Anisha Wadhwa
Springboard: Introduction to Data Science
Capstone Project, March 2016
Mentor: Joel Bangalan

Predicting the Presence of Internet Ads on Websites

ABSTRACT

Internet Ads serve as a revenue-generating platform for online media companies while allowing third party sponsors to target potential customers during the consumer's web-browsing activities. However, this severely disrupts the user-experience and hence companies and even individuals are willing to invest in technology that blocks these advertisements.

This paper attempts to develop a classification model that can predict the presence of an advertisement on a website. By using the LASSO methodology to select and regularize the relevant variables, a model was developed with 95.91% accuracy and 0.48% false positive rate on the testing dataset.

I BACKGROUND

The Rising Prevalence of Internet Ads

In 2015, the average user spent 1.72 hours on social networks per day, up from 1.66 hours in 2013. The average time spent on the Internet in general, far surpassed these numbers, at 6.1 and 5.5 hours in 2015 and 2013[1] respectively. Correspondingly, younger populations are spending more time on the web than watching real-time television, creating a large opportunity for online media companies to use the Internet as a revenue platform from third-party advertisements.

By using different websites to shop, date, make travel plans or simply browsing pages for pleasures, users are allowing their personal preferences to become available to third parties. Web Browsers like Google Chrome and Search Engines like Google, Bing and Yahoo, collect enormous amounts of data about each user and leverage this information to drive their main source of revenue from Internet ads.

Internet Ads are banners that pop up while on web pages like social media and search engines that link out to the third-party's (sponsor's) website. With targeted consumer marketing becoming necessary to for companies to remain competitive, these Ads are becoming more and more relevant to the user's interests, tastes and preferences based on the data received from user's browsing history.

What is the Problem?

While Advertisements have always been integral to consumer's day to day life, Internet ads have been threatening individual privacy as well as degrading user's browsing experience.

Internet Ads make the browsing process unpleasant to users who have no desire to link out to these third party websites.

Ads also significantly increase download time, adding to the user's usage of data and time spent on the online task.

Lastly, and more specific to businesses and enterprises, these advertisements can serve as a distraction to employees, significantly reducing worker productivity.

Learning to Detect and Remove Ads: Who Benefits?

Companies are willing to invest in technology and software that will serve as a browsing assistant to eliminate Ads from the user experience, especially from the company email server. While software like AdBlock Plus cater to this purpose, this comes at a licensing price to large enterprise customers. My hypothetical client is an information services company of 10,000 employees that is testing the feasibility of building an ad-detection technology in-house for both web pages as well as private email servers.

Building the technology within the company will serve three short and long-term benefits:

[1] "28% of Time Spent Online Is Social Networking." SocialTimes. Accessed March 12, 2016. http://www.adweek.com/socialtimes/time-spent-online/613474.

- It will avoid paying yearly licensing fees to software companies like AdBlock Plus. According to AdBlock Plus' website, the total revenue from licensing payments from large enterprises represents 30 percent of the additional revenue created by whitelisting its acceptable ads.[2]
- An in-house software or technology will also allow the company to customize its detection system. It will be able to build a technology based on **features** of the Ad "HTML" rather than on handcrafted filters. As the Internet grows, filtering will be limited in learning and detecting different types of advertisements. Therefore, this one time investment in a feature-recognition technology and software can have long-term benefits for the firm.
- A technology that removes Ads before the page loads will significantly increase the browsing speed while decreasing the required Internet bandwidth for each of the firm's offices, significantly reducing the dollar amount spent on internet speed.

II THE DATASET: Description, Wrangling and Limitations

The Dataset used in this study is the "Internet Advertisements Dataset" from the Machine Learning Repository, University of California Irvine, donated in 1998 to the archive.

https://archive.ics.uci.edu/ml/datasets/Internet+Advertisements

Description

This dataset represents a list of possible advertisements on Internet websites. An *Internet Ad* in this dataset and analysis is defined as an image that appears as a banner on the side panel of the webpage.

The original dataset consists of a list of 3279 observations of both ads and non-ads, with 1559 variables.
The variables are features of the HTML link, for example, whether the HTML contains certain words, links out to another site or the presence of certain words in the anchor text.
The categories of the *independent variables* can be found in the table below. Only 3 of the 1559 variables are numeric, the remaining being factor with values 0 or 1 representing not present or present.
The *dependent variable* is the 1559th variable, **a binary/boolean variable** of whether the observation is an "ad" or "nonad".

[2] "About Adblock Plus." About Adblock Plus. Accessed March 12, 2016. https://adblockplus.org/en/about.

Table 1: Independent Variable Categories

Variable Category	**Added Description**	**Class**	**# of Variables**
Dimensions/Geometry of the Image	• Height • Width • 'aratio' (ratio of height to width)	Numeric	3
Local	Common domain of destination and image servers	Factor: 0, 1	1
URL Features		Factor: 0, 1	495
Image URL		Factor: 0, 1	457
Alt Text		Factor: 0, 1	
Anchor Text		Factor: 0, 1	
Words Below Anchor Text		Factor: 0, 1	

Data Wrangling

The data in this dataset needed to be cleaned in two manners:

a) Variable Class Conversions: 1556 of 1559 variables are Boolean variables with values 0 or 1 (FALSE or TRUE) and hence had to be converted to the 'factor' vector from the 'integer' vector. The reverse is true for the other three variables that represent dimensions of the advertisement.

b) Missing Attributes: 28% attributes in the original dataset were missing. They were subsequently removed by first taking out 3 variables that had more than 5% of attributes missing (~ 163 observations). Then, 15 observations were removed on the basis of the existence of NA value/values.

After Data Wrangling was completed, the modified and relevant dataset was reduced to ***3264 observations of 1556 variables (1555 independent and 1 dependent).***

Data Limitations

From an Analysis Point of View, this data set had two major complexities that made extraction of useful insights challenging. a) Large Number of Abstract Variables b) Binomial Variables

After wrangling and cleaning, there are 1555 independent variables in this dataset, 1552 of which representing a binary feature linked to the associated web page's URL/HTML. Therefore, it was difficult and time-consuming to understand the relevance and meaning of variables like "url*shareware.com" and "alt*more".

In addition to the lack of knowledge about the meaning of the variables, the large number of binomial variables also made it hard to determine which independent variables had the highest impact on the dependent variable, thereby lengthening the variable selection process. The initial strategy was to produce a correlation matrix to select the most important variables for the model, however, that method was quickly discarded on learning the class of majority of these variables.

III THE METHODOLOGY[3]

Taking into consideration the large number of binary variables and the need to significantly reduce the number of variables in the model, the best method to obtain the most accurate model was through ***LASSO: the Least Absolute Shrinkage and Selection Operator.*** This is a machine learning algorithm that performs both regularization and variable selection at the same time therefore making it a stronger alternative to stepwise selection (only conducts selection on covariate variables) and ridge regression (only predicts the relationship after variables have been selected). Unlike ridge regression, LASSO also allows the reduction of variables to 0 coefficients.

The LASSO adds a penalty for every variable added to the model and this total penalty is assigned to a term ***lambda***. Therefore, the goal was to determine the minimum penalty value for the model that will in turn result in appropriate variable selection.

By using the 'glmnet' library in R, it was possible to automatically conduct cross validation and determine the least lambda value.

The LASSO algorithm and corresponding use of the 'glmnet' function returned the coefficients of all of the 1556 variables. The result returned ***only 57 variables with non-zero coefficients***, these representing the relevant variables for this model. A list of these features and their corresponding coefficients can be found in the appendix.

[3] "Introduction to Statistical Learning." Introduction to Statistical Learning. Accessed March 12, 2016. http://www-bcf.usc.edu/~gareth/ISL/data.html.

IV ANALYSIS

The LASSO method allowed for 1555 independent variables to be reduced to 57 relevant independent variables.
After splitting the dataset into a testing (30% = 979 observations) and training set (70% = 2285 observations), it was possible to check the accuracy of the model by two corresponding confusion matrices.

Goal of the Predictive Model

Before getting into specific numbers on model accuracy, it is important to highlight the goal:
Given the application of this model, it is safer to have ad-blocking technology that allows certain ads to slip through the cracks and be present than to have it block actual content. Therefore, the goal of the model is to have a low false positive rate: the model should show low instances of detecting and blocking images/content that are not an ad.

Table 2: Confusion Matrix

```
> table(y, train.lasso)
        train.lasso
y         ad. nonad.
  ad.     248     62
  nonad.    3   1972
> table(y.test,lasso.pred)
        lasso.pred
y.test   ad. nonad.
  ad.    108     36
  nonad.   4    831
```

Testing/Predictive Accuracy

Looking at the second matrix that provides the results of the predictive capabilities of the model on the testing dataset, several conclusions can be drawn. First, the model or classifier is correct (***accurate)*** 95.91% of the time. Therefore, the ***misclassification rate*** is only 4.09%.
The ***sensitivity/precision*** of this model, or the ***true positive*** of the model (when there is an ad and the model accurately predicts it) is 75%. The ***specificity*** (correctly predicts nonad) is 95.81%.
Furthermore, the ***false positive rate*** for ads (predicting an ad where there is not one) is extremely low at 0.48%. This proves that our model has been very successful in predicting the presence of ads while not blocking significant real content on this testing dataset.

False Positive VS False Negative

We do want to ensure that there is a ***balance between the false positive and false negative rate***, and that the model is not being too lenient. The false negative is when the model does not detect an ad when there is one and this is 25%, that is, significantly high.

Therefore, this model is detecting relatively fewer ads at the risk of blocking user's required/desired content.
Since the goal is to have the lowest possible false positive rate, we can pardon the relatively higher false negative rate.
However, in other use cases where high false negative rates can have a significant detrimental impact to the model and its predictive ability, it would be necessary to find the threshold point between the false positive and false negative through an ROC curve.

V SCOPE & LIMITATIONS

Since the focus of this model was to use the features of the HTML link to determine if an image on a webpage was an ad or nonad, the model has proven a high level of accuracy in the testing and training dataset as well as a very low false positive rate.
However, we do face two main constraints when applying this model to other datasets.
The first is that the observations itself was outdated. The observations and corresponding features used in this model were from 1998. The coding behind webpages, type of ads and technology behind these web based advertisements have evolved in the last ~20 years. Therefore, we face limitations as to the relevancy of this model to ads that dominate our user experience today.
Ads are now not necessarily restricted to the columns outside of the main text. Therefore, this model may not be as effective in detection different types of ads on websites such as social media where the ads are not in the form of banners but are embedded within the main content of the website.

VI FUTURE WORK

When looking at this dataset itself, on future research, it would be beneficial to have a larger number of observations. An added plus would be is if the observations are not only from public web domains but also from private email servers (eg.gmail). This would help diversify the types of ads we are trying to detect and will increase the applicability and reliability of the model.
Given the advancement of technology behind website domains and advertisements themselves, the training and testing dataset need to evolve with the new types of advertisements, thereby capturing all possible features that can predict the existence of an advertisement.

VI REFERENCES

"About Adblock Plus." About Adblock Plus. Accessed March 12, 2016. https://adblockplus.org/en/about.

"Introduction to Statistical Learning." Introduction to Statistical Learning. Accessed March 12, 2016. http://www-bcf.usc.edu/~gareth/ISL/data.html.

"28% of Time Spent Online Is Social Networking." SocialTimes. Accessed March 12, 2016. http://www.adweek.com/socialtimes/time-spent-online/613474.

VII APPENDIX

A] CODE

```
####Importing Data set from text file:
read.csv("~/Practice Internet Ad/Ad Practice/ad-dataset (2)/ad.data", header=FALSE)

###Summary Stats
summary(ad)

### Get the shape of the data
## Get the "shape" of the data
dim(ad)
head(ad)
tail(ad)
str(ad)

###Variable labels (df dataset)
install.packages("readxl")
library(readxl)
df <- read_excel("Variable name .xlsx")

##Transposing from column to row
View(t(df))

var.names <- t(df)

##Attaching Varible Labels to the dataset
colnames(ad) <- var.names


### Converting variables into relevant vectors:
## Variable 1:4 from factor to numeric
install.packages("plyr")
install.packages("dplyr")
library(plyr)
library(dplyr)

#(Delete this possibly)
indx <- sapply(ad, is.factor)
```

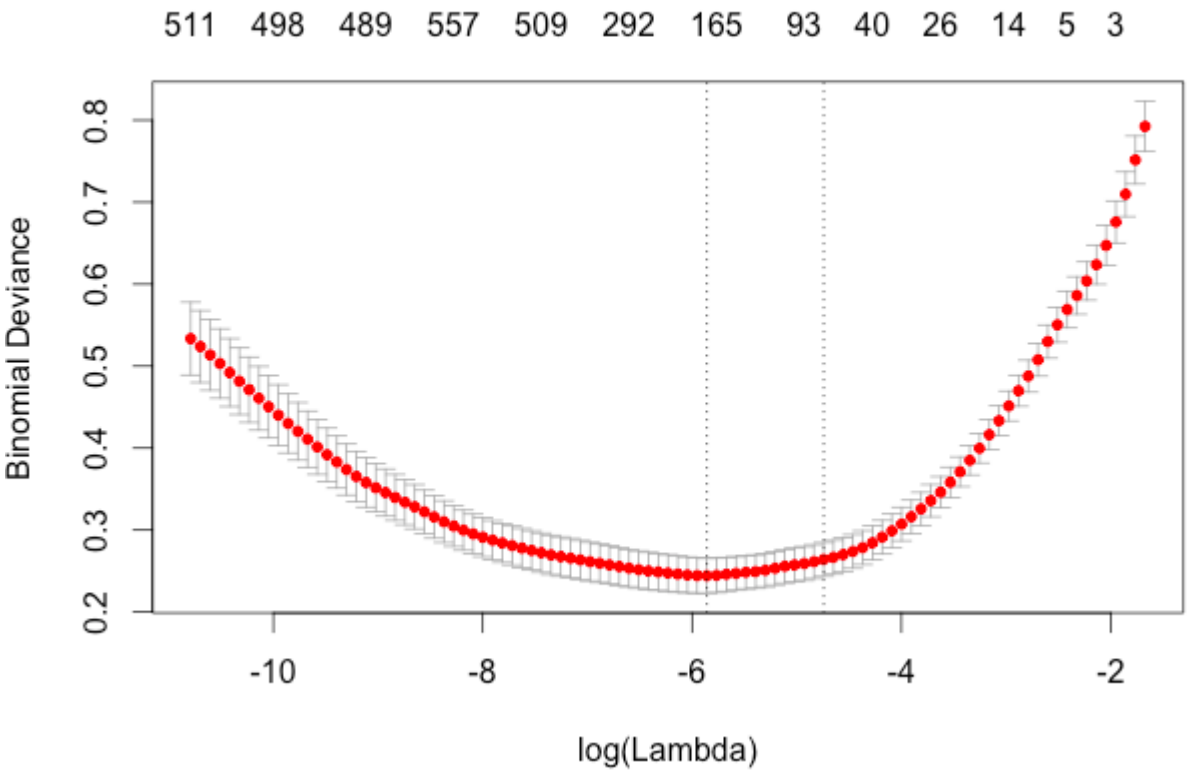
```
ad[indx] <- lapply(ad[indx], function(x) as.numeric(as.character(x)))

##Subsetting the dataset to keep out the dependent variables from all the conversions
newdata <- ad[c(-1559)]
indx <- sapply(newdata, is.factor)
newdata[indx] <- lapply(newdata[indx], function(x) as.numeric(as.character(x)))

##Adding the dependent variable back in
newdata$ad_detected <- ad$`ad/nonad`

##Converting integers to factors
## variable 4:1558 from integer to factor

indx2 <- sapply(newdata, is.integer)
ad[indx2] <- lapply(newdata[indx2], as.factor)


## Checking str(ad)
str(newdata)

## Convert "local" variable from numeric to facto
as.factor(newdata$`local `)
newdata$`local `<- as.factor(newdata$`local `)


##Find a threshold value for when to not use a specific column?
### remove any columns that may have more than 5% values missing

final_data <- newdata[,colSums(is.na(newdata)) < 163]

###check how many NA are left
sum(is.na(final_data))

### Removing rows with NA data --
ad_data2 <- final_data[rowSums(is.na(final_data)) < 1,]

## "ad_data2" is the dataset used in the model after wrangling is completed and before LASSO

## Logistic (LASSO)
### Creating the test and train samples
test <- sample(nrow(ad_data2),0.3*nrow(ad_data2))
data.train <- ad_data2[-test,]
```

```
data.test <- ad_data2[test,]


x <- model.matrix(ad_detected~.,data=data.train)[,-1]
y <- data.train$ad_detected
x.test <- model.matrix(ad_detected~., data=data.test)[,-1]
y.test <- data.test$ad_detected

install.packages("glmnet")

##cross validation for lambda
library(glmnet)
grid = 10^seq(10,-2,length=100)
lasso.train <- glmnet(x,y,family="binomial", alpha=1, lambda=grid)
dim(coef(lasso.train))

##determining minimum lambda value
set.seed(123)
cv.out = cv.glmnet(x,y,alpha=1,family="binomial")
plot(cv.out)
bestlam = cv.out$lambda.min
#bestlam = cv.out$lambda.1se



### Training Accuracy
train.lasso = predict(lasso.train, s=bestlam, newx=x, type="class")
table(y, train.lasso)

### Test Accuracy
lasso.pred = predict(lasso.train, s=bestlam, newx=x.test, type="class")
table(y.test,lasso.pred)
## 4 is the false positive -- which we want to control. we dont want them to be taking ads out.

###performing regularization to get coefficients
out = glmnet(x,y,alpha=1,lambda=grid,family="binomial")
lasso.coef = predict(lasso.train,type="coefficients",s=bestlam)[1:1556,] ####THE 1556 represents
the number of variables.
lasso.coef
lasso.coef[lasso.coef!=0] ## This should give you our model itself. Which variables are relevant.
The ones with 0 as coefficients will be removed
```

length(lasso.coef[lasso.coef!=0])

###Remaining Variables? 57 -- down from 1556

B] LAMBDA: Finding the Best Minimum Value



C] VARIABLE LIST AFTER LASSO

(Intercept)	`url*keith+dumble `
3.392362e+00	-3.336851e-01
`url*site `	`url*banners `
-1.369848e+00	2.257758e+00
`url*keith `	`url*members+keith `
-8.803517e-03	-4.013447e-13
`url*ads+media `	`url*memberbanners `
-1.759170e+00	-4.664326e+00
`url*members `	`url*images+home `
-1.917020e-02	-2.627377e+00
`url*memberbanners+live `	`url*bin `
-1.874952e-02	-1.665390e+00
`url*ads `	`url*banner+gif `
-2.830874e+00	-1.106356e+00
`url*live `	`url*uk `
-1.275540e-01	-2.342819e+00
`url*ad `	`url*dumble `
-2.841694e+00	-4.059982e-02
`url*ukonline.co.uk `	`url*logo+gif `
-6.339843e-13	-6.958842e-01
`url*web.ukonline.co.uk `	`origurl*www.thriveonline.com `
-1.294908e-03	-1.242933e+00
`origurl*web.ukonline.co.uk `	`origurl*jun `
-5.768228e-03	-1.947823e+00
`origurl*ukonline.co.uk `	`origurl*thriveonline.com `
-2.577674e-03	-4.433965e-04
`origurl*zdnet.com `	`origurl*bin `
-5.129774e-01	-2.195228e-01
`origurl*dumble `	`origurl*netcenter `
-7.741775e-03	-2.305644e+00
`ancurl*redirect `	`ancurl*nph `
-1.295927e+00	-2.347985e+00
`ancurl*adclick `	`ancurl*n+a `
-1.873066e+00	-1.019329e+00
`ancurl*site `	`ancurl*redirect+cgi `
-4.651971e-01	-1.377642e+00
`ancurl*main `	`ancurl*click+ng `
-1.618021e-01	-1.254123e+00
`ancurl*download `	`ancurl*http+www `
-1.618025e-01	-1.941972e+00
`ancurl*com `	`ancurl*cid `
-1.327236e+00	-1.440290e+00

`ancurl*thejeep.com `	`ancurl*url `
-9.426310e-01	-1.705360e+00
`ancurl*pl `	`ancurl*co `
-6.962300e-02	-4.653029e-02
`ancurl*marketing `	`ancurl*click `
-2.482932e+00	-2.583500e+00
`ancurl*www.thejeep.com `	`alt*your `
-2.309684e-02	-3.007084e-01
`alt*for+a `	`alt*visit+our `
-7.408751e-01	-3.265241e+00
`alt*now `	`alt*click+here `
-1.223863e+00	-1.066330e-01
`alt*click `	`alt*banner `
-1.398854e+00	-2.052604e+00
`alt*for `	
-3.479633e-01	