only 13 features, the CFS datasets outperformed a pair-wise correlation reduction ($\rho = 0.25$) selection with 421 features, a Random Forests importance-based selection with 50 features, and the original dataset with 980 features (see Table 4). This is likely because CFS emphasizes the importance of certain factors like age, cholesterol, BMI, renal disease, high blood pressure, and blood disorders–all factors strongly related to diabetes. While the feature count is low, the power of the features is very high and the "fat" is non-existent.

Despite the power of feature selection, when it comes to building the best models ensembles work better than individual models (Tables 5 and 6). Multiple datasets can add to the variation of ensembles (thus reducing the overall error), and feature selection provides a means of creating these multiple datasets. In our results, we saw that even basic ensembles consisting of means of discrete predictions performed well. We also saw that multiple modeling techniques combine together to create the most effective ensembles.

Additionally, we showed that computationally demanding modeling techniques like SVMs and GBDTs gave decent results on datasets of reduced features (Table 5). With proper feature selection, a great deal of computational time could be saved without any reduction in predictive power.

Lastly, we confirmed domain-based assumptions about the nature of diabetes through the use of feature selection. It is clear from Table 2 that feature selection provided relatively logical features for this diabetes identification problem. However, feature selection also highlighted some less-than-obvious features, like range of weights, body temperature, and allergy medication count. There may be an application of feature selection as a form of learning, where the goal is not building better models, but learning more about datasets–with possible real-world applications

9 Limitations, Shortcomings, Compromises, and Lessons Learned

9.1 Sample Mean vs. Population Mean

The patients in the Kaggle dataset had a rate of diabetes 19.7%, but the US population has a rate of diabetes closer to 8.3%. We recorded lifts at ranked prediction rates of 8.3% and 19.7% to determine potential model effectiveness at both the population and sample diabetes prevalence rates, but without further information there is little we can do to determine how applicable our model is to the population outside the sample.

9.2 Failure to Stay Simple or Build a Proper Baseline Model

Our initial inclination was to generate as many feature-increasing data transformations as possible, then apply many different feature selection criteria to those datasets in order to find the best combination of features to induce models from. While this approach yielded several datasets and models to choose from, we should have started with a relatively simple dataset and model in order to have a true standard to improve upon instead of the nave 80% diabetes prevalence rate. Having a baseline model would also allow us to compare increases in explanatory power with increases in model complexity. We did attempt to create a baseline model with Logistic Regression, but the test performance of the Logistic Regression model was awful.

9.3 Data Leakage from Diabetes Features

In the `SyncDiagnosis` table, several ICD9 codes indicated complications from diabetes. While these made good predictors in the training model, patient information about these conditions should not have been included in the Kaggle data–Kaggle removed all diagnosis codes directly referencing type 2 diabetes, all medications for treating diabetes, and all lab results showing the presence of diabetes, but left these complications in. More generally, this information did not advance any knowledge about what conditions are have high co-occurance with diabetes, as diabetes complications are already known to be associated with diabetes.

9.4 Waiting to Look at Kaggle Winners was Smart

The winners and runners-up of the Kaggle competition used significantly different methods than our own to build their prediction models–methods on domain knowledge of the medical conditions associated with diabetes. Our approach was markedly different, relying on bulk feature generation followed by automated feature selection. While there was significant overlap in the information captured by the different approaches, waiting to read their documentation until after we had created several predictor models led us to invent many predictor features (like our diagnoses scores) which the winners had not considered, but which contained significant predictive power. Their models also contained some predictors we did not think of, but due to the veracity of our dataset and model creation, our feature creation was more voluminous and varied than the winners, and we learned significantly more by thinking up our own features than we would have by copying and improving on the work of others.

Towards the end of the project, we went back and added some features to our data based on feedback and the Kaggle reports, specifically a total count of physician visits and a transformation of a patient's home state.

9.5 Kaggle Projects are Difficult

Kaggle projects are problems where it is difficult to find solutions that are significantly better than the nave method. Kaggle is built around the idea of letting the best and brightest work to solve a problem, and this dynamic requires challenging problems to keep interest high. We did not expect this project to be as hard as it was, and we need to keep Kaggle's difficulty in mind when selecting projects in the future. That being said, the difficulty of the project led to most of our learning, and we are significantly better experienced in data cleaning and modeling due to attempting this project.

9.6 Too Many Datasets with too Few Benefits

Our instinct to constantly create more datasets for testing left us with many more datasets than are necessary for the boost given in model accuracy. Our strategy led to creating many datasets created from the same data, hoping that somewhat small changes would reveal new insights. However, we were not able to glean much new information from the large number datasets (and features) we generated. While this exercise was manageable in a school setting, in the real world where datasets are larger and deadlines come faster, this strategy could have easily backfired and left us with too many datasets and too few meaningful results.

9.7 Binarization Was a Bust

As can be seen in Table 4, the binarized datasets performed worse than the untransformed, standardized, and log-transformed datasets for almost all feature selection and modeling techniques. This result was not surprising, given that the binary transformation significantly reduces the information gained from many of the features. It is interesting to note that CFS picked a binary feature (cholesterol medication) as the single highest correlated feature with diabetes, but this merely points to the need to try binary features alongside non-binary features, instead of making every feature binary.

9.8 Feature Selection Worked

The model accuracies in predicting diabetes on the datasets with fewer features were comparable to the model accuracies on datasets with more features. This result validates the idea that feature-reduction techniques that seek to preserve information while reducing feature count can be effective, especially when they are preserving information relating to the feature of interest, in this case diabetes condition. It is worth noting that the dimension reduction method that produced the worst results, PCA, was the one preserved information without regards to the variable of interest. Since the datasets contained hundreds of scarce features, and only a few of these were correlated with diabetes, a method like PCA, which does not seek to preserve variance correlated with the feature of interest, is predisposed to produce less accurate models than other data transformations, and even than the untransformed data.

9.9 Differences in Feature Selection

Using several different feature selection methods on the same dataset allowed for easy comparison of the different biases inherent in each selection criteria. In this project, it became obvious that the Random Forest feature selection favored numeric features over binary features, as the numeric features had more points to divide by compared to the binary features (and thus more opportunities

to reduce node impurity). Likewise, the Lasso selector favored variables with large magnitudes, since predictors requiring smaller coefficients would tend to be favored (which is a reason to standardize data before running Lasso). We failed to standardized our Lasso data at first, but it seemed that the smart code behind the `glmnet` package which dealt with this issue saved us from going down a wrong path.

9.10 Lift vs. Accuracy

Initially, we were evaluating all of our models on their accuracy on a test set. Based on these results, we assembled an ensemble of models that had good test accuracy, then evaluated them on the holdout set. As Professor Ghosh pointed out, the metric we should have been more concerned about was lift, especially for the patients predicted most likely to be diabetic. Based on this feedback, we re-evaluated our models, and our final ensemble was composed of models which each had impressive lift charts on their own. There is still the possibility, however, that our models could have been improved by paying more attention to lift from the beginning. Some of the best-performing models were Random Forests, and while the accuracy of these models stopped growing before our selected number of trees (300), we never plotted lift versus tree number. Given how the Random Forests models performed so well in aggregate across our datasets (which is essentially a case of more trees with a weighted dataset), it is possible we could have increased lift by building more trees.

9.11 The Pitfalls of Excel

Some tables were transformed in Excel rather than R, due to their small dimensions and user preference. While this initially seemed to be an efficient move, the pitfalls of manipulating data in Excel soon became obvious: data manipulations done in Excel were not easily replicated, whereas a code trail existed in R. Additionally, mixups occurred in Excel which required considerable time to repair, whereas in R it would have meant a code change and a re-run. Lastly, it was much more difficult to handle missing values in Excel than in R.

9.12 Throwing out Good Data vs. Keeping Bad Data

Filter feature selection methods, while useful for feature reduction, do not select for predictive power. Wrapper methods, however, select features based mainly on their predictive power. Because of this difference, there is a risk of throwing out low-information but high-predictive-power data when using filter feature selection methods. In this project, this risk was realized with the Low Correlation ($\rho = 0.25$) dataset, which induced significantly worse models than the High-Correlation ($\rho = 0.80$) dataset and other datasets with far fewer features.

9.13 Missing "Test" Data and Little Cross Validation

The final step before modeling was to create a training set and test sets. While Kaggle had a set of test data, they hid the test data when the competition ended. Since we needed a standard to measure our final models against, we set aside 2948 patient records as a "holdout" to determine model performance on untouched data. Of the 7000 remaining records, 5200 were designated as training data and 1800 used as test data. While cross-validation would have been a more robust way to build models, several models were very cumbersome to evaluate with cross-validation in R, and we preferred to spend our time testing more models instead of refining only a few, especially since we had 25 datasets to test the models on.

10 Conclusion

Our group took 17 tables containing a wide variety of healthcare information about 9948 patients and combined it into one large dataset with 980 features per patient. We then created 24 distinct versions of this dataset, applying six different methods of feature selection to the data and 3 different methods of numeric transformation. We then applied a variety of modeling techniques, which we evaluated using accuracy and lift. We found our best-performing models were two ensembles, and they achieved instantaneous lift ratios of 3.17 (at a positive prediction rate of 8.3%) and 3.012 (at a positive prediction rate of 17.97%) on unseen data. While we cannot submit these models to Kaggle since the competition is closed, we have accomplished our goal of becoming familiar with healthcare data and applying advanced predictive analytics to a complex problem which we did not possess significant domain knowledge of. We also showed the general efficacy of feature selection, both as a basis for model induction and as a technique for exploring datasets.

References

CB Online Staff. PR has highest rate of diabetes in us. *Caribbean Business*, November 2012. URL `http://www.caribbeanbusinesspr.com/news/pr-has-highest-rate-of-diabetes-in-us-78613.html`.

Centers for Disease Control and Prevention. International classification of diseases, ninth revision (ICD-9): Classification of diseases, functioning, and disability, September 2009. URL `http://www.cdc.gov/nchs/icd/icd9.htm`. US Department of Health and Human Services.

Centers for Disease Control and Prevention. Diabetes report card, 2012. US Department of Health and Human Services.

T. H. Cheng, C. P. Wei, and V. S. Tseng. Feature selection for medical data mining: Comparisons of expert judgment and automatic approaches. *CBMS*, pages 165–170, 2006.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. pages 1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`.

Mark A. Hall and Lloyd A. Smith. Feature subset selection: A correlation based filter approach. 1997. URL `http://researchcommons.waikato.ac.nz/handle/10289/1515`. The Library Consortium of New Zealand.

Kaggle.com. Practice fusion diabetes classification, May 2014. URL `http://www.kaggle.com/c/pf2012-diabetes`.

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

Max Kuhn, Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, and the R Core Team. *caret: Classification and Regression Training*, 2014. URL `http://CRAN.R-project.org/package=caret`. R package version 6.0-24.

Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3): 18–22, 2002. URL `http://CRAN.R-project.org/doc/Rnews/`.

Office of the National Coordinator for Health Information Technology. Test procedure for 170.302 (g) smoking status, 2010. US Department of Health and Human Services.

Practice Fusion. Free, web-based electronic health record (EHR), May 2014. URL `http://www.practicefusion.com`.

RxList. Rxlist: The internet drug index, May 2014. URL `http://www.rxlist.com/script/main/hp.asp`.

T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer. Rocr: visualizing classifier performance in r. *Bioinformatics*, 21(20):7881, 2005. URL `http://rocr.bioinf.mpi-sb.mpg.de`.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. URL `http://statweb.stanford.edu/ tibs/lasso/lasso.pdf`. Volume 58, Issue 1.

WebMD. Webmd: Better information, better health, May 2014. URL `http://www.webmd.com/`.