CMSC 12300 Final Project

Andy Liu and Nelson Auner

June 3, 2013

1 Introduction

Our project investigates the use of census data to predict whether a household self-reports that its income is over or under $50,00. Our goals, put succintly, were to

1. find and use a relevant prediction algorithm
2. test relevant models and select the best one, and
3. develop an accurate estimation of our model's prediction of a similar, yet different data set.

To do so, we used R and Python, deployed on personal computers as well as AWS, to implement logistical regression of various combinations of regressors. We took the best models, as defined by their performance within the training set, and subjected them to cross-validation. To deal with missing data, we utilized k nearest neighbors to "fill in" missing variables with the nearest closest observation.

Our results suggest the following: Education as a stand-alone predictor is a very good model, with around 5.4% false positive rate and .5% false negative rate. By very good, however, we mean compared to other non-trivial (majority classifier) models, considering that the relative lack of diversity in our responses, given an absolute error weighting, favors a classification scheme with all-negative predictions.

2 Description of the Data set

Our analysis is on the Census-Income (KDD) Data Set, a classic machine learning dataset of census data from the 1994 and 1995 Current Population Surveys conducted by the U.S. Census Bureau.

The actual data is made freely available from the UC-Irvine Machine Learning Repository, and consists of 46 variables (see the appendix for a full list), with features such as race, capital gains, country of birth of parents, and weeks worked in the year. Because the census data if completed by participants on a voluntary basis, many variables had more missing observations than observed observations. This is a challenging statistical problem because a household's decision to respond or not to a given question may be correlated with their household income, our choice variable of prediction, normal regression methods will result in inconsistent estimators. For example, for a standard regression model $Y = X\beta$, we would have:

$$E(y_i|x_i) = x_i\beta + E(y_i|x_j = NA) \tag{1}$$

where x_j represents the variables $j = \{j_1, j_2, ..., j_k\}$ with x_j not reported. Put intuitively, the second term $E(y_i|x_j = NA)$ descibes how we would alter our prediction given that we have NA in columns j; for example, it is possible that wealthier households are more likely to not self-report capital gains.(Another problem all-together is that households could have purposefully mis-reported-a complexity that we avoid all-together to simplify our analysis). The KNN method (described later) seeks to alleviate, but does not solve this problem. Perhaps due to this interaction between household income and missing data, several variables that intuitively would be a very good measure of household income, like wage/hr, weeks worked per year, and capital gains, are not only missing in many observations, but are not good predictors of household income.

One of the most pertinant characteristic of the data set is that the "over 50K" variable only takes on a value of true for 6.2

3 Logistic Regression

The task of predicting whether a household self reports that its income is over or under \$50,000 is a binary classification, as our outcome takes a value of either true (income over \$50K) or false (income under \$50K). We decided early on in the project to use logistic regression, and aim to find the best classification method within the subset of logistic regression classifiers. Logistic regression transforms the predictors to

$$\pi(X) = \frac{e^{(X\beta)}}{1 + e^{(X\beta)}} \tag{2}$$

which looks like:

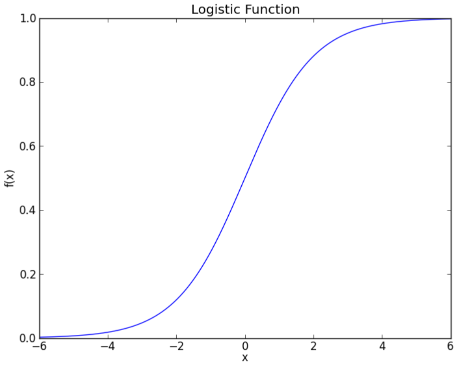


Figure 1: A graph of the logistic function for reference, graphed by the authors using the numpy module of python

This function can be viewed as a CDF (cumulative density function) and closely mirrors that of a normal distribution, and projects the space of $X\beta$ to the interval $[0, 1]$, allowing us to view the result as the probability that our outcome is 1. Our reasons for selecting a logistic regression model are simple: although it is more complicated than a basic regression mode $Y = X\beta$, logistic regression allows us to predict a binary outcome, while resulting in more interpretable coefficients β than "black box" methods such as neural networks or random forest.

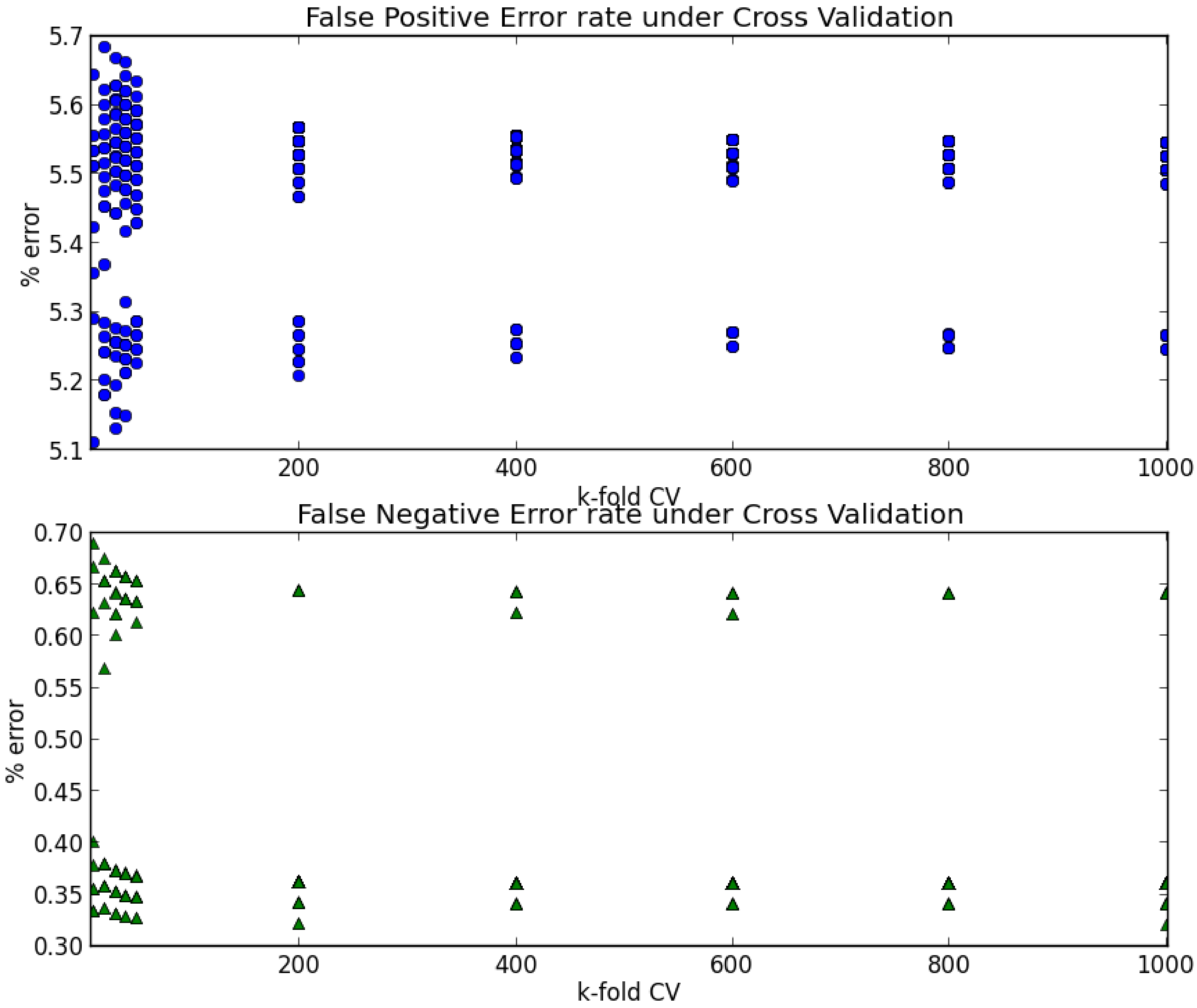
4 Initial analysis

Our inital analysis was done in R, and much work was required to find and adapt the necessary statistical methods for use in python. We used the Akaike Information Criteria (AIC), defined as $AIC = 2k - 2ln(L)$, where k is the number of the parameters in the model, and L is the maximized value of the likelihood function - in our case, the logistical regression equation defined above. By maximizing AIC, we produced a subset of predictors that AIC evaluated as the "best" predictor sets. We noticed that among single-variable predictive methods, the education variable was one of the most effective, with a false positive error rate of 5

5 Filling in missing observations with KNN

Andy spits his magic here.

6 Cross Validation


False Positive Error rate under Cross Validation
% error
k-fold CV
False Negative Error rate under Cross Validation
% error
k-fold CV