Brief Analysis of the National Survey on Drug Use and Health, 2012

Jason E. Piccone, Ph.D.

Abstract

This analysis demonstrates data science methodologies and presents insights on a national health survey. The National Survey on Drug Use and Health is a state-stratified sample of residents of the United States, age 12 and over, and was conducted by the United States Department of Health Services. The raw data consisted of 55,268 respondents and 3,119 features. The current demonstration begins (part 1) with data acquisition, cleaning, and visualization. It continues (part 2) with machine learning algorithms generating predictive benchmarks for select outcome variables and feature reduction strategies (principle components analysis), and concludes (part 3) with attempts at improving the benchmark analysis through stacked generalization ensemble methods and follow up analysis.

Brief Analysis of the National Survey on Drug Use and Health, 2012

Understanding the complex relationship between drug use and demographics with health and criminal outcomes is exceedingly important. One goal of this report is to explore these relationships, utilizing a national survey of residents of the United States. The primary goal, however, is to demonstrate the data science approach to such a dataset. I chose two outcome variables (overall health and whether the respondent has been booked/arrested in the past twelve months) as examples and experiment with different machine learning models to assess those outcomes.

I chose a database from the National Survey on Drug Use and Health, 2012 (ICPSR 34933) from: http://www.icpsr.umich.edu/icpsrweb/content/samhda/dataportal.html. This database was selected for two major reasons, (a) I'm interested in health data. I based my Ph.D. dissertation on a large scale dietary health intervention program and have experience analyzing several other health-related databases, and (b) it met my general desires in regards to its dimensions. The raw data consisted of 55,268 respondents and included 3,119 features (I'll also refer them as variables). While it would be fun to have more respondents, we do have enough to perform complex predictive analyses. The analyses I present would apply well to much larger datasets, as Python (the language that I use throughout) is efficient in dealing with bulky data. In addition, with over 3,000 features, it provides an opportunity to show how to deal with relatively high dimensionality. This report is divided into three segments, (a) data acquisition and cleaning, (b) basic predictive analysis, (c) enhanced predictive analysis. If you are interested in reviewing the complete code, I have uploading scripts associated with each of these three segments. The scripts are ordered in line with the following text, which I hope will make it easy to follow along.

1. Data Acquisition and Cleaning

Introduction

The survey was conducted through a stratified sample such that samples were taken in proportion to the population of each state. Responses were gathered in field interviews and respondents included individuals aged 12 and over across the United States (Morton et al., 2012). The dimensions of the data frame indicate 55,268 respondents and 3,120 features. Assuming we don't lose large portions of the data to missing cases, this data set lends itself well to most analyses. There are enough cases so that we have the power to perform complex analyses, but not so many that it will require significant computing resources. The number of features, relative to the number of cases is quite high, however. We will start by reviewing the features, to both get a feel for how we might limit them, and eventually how to approach analysis. Fortunately a code book is included with this dataset, so the often ambiguous feature names can be discerned, which greatly helped in this process.

I began by reviewing the codebook and running a series of frequencies for each variable. This is a representative question with the corresponding distribution of values:

"Have you ever smoked a cigarette?"

Yes: 27,881

No: 27,387

It turns out that of the 3,120 features, a sizable portion are redundant. For example, there are over twenty five questions regarding cigarette use, including: “Have you ever smoked a cigarette”, “Year of your first cigarette use”, “Month of first cigarette use”, “How many cigarettes smoked in the past 30 days”, “How many cigarettes smoked on average per day”, and so on. I was able to remove many variables based on logic. For instance, many questions were included for the more common substances like smoking and drinking, but only a couple of questions were included for most (of the many) other substances. For each of the many substances for which respondents were questioned, they were all asked “Have you ever used substance X?” with a yes or no response. Since this was standardized across substances, I chose to include this feature (for each of the substances surveyed). In addition, the dataset included the same questions recoded. For example, missing values for a variable may have been imputed in one version of the question. After removing redundant and recoded variables, the number of features was reduced to 254. While this is a substantial reduction, enough features remain to justify the use of feature reduction strategies, such as principle components analysis – which is demonstrated in section two.

Recoding Variables

The dataset required significant recoding. The typical item, for example looked similar to this: “Have you ever, even once, used peyote?”, with response options of: 1 = Yes, 2 = No, 3 = Yes (logically assigned), 91 = never used hallucinogens, 94 = don’t know, 97 = refused to answer. Responses were logically assigned if the respondent indicated elsewhere that they tried the given substance. A score of 91 was given in cases where this question was skipped because the participant had previously indicated that they never tried hallucinogens. In this case I retained responses of 1 and 2, 3 was recoded to 1, 91 was recoded to 2, and 94 and 97 were recoded to missing. That was one example of a great number or recodes necessary. The full documentation of the data recoding can be viewed in the first script file.

Missing Data

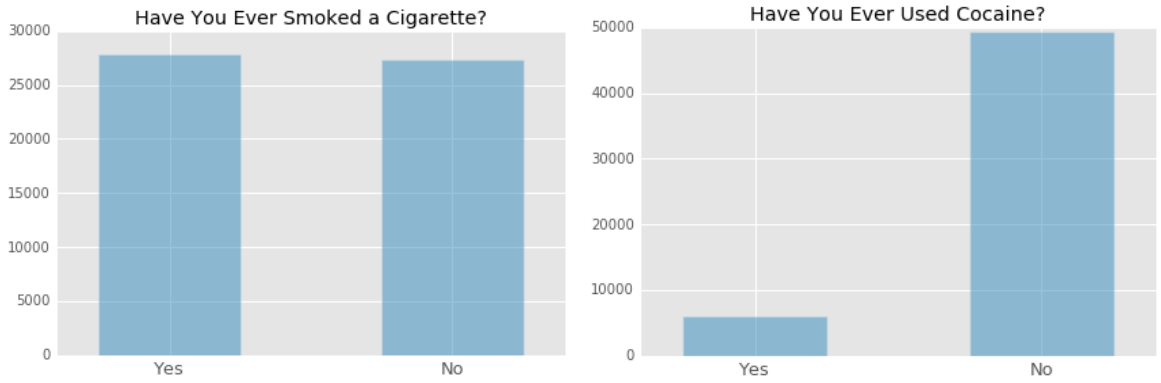
I printed out tables documenting how many missing values are present in each feature. Quite a few variables have significant proportions of missing values. This is in part because some variables, such as “How old were you the first time you tried cocaine” included many respondents who never tried cocaine, so were coded as missing. These missing values will have to be dealt with before we attempt predictive modeling. My first inclination is to exclude such variables from the primary analyses but leave open the option of using them for follow-up analysis. For instance, if we find that people who have tried cocaine are at greater risk for being arrested, we might then look at those individuals more closely, where the age of first use variable will be useful. It also became apparent that missing values varied by respondent age. The reason is that many questions were tailored to youths (such as questions about their current education experiences) that adults were not asked.

One option to deal with missing values is to impute them. This is a fairly common process that entails predicting the missing values based on the non-missing values. This is easy to accomplish, assuming that the non-missing values are strong enough predictors for the missing variables. I am not going to do this as many of the missing values are logically missing. If an individual is not in school, for example, it makes little sense to impute a value for how many extracurricular activities they are involved in at school. There are only a small handful of legitimate missing values that would be candidates for imputation, which is simply not worth the time in this case. Imputing missing values also introduces error in the equation (as any predictive equation will be less than perfect) and we have a large enough

sample that losing a small fraction is not inhibitive to our goals. If the number of missing values was greater and we decided to omit such cases from analysis, it would be wise to determine if cases with missing data differed from cases without. In other words, if people who didn't answer questions differed from those who answered all of the questions, then we would suspect that the results of our analyses would not generalize to such individuals.

Visualizing Data

Prior to serious data analysis, it is important to become intimate with your data. Seriously, I choose the word intimate for a reason. You want to look at it from multiple directions, plot the relationship between variables and so on. I'm at my best when I dive entirely into a dataset. I wake up to dreams of the data, have ideas about different ways to explore the data in the shower, and so on. Examining plots is especially important after the massive recoding we engaged in previously to ensure no errors were made. It might be important to remember later that many of the variables are extremely imbalanced. For example, only about 6,000 people out of about 56,000 have ever used cocaine. For some substances the ratio is even more imbalanced. This may be important to remember because if we attempt to predict one of these variables, we want to use a metric other than simple prediction accuracy. If 95% of the population never used drug X, then it would be easy to achieve 95% predictive accuracy simply by predicting "no" for everyone. Instead we should use AUC (area under the curve) as our metric of model precision (which we'll discuss further later). Let's take a look at a couple example plots:



One final step before we begin analysis is to standardize the variables. Standardization is the process of mean centering each quantitative variable at zero with standard deviations of 1. This is important so that each variable is weighted equally when included in an analysis. For instance, a variable with a range of values from 0-100 might be given more weight in an analysis than another variable with a range of values from 0-1. Scikit-learn (the primary machine learning package for Python) has a nice standardization feature, but as far as I can tell it only works when a data frame has no missing values; therefore, standardization was calculated manually.

Similarly, categorical data requires recoding. Categorical data with more than 2 levels needs to be dummy coded. For example, the question regarding marital status has four levels: 1=married, 2=widowed, 3=divorced/separated, 4=never been married. Assigning numerical values incorrectly implies that 1 is less than 4 in terms of married-ness. This variable should instead be recoded into four variables with binary values. So the first of the following cases is married, and the second is divorced:

	Married	Widowed	Divorced/Separated	Never Married
Case1:	1	0	0	0
Case2:	0	0	1	0

Technically, you could code this into just three variables – where respondents would be coded as never married if they received zeros on the other three variables. This method is preferred in a sense that it saves a little bit of statistical power, but for ease of interpretation (and since we have plenty of power with over 50,000 respondents), I chose to use the full coding method. The categorical features with only two levels require no dummy coding. Fortunately, the vast majority of categorical variables were binary and only five features required recoding. It is also a good idea to examine collinearity among features. For instance, if several variables are highly correlated, they become redundant and their inclusion reduces statistical power. While some variables were strongly correlated, the issue did not appear to be severe enough to necessitate action.

Conclusion

Preparing data is typically the most time-consuming aspect of any data science project. It is tempting to skip ahead to the juicy analyses, but without proper preparation, those analyses would likely be meaningless. The current dataset required substantial cleaning and recoding, and it is comforting to proceed to the next stage knowing that it has been managed thoroughly (and intimately).

2. Basic Predictive Analysis

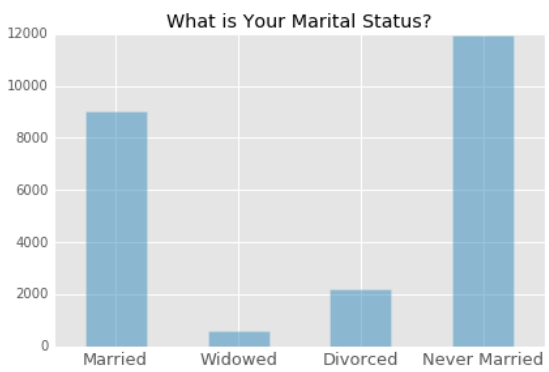
Introduction

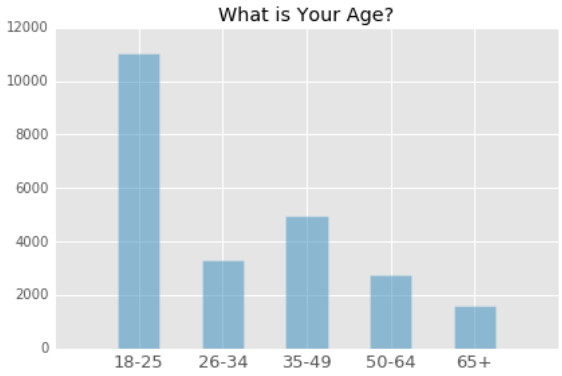
In the previous section we prepared the dataset for analysis. In doing so and investigating the pattern of missing data, it became apparent that analysis should proceed in two levels: by youths and by adults. For the sake of brevity, we will only explore the adult (older than 17) population. The following presents predictive modeling for two outcome variables.

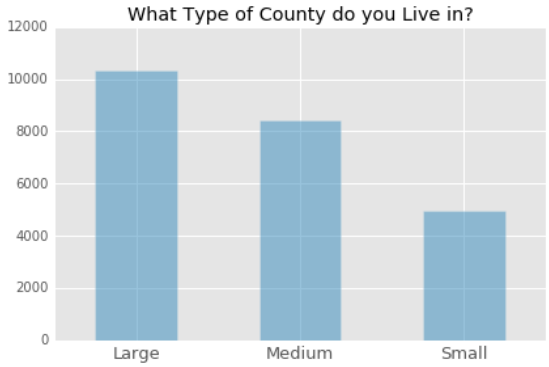
Explore Basic Adult Respondent Information

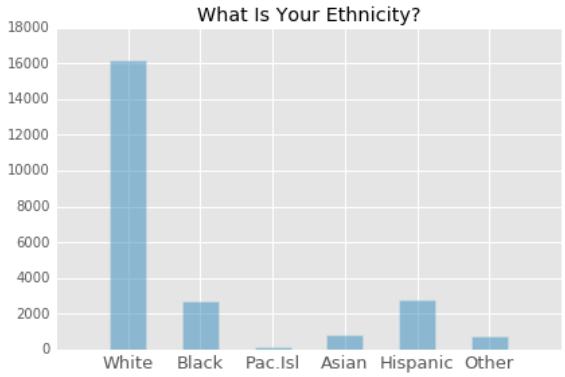
After separating the data set into adults and youths, I again explored the distribution of missing values. A large scale, supervised learning predictive model includes throwing in all of the features to determine which are important in predicting the variable of interest. Missing values must be handled prior to these analyses. I first removed variables with large amounts of missing data from the dataset (dropping the number of features from 275 to 126) and then removed the individual cases with missing values within the remaining 126 features (dropping from 37,869 to 23,699 respondents).

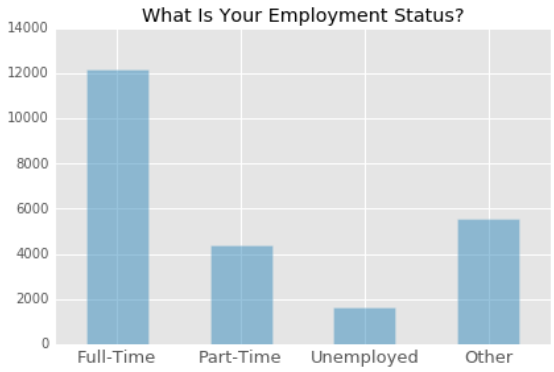
It’s always a good idea to have a good handle on basic demographic information. Now that the adult sample has been pared down, let’s explore these variables (bar charts are in blue, histograms in red):

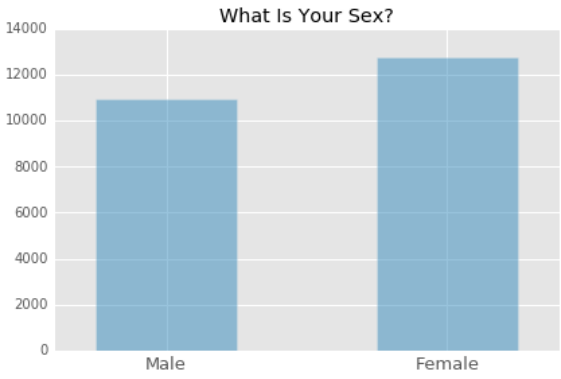

What is Your Marital Status?
12000
10000
8000
6000
4000
2000
0
Married
Widowed
Divorced
Never Married

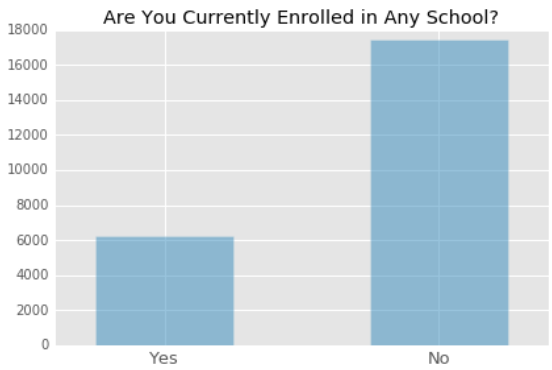

What is Your Age?
12000
10000
8000
6000
4000
2000
0
18-25
26-34
35-49
50-64
65+

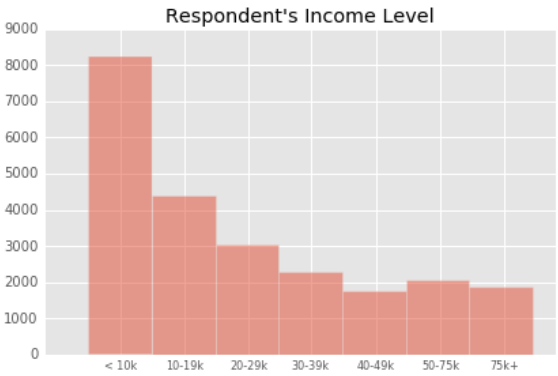

What Type of County do you Live in?
12000
10000
8000
6000
4000
2000
0
Large
Medium
Small


What Is Your Ethnicity?
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
White
Black
Pac.Isl
Asian
Hispanic
Other


What Is Your Employment Status?
14000
12000
10000
8000
6000
4000
2000
0
Full-Time
Part-Time
Unemployed
Other


What Is Your Sex?
14000
12000
10000
8000
6000
4000
2000
0
Male
Female

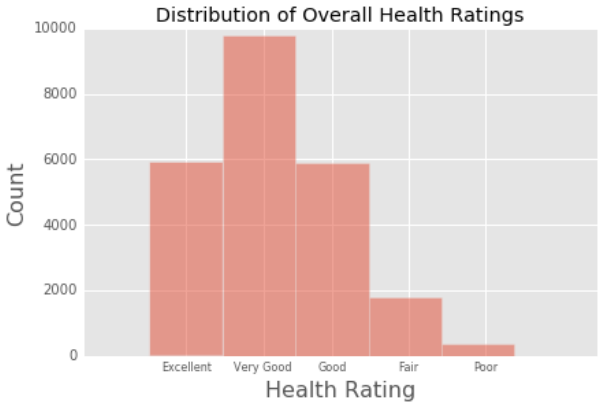

Are You Currently Enrolled in Any School?
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Yes
No

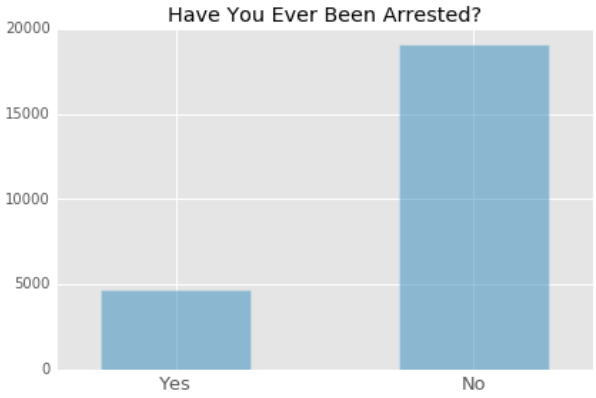

Respondent's Income Level
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
< 10k
10-19k
20-29k
30-39k
40-49k
50-75k
75k+

The most striking information here is that a vast majority of respondents remaining for analysis are young and white. In addition, most appear to be early in their careers – since a large majority are full-time employed and a large majority make $10,000 or less per year (note that the "other" category in employment refers to cases such as individuals who are no longer in the workforce). In addition, most of the respondents live in more populated counties. I can't help but lament at this point that the data did not include zip codes or other useful location information. There are a lot of neat visual representations you can compose by transposing data on maps.

Exploring the Outcome Measures

There were many outcome measures to choose, but I wanted to choose variables that were interesting and I wanted both a continuous (overall health) and a categorical (have you ever been arrested?) variable. Continuous variables are regression problems while categorical variables are classification problems – requiring different statistical methodologies. It is important to understand the distributions for the outcome measures. Below we can see that overall health is positively skewed, so that most people are distributed around 'very good' while a few scores tail off to the right. The variable of "have you ever been arrested" is even more imbalanced as less than 5,000 respondents report ever being arrested while just under 20,000 report never being arrested. It is beyond the scope of this review to speculate on reporting bias, but of course it is possible that people underreport their arrest history. People can be quite reluctant to report socially undesirable information.



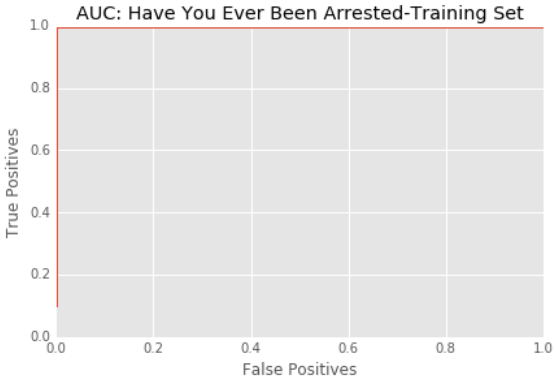


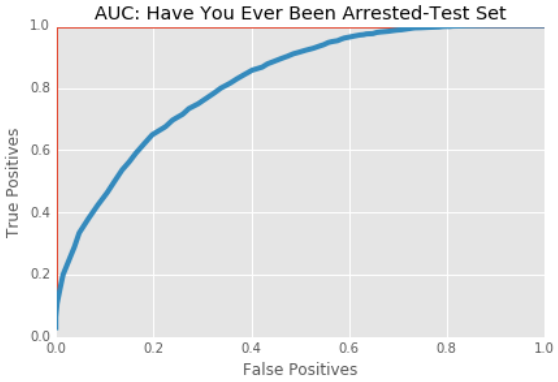
Setting Benchmark Predictive Models

Random forest analyses are a standard choice to determine a strong benchmark. They are easy to compute, require little tuning, and are extremely powerful. Random forests are referred to as an ensemble method because they consist of computing a set of individual analyses on multiple subsets of the data and then combining the results into an overall model. Prior to running these analyses, I divided the data into a training set (75%) and a test set (25%).

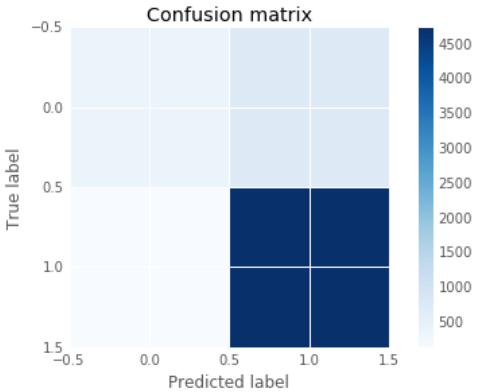
The random forest analysis for the outcome measure of 'Have you ever been arrested' was computed with the number of trees in the ensemble set to 100. This is a number we can experiment with to find the optimal model. You generally reach an ideal trade off as the number of trees increases, predictive precision increases on the training set, but with too many trees, you run the risk of overfitting – and thus performing poorly on new data (such as our test set). But again, we are simply establishing a benchmark with this initial analysis. We evaluate this model based on the area under the curve (AUC)

metric, which determines the ability of a model to distinguish between cases that are accurately vs. cases that are falsely classified. The greater the AUC, the more accurate the model is. In looking at the figures below, a useless model would be one with a line that stretched directly from the bottom left corner to the upper right corner. This would be an AUC of 0.50, where we had a 50% chance of accurately classifying each case. The AUC, when tested on the training sample (the same data we modeled the random forest with) = 1.00. The line cannot be seen because it hugs the y axis up and the top axis across. This demonstrates how powerful the random forest is, it trained the data perfectly, with the parameters given. This also demonstrates the risk of overfitting, which we have to assume we've done. Nevertheless, using this model to predict on the test set indicates an AUC = 0.822, and is displayed in the second figure below. Interestingly, I tested the results when setting the number of trees = 25. The AUC on the training set was still greater than 0.99 but the test set AUC dropped to 0.80.



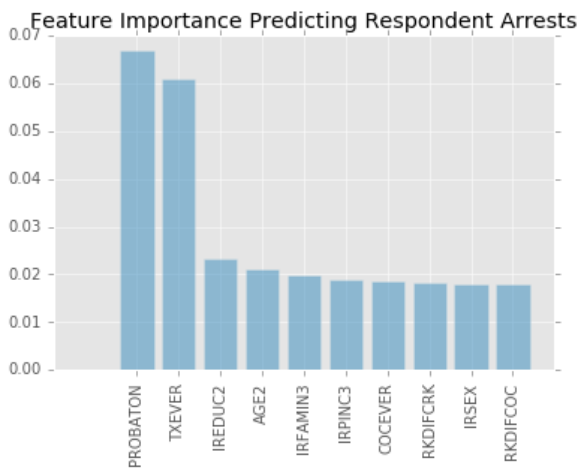


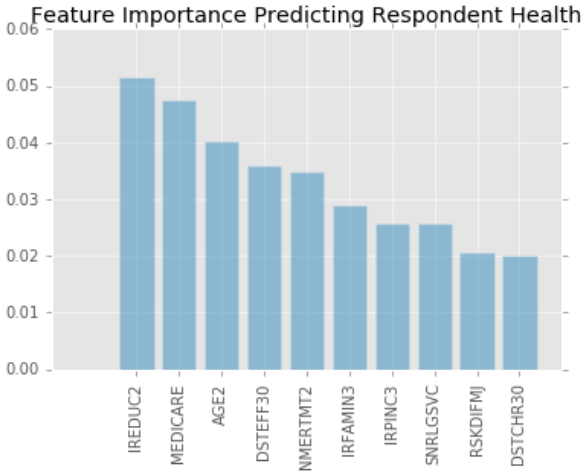
I mentioned previously that the AUC is a more appropriate measure of accuracy since the outcome measure is highly imbalanced. Just for fun, let's take a look at the proportion of correct predictions through a confusion matrix. The confusion matrix for the test set indicates 411 correctly identified positives (arrested), 4,741 correctly identified negatives, 135 incorrectly identified positives, and 739 incorrectly identified negatives, which equates to over 85% accuracy. As can be seen in the confusion matrix plot below, we were extremely good at predicting true negatives – which is not surprising since most people were never arrested. We also predict too many people to have never been arrested who in fact were. Of the 1,150 people who have been arrested, we only correctly predicted 411 (36%) of them. Of the 4,876 individuals who have never been arrested, we correctly predicted 4,741 (97%).



Now we turn to our second outcome of interest, overall health. Since this is a continuous variable, we will be conducting a regression analysis (rather than classification). The difference quite simply is that we're predicting a numeric value rather than class membership. I again ran the random forest with 100 trees. The mean square error (MSE) is the outcome of interest, and the smaller the value represents less prediction error. The square root of the MSE undoes the squared, returning the metric to the same units as the quantity being estimated. In other words, if you are predicting age, and the squared MSE = 1, then the average deviation from the true score across cases would be one year. We are predicting overall health on a quantitative scale where the values don't equate to an easily interpretable scale, however, so we will have a more difficult time interpreting the MSE. Nevertheless, generating a random forest and having a MSE benchmark is important as we attempt to refine our model in future iterations. The root MSE for the random forest predicting overall health = 0.889.

Another great attribute of random forest analyses is that they can produce a feature importance metric. Quite simply, this tells you which features were the greatest contributors to the model. Since we included so many features in each analysis, I printed out the top ten features and plot them below. In predicting whether participants have been arrested, the variables that contribute the most include (in order of importance): (1) whether participants have been on probation in the past twelve months, (2) whether participants have received treatment for drugs or alcohol in their lifetime, (3) education level, (4) age, (5) family income, (6) respondent's income, (7) if the participant has ever tried cocaine, (8) how difficult is it to get crack, (9) sex, (10) how difficult is it to get cocaine. The top ten most important features predicting respondent health include: (1) education level, (2) if respondents are covered by medicare, (3) age, (4) how often you have felt that everything is difficult in the past 30 days, (5) how many times been in the emergency room in the past 12 months, (6) family income, (7) respondent's income, (8) how many religious services attended in the past 12 months, (9) how difficult is it to get marijuana, (10) how often you have felt sad and nothing could cheer you up.

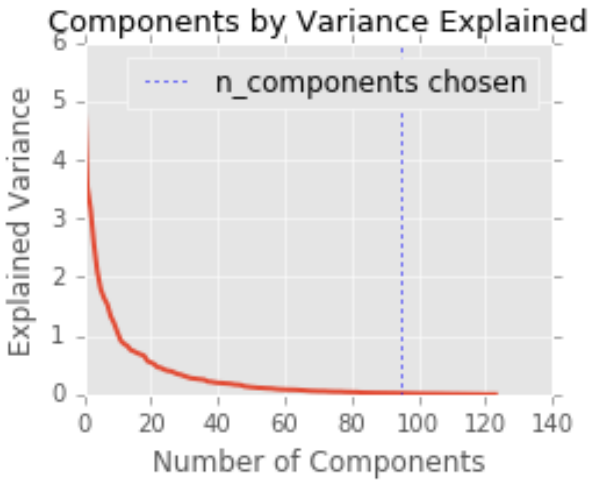




Exploring Data Reduction through Principle Components Analysis

We refer to data for which we have many features as high dimensional. This high dimensionality can be prohibitive in terms of computational complexity. This is more of an issue for big data – where we have somewhere in the order of millions of cases and thousands of features. In such instances, it

may be necessary to reduce the number of features. A principle components analysis finds similarities among features and combines them into uncorrelated components. Thus the number of components will always be less than the number of features. Choosing the number of components to select is not always a straightforward process. I started by viewing figures such as the one below, which shows how much of the total variance is explained by each additional component. We can see that after about the 10th component, the total variance explained by each additional component drops to about 1%, and at about the 25th component less than 0.5% is accounted for. I expected that the ideal cutoff would be around that point, as additional components that account for little variance take away from power while contributing little to the predictive ability of the model. Scikit-learn has a nice feature that tests the predictive ability of the model based on models of different numbers of components. The figure below shows that the optimal number of components is 95.



When we run the same random forest analysis on this model, with 95 principle components predicting arrests, the AUC = 0.797, which is a reduction from the benchmark model of 0.822. In essence, we gained marginal computation efficiency with a small loss of predictive ability. If we started off with 1,250 features instead of 125, this may have been an acceptable trade-off, but for the current purposes, the benchmark model is preferable. Turning to the outcome of overall health, fitting the same model with 95 components increases the root MSE to 0.913 (from the benchmark of 0.889). I honestly expected a greater drop in the model's effectiveness – this demonstration begins to show the efficacy of PCA as the benefits of this approach will likely increase as the size of the dataset also increases. I did try the model with different numbers of components, but 95 exceeded them all.

Conclusion

We explored two outcome measures for the adult sample and established stable benchmark models using random forests. Our model predicting prior arrests had an AUC score = 0.822 while the model predicting overall health had a root MSE = 0.889. We attempted to improve upon these models using PCA and while the predictive efficacy of these models suffered only a small drop, we elected to retain the benchmark models due to the following reasons: (a) we prefer the superior predictive ability, (b) our data is not so large that computation resources are relevant, (c) ease of interpretation. While the benchmark models perform adequately, there is always room for improvement, which we'll address in the final section.

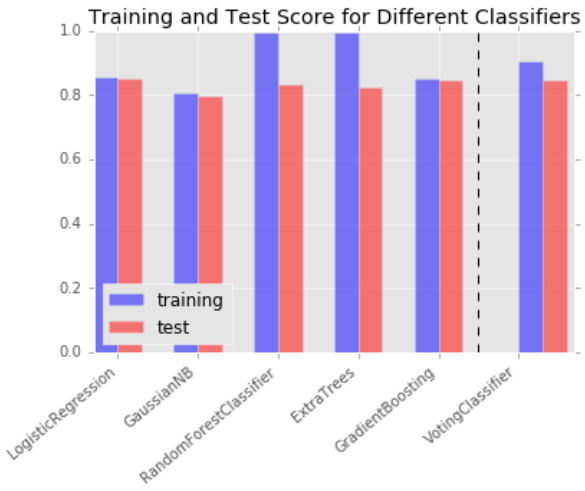
3. Enhanced Predictive Analysis

Introduction

While random forest analyses are a great place to start, if predictive accuracy is our goal, improvements are possible. In some contexts an improvement of a percentage point or two may be meaningless, whereas in other contexts this could be of profound interest. A great deal of attention among data scientists has recently focused on ensemble methods. Ensembles have two contexts – the first refers to models such as random forests where you resample your training set repeatedly and combine the results. I'm referring to ensembles here as combining the results of several distinct models into an overall meta-model. One simple example would be to take the average predicted value from a random forest, a linear regression, and a gradient boosting model. Below we explore a few of these approaches.

Enhanced Models through Ensemble Learning

In the following analysis I perform a weighted voting ensemble on five different classifiers predicting respondent's arrest history. The figure below (utilizing code adapted from Julian, 2016), shows how each of these classifiers performs both on the train and the test set, which is a nice feature to visualize as an indication for which models tend to overfit. As we saw in the previous section, the random forest classifier overfit and achieving a perfect AUC score (1.00). Both the random forest and the extra trees classifier overfit dramatically. I gave each classifier an equal weight, with the exception of the GaussianNB, which was given slightly less.



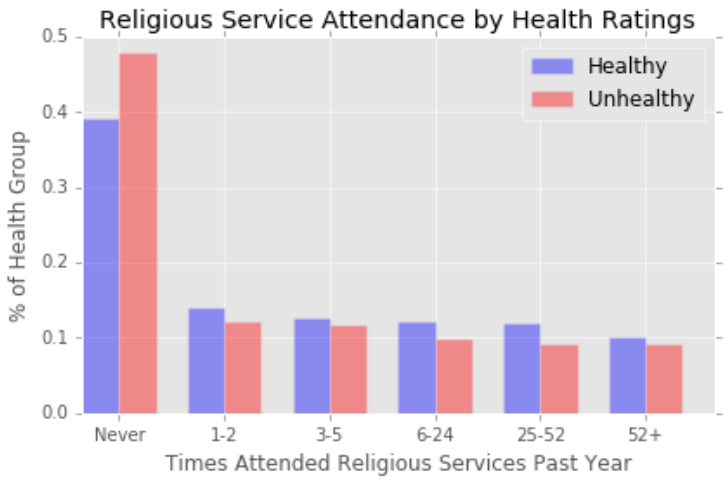
The weighted ensemble model produced an AUC = 0.830, which is an improvement over the benchmark AUC = 0.822. Again, this is not a dance in the streets moment, but it is an improvement that depending on the situation, could prove highly meaningful. We could further work towards improving this score, such as through tuning the parameters of the individual models, feature engineering, adding additional models to the voting classifier, and so on.

For the overall health outcome measure, I applied a "simple" two-fold stacked generalization ensemble model. Variants of such ensemble models are the hottest trend in refined predictive analysis, and invariably the top scorers of Kaggle competitions. For the current analysis, I split the training set in half. I then fit five separate regression models (support vector machine, lasso, random forest, extra

trees, gradient boosting) on each half of the train data and generated predictions on the other half. These predictions were then used as meta-predictors for the final model. I tried both a linear regression and a random forest as meta-regressors. The linear regression meta-regressor generated only a slight reduction of root MSE (.883 compared with the benchmark = .889). The random forest meta-regressor, however, generated a huge reduction (root MSE = .342) on the train data. When I test this model on the hold-out test data, however, it appears that the random forest meta-regressor overfit the data. The root MSE for the random forest rose to .904 while the linear regression meta-regressor produced a root MSE of .882 (which is a minor improvement over the benchmark model of .889).

Looking Deeper – Case Study of Religious Attendance

At this point we could direct our attention to any number of smaller questions. We could look at subgroups of the data, such as only people who have used cocaine and determine what variables predict cocaine use. I found it curious that how frequently people attend religious services was a top predictor of overall health. Of course we cannot assume there is a causal connection between these two variables, nevertheless, their relationship is interesting. To simplify the visual presentation of this relationship, I combined respondents who reported their health as fair or poor into the unhealthy group and those reporting their health as excellent, very good, or good as the healthy participants. There were many more respondents in the healthy group (n = 23,873) than the unhealthy group (n = 3,818), so the following figure is calculated by percentages of both groups rather than counts. As we can see, the unhealthy respondents were less likely to have attended a religious service in the past year – whereas people who self-reported as healthy were more likely to attend a religious services. Future research could explore possible contributing factors to this relationship. Perhaps unhealthy people are less physically able to make it to church (Thoresen & Harris, 2002), or perhaps attending religious services provides social connections, which is also associated with better health (Strawbridge et al., 2001). I'm less inclined to posit divine intervention!



Conclusion

Stacked ensemble methods are useful in boosting the effectiveness of your predictive models. They are, however, generally a small benefit for the increased time and computational resources. There are times when this trade-off is well worth it. You could also spend days attempting to improve upon the model through a variety of strategies. With our fairly simple ensemble methods we were able to

generate a minor improvement in our analysis, which I'll accept at this point. Given that this model is also a combination of several sub-models, I also feel more comfortable that these results will be stable when applied to new data. Finally, we demonstrated the relationship between religious service attendance and overall health self-reports.

Final Remarks

This demonstration is just a tip of the iceberg overview of the data science process. Obviously, we could extract a great deal more out of this dataset and work further to refine the models presented. Nevertheless, I hope it is effective in showing the reader the power of current machine learning practices. Please feel free to contact me if you have any questions, suggestions, or thoughts.

References

Julian, D. (2016). Designing machine learning systems with Python: design efficient machine learning systems that give you more accurate results. Packt Publishing, Ltd: Birmingham, UK.

Morton, K. B., Martin, P. C., Shook-Sa, B. E., Chromy, J. R., & Hirsch, E. L. (2012). 2012 National Survey on Drugs and Health: Sample Design Report.

Strawbridge, W. S., Shema, S. J., Cohen, R. D., Kaplan, G. A. (2001). Religious attendance increases survival by improving and maintaining good health behaviors, mental health, and social relationships. *Annals of Behavioral Medicine, 23* (1), 68-74.

Thoresen, C. E., Harris, A, H. S. (2002). Spirituality and health: What's the evidence and what's needed? *Annals of Behavioral Medicine, 24*(1), 3-13.