Predicting Employment through Internet, Social Media, and Gaming Activity: Comparing CART and Random Forests

Abstract

This study compares two different classification methods in predicting employment status through demographic characteristics, social media use, and other factors. We wish to compare the predictive ability of various classification methods, specifically two algorithms of CART and Random Forests, in predicting this binary outcome with survey data using cross validation. Using sensitivity and AUC to assess our classifiers, we found Random Forests to best predict employment status.

Introduction

The 2008 financial crisis was the worst financial crisis since the Great Depression, increasing unemployment by over 5.0% in the United States (BLS, 2012). The United States has since dropped unemployment down to less than 5.0%, unemployment continues to concern our government and its economic progress. Part of the large changes in the job market since then includes further development of technological resources that can be utilized in the job search or employment process. Knowing the employment status within a certain amount of population, governments and organizations are able to get a broad view of the employment market and thus enact policies in a more effective manner. Given the increased use of the internet to advertise jobs and as a medium for job applications (Pew Research Center, 2015; Frost Brown Todd, 2012), we explored how various machine learning algorithms reduced the dimensionality of the data to determine the best survey questions to predict employment status.

Data Acquisition

For this research study, we used the *June 10-July 12, 2015 – Gaming, Jobs and Broadband* data set retrieved from The Pew Research Center's Internet, Science, and Technology project(ref). The dataset contained 2001 observations with 140 columns of survey data. Each column was coded in accordance with the variable guide provided by The Pew Research Center.

We cleaned the dataset by removing conditional questions and recoding missing responses as a new factor. This cleaning process ensured that we obtain maximum relevant information from the data set. Please refer to the appendix for more information regarding the cleaning process. Lastly, we recoded the survey question about employment status as a binary variable indicating whether the subject is employed or unemployed. We removed observations of students and those who were disabled, veterans, or those who refused to answer the question. The final dataset consisted of 1325 observations with 40 columns.

Methodology

Cross-Validation: Assessing Out-of-Sample Performance

Our research focuses on finding the best predictive model, so we utilize the statistical and machine learning practice of cross-validation. We separate our data into a "training" dataset and a "testing" dataset, where the models are constructed on the "training" set and then evaluated on the "testing" set. This tests the model with data it has not seen yet, indicating out-of-sample predictive ability.

Models

1. **CART**

The primary method we will be examining is CART, or classification and regression trees. These 'trees' serve as method of binary or continuous prediction and are in many ways similar to regression methods. Both regression and CART seek to explain some response variable by using explanatory variables. However, they differ in their approach; where regression seeks to find the optimal parameterization of variables, CART seeks to iteratively split the data into different 'bins' based on the variable that results in the optimal split. The CART algorithm recursively looks through all variables to define a 'split', which is a decision based on a variable's value. Each 'split' or 'decision' seeks to divide the data into two groups that are as homogenous in the outcome variable as possible, with each split narrowing this down further. The data is funnelled through these splits until they are dropped into final 'bins' with an estimated value. One issue with these tree methods is overfitting; it is possible to get perfect in-sample prediction if you have enough branches in a tree. As a result, a tree needs to pruned, which means that splits with smaller contributions are removed from the tree in order to improve out of sample predictive ability. While there are numerous algorithms for CART, we used the *rpart* and *tree* algorithms in our comparison.

2. **Random forests**

Random forests is an extension of CART developed to address overfitting issues. This is done through taking many random samples of the variables in the training set and creating a multitude of trees. Predictions are made based on all of these trees, and the mode of these predictions is presented as the estimate. Normally, if the response variable is strongly correlated with predictors, they will reliably be selected as splits on multiple trees constructed from the data. Through the random forests method, the features to be split on are selected from a random subsample of all the features, removing spurious correlation from at least some of the trees. Using the mode of the predictions given by this forest of classification trees gives us a model with less overfitting issues than an ordinary classification tree.

Assessing our Classifiers

We will use a few metrics to assess the out-of-sample predictive performance of our models. We used sensitivity and AUC, a metric relating true positive and false positive rates. Sensitivity is the proportion of subjects whose employment status was correctly identified. *A priori*, we decide on a 0.5 cutoff that is needed for accuracy; if the predicted probability of being employed is above 0.5, we treat it as a positive. Conversely, observations with predicted probabilities below 0.5 are treated as negatives.

We are also interested in the relationship between true and false positive rates. The receiver-operating characteristics (ROC) curve plots this relationship, with the false positive rate on the x-axis and the true positive rate on the y-axis. The area under the curve, or AUC, is a commonly used method of evaluating a classifier. The closer an AUC is to 1, the better the classifier. If the AUC is 0.5 or less, random guessing performs just as well as the model in question. As the ROC curve is drawn along different cutoff values, a threshold does not need to be selected to determine the AUC of a model. AUC also has other desirable qualities compared to accuracy, and is generally a better measure than accuracy for evaluating classifiers (Huang et al, 2003).

Results

1a. *CART (rpart)*

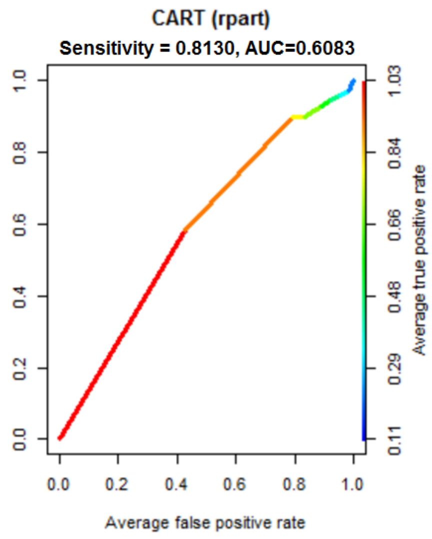
The *rpart* algorithm created a full model with 17 splits. However, given the aforementioned issue with trees and overfitting, we selected a complexity parameter of 0.02 to prune the tree to 8 splits. The resulting tree is in the appendix.

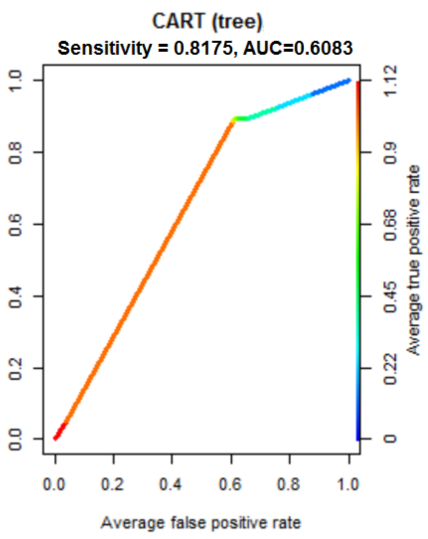
1b. *CART (tree)*

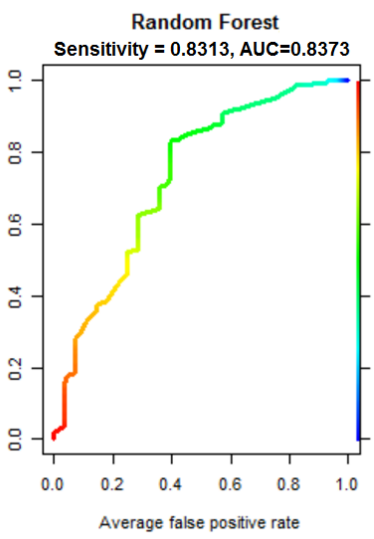
The *tree* algorithm created a full model with 13 splits. This algorithm used deviance to prune the full model. Using the smallest deviance of 540.34, the pruned model had 9 splits. The resulting dendrogram is in the appendix.

2. *Random Forests*

Due to the ensemble nature of random forests, we do not have a dendrogram for random forests because it constructs many different trees. However, we can see which variables were important; the beside figure shows the variables that were consistently selected in trees as strong predictors. Like in the individual classification tree, we see some of the same variables selected; education, age, marital status, and race are all viable split variables to describe this data.







All three models performed fairly similarly in terms of sensitivity; all were around 81-83% accurate for the cutoff of 0.5. On the other hand, all three models had very high False Positive Rates at the same classification threshold of 0.5. After looking at the data, this makes sense; a majority of the people in our dataset were employed, so all the models overestimate the probability of being employed to some degree. Very few predicted probabilities fell below 0.5 overall, so a majority of true negatives were also classed as positives at this cutoff. As the cutoff increases, the false positive rate decreases. This can be seen in the above graphs by the color of the curve; the cutoff value moves from blue to red as the ROC curve is drawn. Since the AUC is drawn from plotting the relationship between true and false positives for different classification cutoffs, AUC is a better measure for overall model performance.

Conclusion

Our analysis shows that Random Forests is the best performing classifier for our dataset in terms of both accuracy and AUC. Random Forests performed marginally more accurately at our heuristically-defined cutoff, but had significantly better model performance overall as measured by AUC. For both CART models, differences were negligible; although we used different methods to prune the models, we ended up with practically identical classifiers in terms of performance.

All three models reduced the dimensionality of the dataset into different variables, but the most prominent variables present in all the models were age and education level. There were 17 variables in all three models combined. See appendix for the important variables. 9 among the 17 were related to technology indicating more evidence that the use of technology is potentially having a growing influence on employment status, but more traditional demographic variables are still significant.

While still not perfect, ensemble methods such as Random Forests can yield fairly reliable predictions, even with this very noisy, highly categorical data. This is because Random Forests uses a modified sampling technique that removes correlations across samples/tree. As a result, using Random Forests to control for overfitting instead of using complexity parameters to prune CART algorithms is comparable to using a nonparametric test, which uses sampling methods, instead of a parametric test, which is contingent on the choice of a theoretical distribution. A possible direction for future research is to stack multiple algorithms that account for overfitting for optimal out-of-sample prediction.

References

Huang, J., Lu J., and Ling C. X., "Comparing naive Bayes, decision trees, and SVM with AUC and accuracy," *Third IEEE International Conference on Data Mining*, 2003, pp. 553-556. Retrieved from http://ieeexplore.ieee.org/document/1250975/

Kelly, J. (July, 2012). All the World's a Stage: The Internet and Employment Law. *Frost Brown Todd Attorneys, LLC.* Retrieved from http://www.frostbrowntodd.com/resources-1433.html.

Smith, A. (November 19, 2015). The internet and job seeking. *Pew Research Center: Internet, Science & Tech*. Retrieved from http://www.pewinternet.org/2015/11/19/1-the-internet-and-job-seeking/.

The Recession of 2007-2009 (February, 2012). *BLS Spotlight on Statistics*. Retrieved from http://www.bls.gov/spotlight/2012/recession/pdf/recession_bls_spotlight.pdf.

Appendix

Data Cleaning
In order to use columns where every individual is given a chance to respond to the survey question, we removed all conditional questions. A conditional question is a question that is only asked if the subject responds to a previous question with a fixed response. We also removed variables that we presumed would be correlated to or associated with employment status. We removed the following variables:

Intfreq, HOME4NW, BBHOME1, BBHOME2, BBHOME3, DEVICE1a, SMART1, Q4, BBSMART1, BBSMART2, BBSMART3, BBSMART4, CABLE2, CABLE3, WEB1-A (randomized), Q5, DATE1a, DATE2a, GAME4, SMJOB1, SMJOB2, SMJOB3, SMJOB4, SNSJOB1, SNSJOB2, KidAge1, KidAge2, KidAge3, EDInst, PARTYLN, BIRTH_HISP, QL1a, QC1, Int_date, usr, form, MONEY5A, MONEY5B, TEEN1, TEEN2, EMAIL, CNFRM, RZIPCODE, EMPTYPE1, EMPTYPE2, EMPTYPE3, STUD, JOB1, JOB2, JOB3, JOB4, JOB5, AUTO1, AUTO2, weight, standwt, psraid, disa, inc, cregion, state

Next, we changed all “Don’t Know” and “Refused” responses to a factor called ‘99’. This ensured that if the refusal to answer a question had significance in the dataset, we could find its significance. We used the following code in R to add the factor ‘99’:

```
survey <- read.csv("data.csv",header=T)
stdvars<-colnames(survey)[c(-1,-4,-5,-34,-35,-39,-41)]
for (f in c(stdvars)) {
  for (g in (1:(dim(survey)[1]))) {
            if (survey[g,f]==8) {
            survey[g,f]<-NA
            } else if (survey[g,f]==9) {
            survey[g,f]<-NA
            }
  }
}
survey$educ2[survey$educ2==98|survey$educ2==99] <- NA
survey$party[survey$party==4|survey$party==5|survey$party==8|survey$party==9] <- NA
survey$inc[survey$inc==98|survey$inc==99] <- NA

newsurvey<-survey
for (f in 1:dim(newsurvey)[2]) {
  levels(newsurvey[,f])<-c(levels(newsurvey[,f]),99)
  newsurvey[,f][is.na(newsurvey[,f])]=99
}
colnam<-colnames(newsurvey)[c(-(match("age",colnames(newsurvey))))]
for (f in colnam) {
  newsurvey[,f]<-factor(newsurvey[,f])
}
```

CART Algorithms:

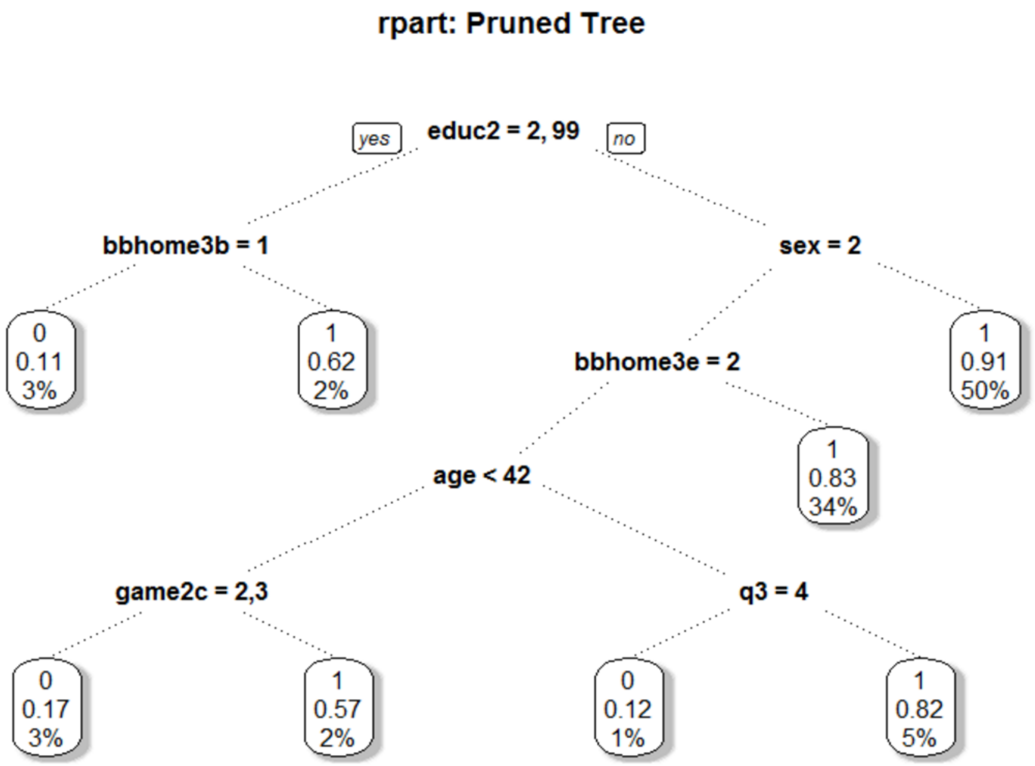
1) *rpart*

We used the following code:

```
tree1 <- rpart(employed ~ ., data = train,
               method = "class")
tree1$cptable #looking at best cp to prune tree by
plotcp(tree1)
tree2 <- rpart(employed ~ ., data = train,
               method = "class",
               cp = 0.0273) #plugging in cp
prp(tree2)
```

Note that each node has three values; the predicted class (top value), the proportion of observations correctly classified at the node (the second value) and the percentage of observations in that node (the third value).

Pruned model:

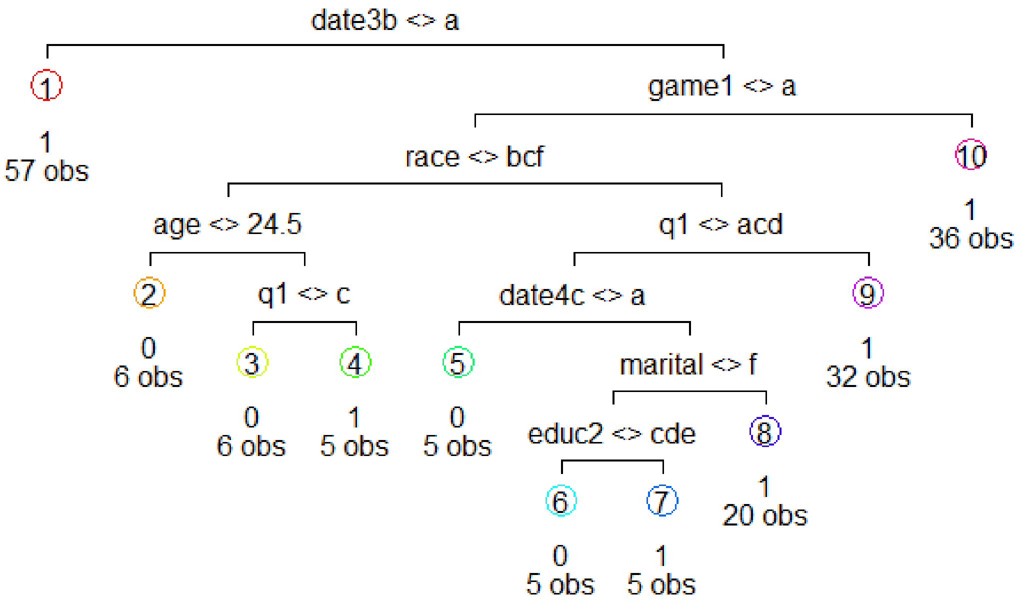


2) tree

We used the following code:

```
tree <-tree(employed~sample+lang+sex+q1+q3+eminuse+intmob+bbhome3a+
          bbhome3b+bbhome3c+bbhome3d+bbhome3e+cable1+date3a+date3b+
          date4a+date4b+date4c+date4d+date4e+date4f+game1+game2a+
          game2b+game2c+game2d+game2e+game2f+game3a+game3b+age+
          marital+hh1+par+educ2+party+ideo+hisp+race,data=train)
tree.cv <- cv.tree(tree); tree.cv
plot(tree.cv)
tree.cv$size[which(tree.cv$dev==min(tree.cv$dev))] #which minimizes deviance
tree.prune <- prune(tree, best=2, method = "misclass")
draw.tree(tree.prune)
```

Pruned model:



Random Forests:

We used the following code:

```
randfor<-randomForest(employed~.,data=train,
importance=T, proximity=T, method='class', ntree=500)

varImpPlot(randfor) #gives us variable importance plot
```

Final Model: Due to the nature of the ensemble algorithm, we cannot draw a nice dendrogram of the whole forest. However, we can show which variables were found to be important in the hundreds of trees created by the algorithm.

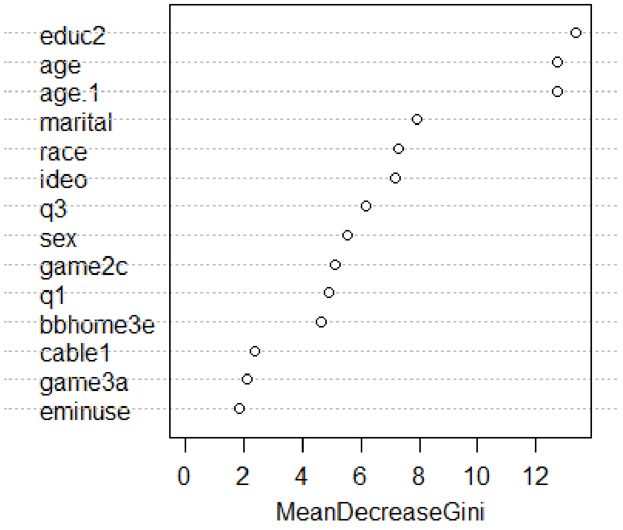


Table 1: List of all variables selected in any of the three models.

Variable Name	**Question**
educ2	Highest level of school/degree finished?
age	What is your age?
marital	What is your marital status?
race	What is your race?
ideo	Where on the political spectrum do you fall?
q3	How would you rate the economy today?
sex	What is your sex?
game2c	Do video games portray women poorly?
q1	How would you rate your community?
bbhome3e	Are those without internet not current on news?
game1	Do you ever place video games?
cable1	Do you receive television?
game3a	Are most people who play games men?
eminuse	Do you use the internet occasionally?
date3b	Do you know couples in a long term relationship who met through online dating?
date4c	Are people who use online dating desperate?
bbhome3b	Are those without internet not current on governmental services?