Matias Berretta
Vaibhav Dixit
Rohini Mandge
December 8th, 2017

Final Project

CISC 6930: Data Mining
Professor Yijun Zhao

After a preliminary exploration of the census data we found both our training and testing data sets contained missing values and that our training data set was unbalanced. For training and test data alike, all of the missing values were confined to three categorical values: *native_country*, *workclass* and *occupation.* 7.45% of our training data instances, that is 2399 rows, contained missing values, whereas 7.5% of our test data instances, that is 1221 rows, contained missing values. Our training data set was unbalanced with a negative skew, where 77% of the instances were classified as negative and only 23% were classified as positive.

We now present the basic outline of our project: (1) First, we dropped all rows containing missing values, z-score normalized our data, and ignored the unbalanced nature of our data in order to run preliminary tests and set a baseline for Classifier performance; (2) second, we tried out different kinds of imputation—(i) Mode, (ii) Logistic Regression, (iii) Random Forest, (iv) and K-Nearest Neighbors—while still ignoring the imbalanced nature of our data; (3) third, we selected the most successful imputation approach and balanced our data with the bagging classifier method.

For our predication we used an ensemble classifier consisting of 5 algorithms: Random Forest, Logistic Regression, K-Nearest Neighbors, Naïve Bayes, and Support Vector Machines.

Most of the project was developed with Python on Jupyter Notebook, making heavy use of the following libraries, Pandas, NumPy, Sklearn, Matplotlib. We also used R with R-Studio for some of the more complicated imputation methods.

We had three distinct kinds of variables: Continuous, Categorical and Ordinal.
For continuous variables we z-score normalized our values. In particular, we used the training data set's mean and standard deviation to normalized the test data.
For multiclass categorical variables such as race, we spread the original column into binary integer columns, one for each class (i.e. since race contains four classes so we spread the original column into four distinct binary columns, one for each race).

We only had one ordinal value, *education*. We deleted the original *education* variable which was recorded in String format, opting instead to use its neighboring column, *education_lvl,* as the new *education* variable, since the former translated the latter into discrete integers ordered by level of education (i.e. HS-grad translated to a 9, whereas Bachelors translated to a 13).