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
```

```
##misclass rate
ceotrainingmisclass = numeric(70188)

for (i in 1:70188)
{
if (glm1$fitted.values[i] >= .5 && data$in_train[i] == 0)
{
```

```
ceotrainingmisclass[i] = 1
}
}

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
```

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



##predicted values on test data
ceophat = predict(glm1, newdata = newdat, type = "response")
summary(ceophat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.54    0.54    0.54    0.60    0.54    1.00   62331
```

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

for (i in 1:67168)
{
if (as.numeric(ceophat[i]) >= .5 && testdata$in_train[i] == 0)
{
ceotestmisclass[i] = 1
}
}

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

```
## [1] 0
```

```
### 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 ~ forbes + numWords, family = binomial(link="logit"), data = comptraindata)
compglm
```

```
##
## Call:  glm(formula = in_train ~ forbes + numWords, family = binomial(link = "logit"),
##     data = comptraindata)
##
## Coefficients:
## (Intercept)   forbesTrue     numWords
##     -0.8105       3.8948      -0.5701
```

```
##
## Degrees of Freedom: 140697 Total (i.e. Null);  140695 Residual
## Null Deviance:       147700
## Residual Deviance: 114600    AIC: 114600
```

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1245  0.2009  0.2183  0.2009  0.9251
```

```
##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] 0.007370396
```

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

newdat2 = data.frame(comptestdata$numWords, comptestdata$forbes)
names(newdat2)[1] <- "numWords"
names(newdat2)[2] <- "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.
## 0.0000001 0.1244636 0.2009002 0.2322639 0.2009002 0.9251226
```

```
### 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] 0.01459092
```

```
finalceos = numeric()

for (i in 1:3542)
{
if (glm1$fitted.values[i] >= .5)
{
finalceos[i] = as.character(data$NAME[i])
}
else
{
  finalceos[i] = 0
}
}

length(ceophat)
```

```
## [1] 67168
```

```
for (i in 1:3542)
{
if (glm1$fitted.values[i] >= .5)
{
finalceos[i] = as.character(data$NAME[i])
}
else
{
  finalceos[i] = 0
}
}
```