Data mining · Computational Task 1

In this task we are asked to examine if there are any social, political and economical dynamics driving the results of the U.S. presidential elections. Or rather, based on 12 question from the aforementioned areas, see if we can predict a win of the incumbent party (Democratic / Republican party) or the challenger. We shall attempt to construct a primitive neural-like network that would identify questions with the highest predictive power, which in our case would be estimated, rather as, consistency.

Methodology and data description

The method we will use is rather straightforward. For each question, we look how many answers "predict" the outcome of presidential party winning, or opposition winning. If, say out of 30 observations, if a question is "y" and presidential wins 15 times and also opposition wins 15 times, then we cannot conclude that the results "y" for given questions might predict any outcome. If, however, the difference is high, for example 25 - 5 to either party, then we can say that should that question be answered "y", then it correctly predicts the party winning.

Hence, for part **1)**, we look at how many times, each answer implies winning a party. Then we pick as correct number of predictions the higher number of the two. Markedly, Question 4 about competition in the incumbent party, gives the most consistent predictions - let us use it to illustrate the algorithm; if Q4 is yes then the presidential party wins 1x, oppositions

10x times - hence, we count 10 correct predictions from the answer "yes". In case "n", it 16x times precedes the win of the presidential party, 3x the opposition. Therefore, for that question, we sum up 26 correct predictions, since "no" consistently predicts the incumbent, and "yes" the challenger. For counts of all questions, refer to Table 4.

Similarly, we can extend this algorithm on an arbitrary number of questions joint together as is asked in parts b and c. The algorithm is the same, it only considered combinations of the two or three questions respectively as presented below.

We assume, that the election mechanism and bipartism has not changed significantly since 1860. In the data, the presidential party has had 17 victories since 1864, the opposition 13 victories from overall 30 elections until 1980. Let us present the simple descriptive statistics of the dataset

Table 1: Presidential wins - question counts

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
y	8	11	2	1	14	2	10	10	2	1	6	2
n	9	6	15	16	3	15	7	7	15	16	11	15

Table 2: Oppositions wins - question counts

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
y	10	6	5	10	5	5	5	4	7	4	2	5
n	3	7	8	3	8	8	8	9	6	9	11	8

This gives us the basic idea about importance of each answer to the respective questions. We can use the algorithm to choose the best single answer.

Table 3: Best question output

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
Correct	19	18	20	26	22	20	18	19	22	20	17	20
Mistakes	11	12	10	4	8	10	12	11	8	10	13	10

Table 4: Questions counts

	Q1		Q2		Q3		Q4		Q5		Q6	
	pres	opp	pres	opp	pres	opp	pres	opp	pres	opp	pres	opp
y	8	10	11	6	2	5	1	10	14	5	2	5
n	9	3	6	7	15	8	16	3	3	8	15	8

	Q7		Q8		Q9		Q10		Q11		Q12	
	pres	opp	pres	opp	pres	opp	pres	opp	pres	opp	pres	opp
y	10	5	10	4	2	7	1	4	6	2	2	5
n	7	8	7	9	15	6	16	9	11	11	15	8

1) The top table shows the correct and incorrect predictions for each question, and below we can find the respective marginal distributions. Clearly, the most consistent is Question 4, with only 4 mistakes. Let us try to pair it with another question to see if we get a higher number of correct predictions. We can assume, that it won't be lower than 26. Firstly, it would not make sense to consider if it did, but we can think of the counts as: number of predictions given Q4 is either "y" or "n". Hence, Q4 would bring its predictive power into any pair.

The question 4 reads: "Was there a serious competition in P-party primaries?". If there was, it might imply a turbulent situation (as we see for example today). Therefore, "yes" predicted 10/13 times the challenger to win, on the other hand if the situation was calm, the incumbent took advantage of it - it might also suggest re-election.

Table 5: Predictions outcomes - paired with Q4

	Q1	Q2	Q3	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
Correct	26	26	26	26	26	26	26	26	26	26	27
Mistakes	4	4	4	4	4	4	4	4	4	4	3

Table 6: Pairwise output

	Q1		Q2		Q3		Q5		Q6		Q7	
$Qi Q_4$	pres	opp	pres	opp	pres	opp	pres	opp	pres	opp	pres	opp
yn	7	2	11	2	2	1	14	2	2	2	9	0
ny	0	2	1	6	1	6	1	7	1	7	0	5
yy	1	8	0	4	0	4	0	3	0	3	1	5
nn	9	1	5	1	14	2	2	1	14	1	7	3

	Q8		Q9		Q10		Q11		Q12	
$Qi Q_4$	pres	opp	pres	opp	pres	opp	pres	opp	pres	opp
yn	9	1	2	2	1	0	6	0	2	3
yn	0	7	1	5	1	6	1	8	1	8
yy	1	3	0	5	0	4	0	2	0	2
nn	7	2	14	1	15	3	10	3	14	0

2) Applying the same algorithm, we are inspecting if any question improves the predictive ability of Q4. We obtain that Q12 improves the correct number of predictions to 27. It concerns if the "O-party candidate was a national hero?", which is hard to interpret and might be just a result of coincidence - firstly, it is hard to define who is a national hero, and secondly, why did Q11 not improve our tests as well, if national heroism were important? In case of the challenger, there were only 2 "heroes", whereas in P-party we find 6 "heroes". It is left for further investigation. However, we select Q12 to be paired with Q4.

Table 7: Predictions outcomes – paired with Q12,Q4

	Q1	Q2	Q3	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Correct	27	27	27	27	28	28	27	28	27	27
Mistakes	3	3	3	3	2	2	3	2	3	3

Table 8: Counts of all possibilities

	Q1		Q2		Q3		Q5		Q6	
$Q_iQ_{12}Q_4$	pres	opp	pres	opp	pres	opp	pres	opp	pres	opp
yyn	1	2	2	2	0	1	1	2	0	2
nyn	1	1	0	1	2	2	1	1	2	1
yny	1	7	0	4	0	3	0	2	0	2
nny	0	1	1	4	1	5	1	6	1	6
yyy	0	1	0	0	0	1	0	1	0	1
nyy	0	1	0	2	0	1	0	1	0	1
ynn	6	0	9	0	2	0	13	0	2	0
nnn	8	0	5	0	12	0	1	0	12	0
	Q7		Q8		Q9		Q10		Q11	
$Q_iQ_{12}Q_4$	pres	opp	pres	opp	pres	opp	pres	opp	pres	opp
yyn	1	0	1	1	0	2	0	0	0	0
nyn	1	3	1	2	2	1	2	3	2	3
yny	1	4	1	3	0	5	0	2	0	2
nny	0	4	0	5	1	3	1	6	1	6
yyy	0	1	0	0	0	0	0	2	0	0
nyy	0	1	0	2	0	2	0	0	0	2
ynn	8	0	8	0	2	0	1	0	6	0
nnn	6	0	6	0	12	0	13	0	8	0

3) By the same token, we compare the triple-wise combinations of questions with Q4 and Q12. The number of correct predictions has increased up to 28/30 and in three cases.

We select Q6 – "Was there a depression or recession in the election year?" – where the answer "no", along with the challenger not being a national hero and calm situation within the incumbent party predicts P-victory 12 times. It follows our intuition, that the economic dynamic would have impact on the likelihood of the presidential party winning.

Also, Q7 – "Was there a growth in the GNP of more than 2.1% in the election year?" – follow the same logic as Q6. However, if it is "yes", the incumbent wins 8x times, in case of "no", he wins 6x times, which is almost random.

Lastly, Q9 – "Did significant social tension exist during the term of the P-party?" – In case there wasn't, then the incumbent wins each time provided Q4="n", Q12="n", which follows our intuition, that a non-turbulent situation within the party with non-heroic opponent (proxy for challenger's excessive popularity) is advantageous to the presidential party.

Conclusion

This project attempts to unravel dynamics driving american presidential election and predict its outcome based on a set of questions. It finds three sets of questions to predict 28 out of 30 outcomes - they are $(Q6, Q12, Q4)$, $(Q7, Q12, Q4)$ and $(Q9, Q12, Q4)$ respectively. It would be possible, to select a subset of the data to create a training set and then try to implement some sort of training, or at least test in-sample forecast power, but that is left for further efforts.

Reference :

R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Hlavac, Marek (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables. R package version 5.2.http://CRAN.R-project.org/package=stargazer

```
#Computational task 1 - Data mining and neural networks
setwd("~/Desktop/Data mining")

library(xlsx)
elections<-read.csv("elections_dataset.csv", sheetIndex=1, header = TRUE, as.data.frame=TRUE)
elections<-elections_dataset #if data added manually
rm(elections_dataset)

#presidential victories
p<-data.frame(elections[which(elections$Winner=="p"),])
#opposition victories
o<-data.frame(elections[which(elections$Winner=="o"),])

myFreq(elections)

require(stargazer)

#Basic counts output
write(stargazer(myFreq(elections[which(elections$Winner=="p"),]), summary=FALSE, title="Presidential wins - ques
write(stargazer(myFreq(elections[which(elections$Winner=="o"),]), summary=FALSE, title="Oppositions wins - questi
#myFreq(p)  #counts for presidential
#myFreq(o)  #counts for opposition


write(stargazer(findQ(), summary=rep(FALSE,2), title="Best questions output"), file="questions2.tex")
#findQ()    #uncomment to see the output without writing a file

  # Q2
# Run through the data fixing Q4 and compute counts for combinations all combinations
write(stargazer(pairs(),summary=rep(FALSE,3), title="Pairs question counts (best selected)"), file="pairs2.tex")
#pairs()    #uncomment to see the output without writing a file

# Q3
write(stargazer(triple(),summary=rep(FALSE,3), title="Trios question counts (best selected)"), file="triplets2.te
#triple() #uncomment to see the output without writing a file



#########################################################################
#                              Functions                                #
#########################################################################


findQ<-function(data=elections){
  q1p<-myFreq(data[1:17,])
  q1o<-myFreq(data[18:30,])
  correctOut<-c()

  bigtable<-c()


for(i in 1:12){
  y<-q1p[1,i]
  yo<-q1o[1,i]
  n<-q1p[2,i]
  no<-q1o[2,i]
```

```
 correct<-c(max(y,yo),max(n,no))
 correctOut<-cbind(correctOut,sum(correct))
 bigtable<-as.data.frame(cbind(bigtable,cbind(rbind(y,n),rbind(yo,no))))
}

correctOut<-as.data.frame(rbind(correctOut,30-correctOut), row.names=c("Correct","Mistakes"))
colnames(correctOut)<-colnames(elections)[2:13]
odd<-seq(from=1,by=2,to=23)
even<-seq(from=2,by=2,to=24)
names<-rep("a",24)
names[odd]<-c("pres")
names[even]<-c("opp")
colnames(bigtable)<-names
 return(list(predictions=correctOut,counts=bigtable)) }


myFreq<-function(data=elections){
  out<-c()
  for(i in 1:12){
    temp<-c(length(which(data[,i+1]=="y")),length(which(data[,i+1]=="n")))
    out<-cbind(out,temp)}
  colnames(out)<-colnames(data[,2:13])
  rownames(out)<-c("y","n")
  return(as.data.frame(out))
  }

######################

pairs<-function(data=elections){
q4p<-data[1:17,]$Q4
q4o<-data[18:30,]$Q4
p<-data[1:17,]
o<-data[18:30,]

subsetp<-cbind(p[,2:4],p[,6:13])
subseto<-cbind(o[,2:4],o[,6:13])

require(plyr)
bigtable<-c()
correctOut<-c()
for(i in 1:11){
#columnbind the columns and counts occurences of each
pairs<-paste(subsetp[,i],q4p,sep="")
count<-count(as.data.frame(pairs))
count<-data.frame(freq=count$freq,row.names=c(as.character(count$pairs)))

yn<-count["yn",]
if(is.na(yn)) yn<-0
ny<-count["ny",]
if(is.na(ny)) ny<-0
yy<-count["yy",]
if(is.na(yy)) yy<-0
nn<-count["nn",]
if(is.na(nn)) nn<-0
#porovnat ty yy,yn,ny,nn pro obe otazky a zase hledat pro ktery krize je tam nejvetsi cislo...
```

```
pairs2<-paste(subseto[,i],q4o,sep="")
counto<-count(as.data.frame(pairs2))
counto<-data.frame(freq=counto$freq,row.names=c(as.character(counto$pairs2)))
yno<-counto["yn",]
if(is.na(yno)) yno<-0
nyo<-counto["ny",]
if(is.na(nyo)) nyo<-0
yyo<-counto["yy",]
if(is.na(yyo)) yyo<-0
nno<-counto["nn",]
if(is.na(nno)) nno<-0

correct<-c(max(yn,yno),max(ny,nyo),max(yy,yyo),max(nn,nno))
correctOut<-cbind(correctOut,sum(correct))
bigtable<-as.data.frame(cbind(bigtable,cbind(rbind(yn,ny,yy,nn),rbind(yno,nyo,yyo,nno))))
}
odd<-seq(from=1,by=2,to=21)
even<-seq(from=2,by=2,to=22)
names<-rep("a",22)
names[odd]<-c("pres")
names[even]<-c("opp")
correctOut<-as.data.frame(correctOut)
colnames(correctOut)<-colnames(subsetp)
colnames(bigtable)<-names

return(list(correct=correctOut,mistakes=30-correctOut, bigtable=bigtable))
}



triple<-function(data=elections){
  q4p<-data[1:17,]$Q4
  q4o<-data[18:30,]$Q4
  q12p<-data[1:17,]$Q12
  q12o<-data[18:30,]$Q12
  p<-data[1:17,]
  o<-data[18:30,]

  subsetp<-cbind(p[,2:4],p[,6:12])
  subseto<-cbind(o[,2:4],o[,6:12])

  require(plyr)
  correctOut<-c()
  bigtable<-c()

  for(i in 1:10){
    #columnbind the columns and counts occurences of each
    pairs<-paste(subsetp[,i],q12p,q4p,sep="")
    count<-count(as.data.frame(pairs))
    count<-data.frame(freq=count$freq,row.names=c(as.character(count$pairs)))

    yyn<-count["yyn",]
    if(is.na(yyn)) yyn<-0
    nyn<-count["nyn",]
    if(is.na(nyn)) nyn<-0
```

```
    yny<-count["yny",]
    if(is.na(yny)) yny<-0
    nny<-count["nny",]
    if(is.na(nny)) nny<-0
    yyy<-count["yyy",]
    if(is.na(yyy)) yyy<-0
    nyy<-count["nyy",]
    if(is.na(nyy)) nyy<-0
    ynn<-count["ynn",]
    if(is.na(ynn)) ynn<-0
    nnn<-count["nnn",]
    if(is.na(nnn)) nnn<-0
    #porovnat ty yy,yn,ny,nn pro obe otazky a zase hledat pro ktery krize je tam nejvetsi cislo...

    pairs2<-paste(subseto[,i],q12o,q4o,sep="")
    counto<-count(as.data.frame(pairs2))
    counto<-data.frame(freq=counto$freq,row.names=c(as.character(counto$pairs2)))

    yyno<-counto["yyn",]
    if(is.na(yyno)) yyno<-0
    nyno<-counto["nyn",]
    if(is.na(nyno)) nyno<-0
    ynyo<-counto["yny",]
    if(is.na(ynyo)) ynyo<-0
    nnyo<-counto["nny",]
    if(is.na(nnyo)) nnyo<-0
    yyyo<-counto["yyy",]
    if(is.na(yyyo)) yyyo<-0
    nyyo<-counto["nyy",]
    if(is.na(nyyo)) nyyo<-0
    ynno<-counto["ynn",]
    if(is.na(ynno)) ynno<-0
    nnno<-counto["nnn",]
    if(is.na(nnno)) nnno<-0


    correct<-c(max(yyn,yyno),max(nyn,nyno),max(yny,ynyo),max(nny,nnyo),max(yyy,yyyo),max(nyy,nyyo),max(ynn,ynno)
    correctOut<-cbind(correctOut,sum(correct))
    bigtable<-as.data.frame(cbind(bigtable,cbind(rbind(yyn,nyn,yny,nny,yyy,nyy,ynn,nnn),rbind(yyno,nyno,ynyo,nnyo
  }

  odd<-seq(from=1,by=2,to=19)
  even<-seq(from=2,by=2,to=20)
  names<-rep("a",20)
  names[odd]<-c("pres")
  names[even]<-c("opp")

  correctOut<-as.data.frame(correctOut)
  colnames(correctOut)<-colnames(subsetp)
  colnames(bigtable)<-names
  return(list(correct=correctOut,mistakes=30-correctOut, bigtable=bigtable))
}
```