IE308HW3

Aneesh Kudaravalli

3/2/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

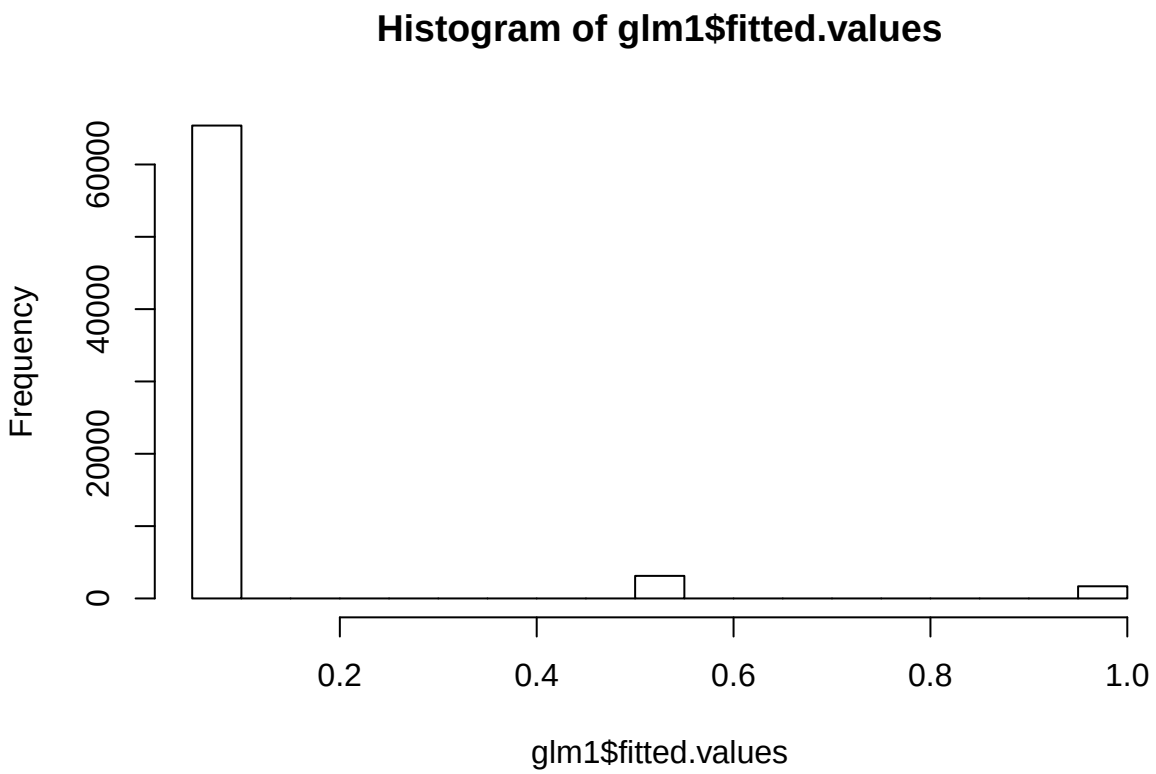
```
####running logistic regression on entities and features to further
### narrow down names to CEOs, and to narrow down organizations to companies


####names training set (exported from python) (all 2013 data)

data = read.csv("NamesTrainingSet.csv", header = TRUE)

data[is.na(data)] <- 0
###fitting a logistic regression model
glm1= glm(in_train ~ as.logical(CEOinSent) + as.logical(pop_CEO), family = binomial(link="logit"), data

# the model
glm1
```

```
##
## Call:  glm(formula = in_train ~ as.logical(CEOinSent) + as.logical(pop_CEO),
##     family = binomial(link = "logit"), data = data)
##
## Coefficients:
##               (Intercept)  as.logical(CEOinSent)TRUE
##                    -2.328                      2.494
##   as.logical(pop_CEO)TRUE
##                     6.535
##
## Degrees of Freedom: 70187 Total (i.e. Null);  70185 Residual
## Null Deviance:       54380
## Residual Deviance: 43710     AIC: 43720
```

```
summary(glm1$fitted.values)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.08883 0.08883 0.08883 0.13052 0.08883 0.99877
```

```r
hist(glm1$fitted.values) ### set .4 as boundary
```




```r
#### (Greater than .4 means they are classified as CEO)


##misclass rate
ceotrainingmisclass = numeric(70188)

for (i in 1:70188)
{
if (glm1$fitted.values[i] >= .4 && data$in_train[i] == 'False')
{
ceotrainingmisclass[i] = 1
}
}

####  training misclass rate on ceos
mean(ceotrainingmisclass)
```

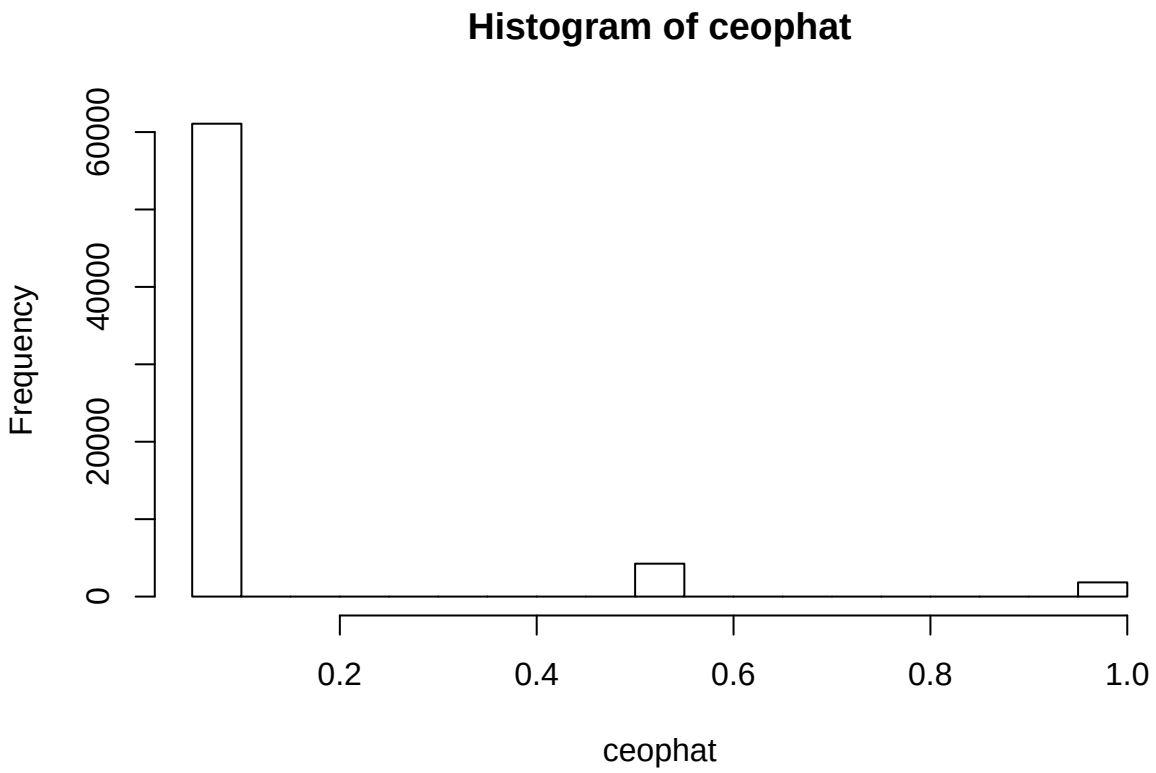
```
## [1] 0.02057332
```

```r
#### testdata and features (all 2014 text data exported from Python)
testdata = read.csv("NamesTestSet.csv", header = TRUE)
newdat = data.frame(as.logical(testdata$CEOinSent), as.logical(testdata$pop_CEO))
names(newdat)[1] <- "CEOinSent"
names(newdat)[2] <- "pop_CEO"
newdat[is.na(newdat)] <- 0


##predicted values on test data
```

```
ceophat = predict(glm1, newdata = newdat, type = "response")
summary(ceophat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.08883 0.08883 0.08883 0.14206 0.08883 0.99877
```

```
hist(ceophat)
```




```
### test data misclass rate
ceotestmisclass = numeric(67168)

for (i in 1:67168)
{
if (ceophat[i] >= .4 && testdata$in_train[i] == 'False')
{
ceotestmisclass[i] = 1
}
}

#average misclass rate on CEOs for the test set from 2014
mean(ceotestmisclass)
```

```
## [1] 0.0363566
```

```
### final list of CEOs are names that the logistic regression model classified as CEOs
### for both the training 2013 set and the test 2014 set with duplicates removed

finalceos = numeric(137356)


for (i in 1:70188) ###training set
```

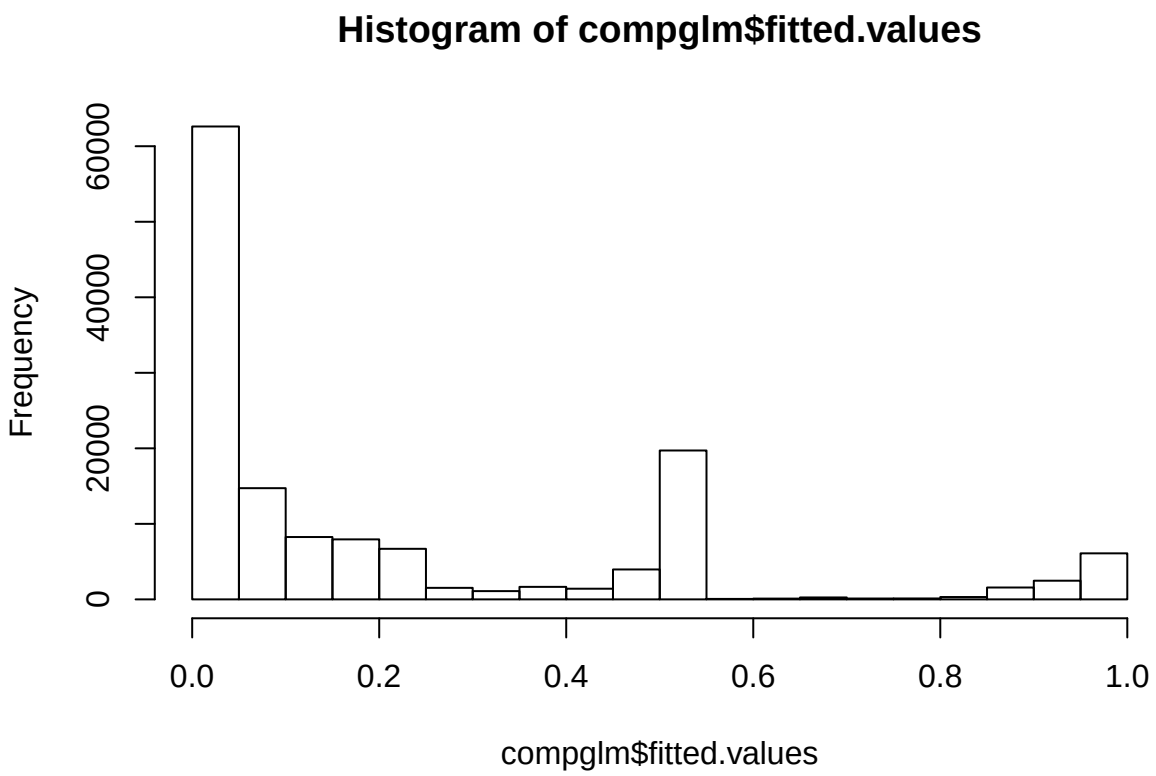
```
{
if (glm1$fitted.values[i] >= .4)
{
finalceos[i] = as.character(data$NAME[i])
}
else
{
  finalceos[i] = 0
}
}

for (i in 1:67168) ###test set
{
if (ceophat[i] >= .4)
{
finalceos[i + 70188] = as.character(testdata$NAME[i])
}
else
{
  finalceos[i + 70188] = 0
}
}


finalceos[finalceos == 0] = NA
finalceos = na.omit(finalceos)

finalceos = unique(finalceos) ###remove duplicates

write.csv(finalceos, "finalfinalceolist.csv")


### orgs/companies training data from 2013 exported from excel
comptraindata = read.csv("CompaniesTrainingSet.csv", header = TRUE)


##fitting a logistic regression model
compglm = glm(in_train ~ stock + price + founder + part + comp + comps + numWords + forbes, family = bi
compglm
```

```
##
## Call:  glm(formula = in_train ~ stock + price + founder + part + comp +
##     comps + numWords + forbes, family = binomial(link = "logit"),
##     data = comptraindata)
##
## Coefficients:
## (Intercept)    stockTrue    priceTrue  founderTrue     partTrue
##     -3.5365       0.4390       0.2764       1.3045       0.2281
##    compTrue    compsTrue     numWords   forbesTrue
##      0.9113       0.5787      -0.1580       3.6834
##
## Degrees of Freedom: 140697 Total (i.e. Null);  140689 Residual
## Null Deviance:       147700
## Residual Deviance: 87640     AIC: 87660
```

```
summary(compglm$fitted.values)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0002542 0.0242564 0.0687108 0.2183187 0.4322985 0.9764992
```

```
hist(compglm$fitted.values)
```




```
length(comptraindata$in_train[comptraindata$in_train == 'True'])
```

```
## [1] 30717
```

```
length(compglm$fitted.values[compglm$fitted.values >= .5])
```

```
## [1] 30813
```

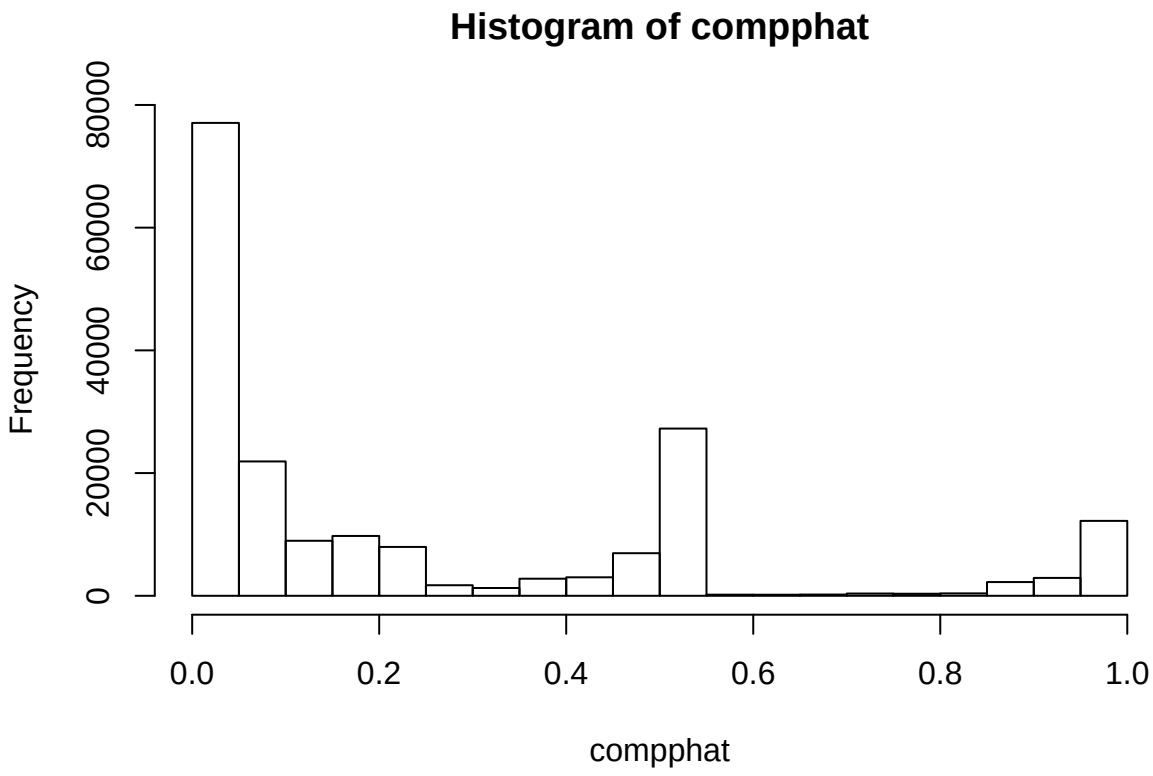
```
##training misclass rate on companies (2013)
comptrainingmisclass = numeric(140698)

for (i in 1:140698)
{
if (compglm$fitted.values[i] >= .5 && comptraindata$in_train[i] == 'False')
{
comptrainingmisclass[i] = 1
}
}

##  training misclass rate on companies
mean(comptrainingmisclass)
```

```
## [1] 0.08481286
```

```r
### orgs/companies test data (all orgs from 2014 and their features)
comptestdata = read.csv("CompaniesTestSet.csv", header = TRUE)

newdat2 = data.frame(comptestdata$numWords, comptestdata$stock, comptestdata$price, comptestdata$founde
names(newdat2)[1] <- "numWords"
names(newdat2)[2] <- "stock"
names(newdat2)[3] <- "price"
names(newdat2)[4] <- "founder"
names(newdat2)[5] <- "part"
names(newdat2)[6] <- "comp"
names(newdat2)[7] <- "comps"
names(newdat2)[8] <- "forbes"


### fitted values on test set of companies
compphat = predict(compglm, newdata = newdat2, type = "response")
summary(compphat)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0004083 0.0242564 0.0831457 0.2461263 0.4714171 0.9764992
```

```r
hist(compphat)
```




```r
### test set misclass rate
comptestmisclass = numeric(187651)

for (i in 1:187651)
{
if (as.numeric(compphat[i]) >= .5 && comptestdata$in_train[i] == 'False')
{
```

```
comptestmisclass[i] = 1
}
}

### test set misclass rate on identifying orgs to companies
mean(comptestmisclass)
```

```
## [1] 0.08939467
```

```
### final list of Companies are Organizations that the logistic regression model classified as
### orgnizations for both the training 2013 set and the test 2014 set with duplicates removed

finalcomps = numeric(328349)


for (i in 1:140698)
{
if (compglm$fitted.values[i] >= .5)
{
finalcomps[i] = as.character(comptraindata$org[i])
}
else
{
  finalcomps[i] = 0
}
}

for (i in 1:187651)
{
if (compphat[i] >= .5)
{
finalcomps[i + 140698] = as.character(comptestdata$org[i])
}
else
{
  finalcomps[i + 140698] = 0
}
}


finalcomps[finalcomps == 0] = NA
finalcomps = na.omit(finalcomps)

finalcomps = unique(finalcomps) ###remove duplicates

write.csv(finalcomps, "finalfinalcomplist.csv")
```