

Main Script - Project 5

Group 14: Boya Zhao, Liangbin Chen, Yaqin Li, Yi Jiang

4/27/2017

Part 1: Problem discription

Where else but Quora can a physicist help a chef with a math problem and get cooking tips in return? Quora is a place to gain and share knowledge—about anything. It's a platform to ask questions and connect with people who contribute unique insights and quality answers. This empowers people to learn from each other and to better understand the world.

Over 100 million people visit Quora every month, so it's no surprise that many people ask similarly worded questions. Multiple questions with the same intent can cause seekers to spend more time finding the best answer to their question, and make writers feel they need to answer multiple versions of the same question. Quora values canonical questions because they provide a better experience to active seekers and writers, and offer more value to both of these groups in the long term.

So we tackled this natural language processing problem by applying advanced techniques to classify whether question pairs are duplicated or not. Doing so will make it easier to find high quality answers to questions resulting in an improved experience for Quora writers, seekers, and readers.

Part 2: Data exploration

```
setwd("~/Desktop/sem 2/Applied data science/Spr2017-proj5-grp14")
Data <- read.csv("~/Desktop/sem 2/Applied data science/Spr2017-proj5-grp14/data/train.csv", header = TRUE)
head(Data)
```

```
##   id qid1 qid2
## 1  0     1     2
## 2  1     3     4
## 3  2     5     6
## 4  3     7     8
## 5  4     9    10
## 6  5    11    12
##
##                                     question1
## 1                      What is the step by step guide to invest in share market in india?
## 2                      What is the story of Kohinoor (Koh-i-Noor) Diamond?
## 3          How can I increase the speed of my internet connection while using a VPN?
## 4                      Why am I mentally very lonely? How can I solve it?
## 5          Which one dissolve in water quikly sugar, salt, methane and carbon di oxide?
## 6 Astrology: I am a Capricorn Sun Cap moon and cap rising...what does that say about me?
##
##                                     question2
## 1                      What is the step by step guide to invest in share market?
## 2  What would happen if the Indian government stole the Kohinoor (Koh-i-Noor) diamond back?
## 3                      How can Internet speed be increased by hacking through DNS?
## 4          Find the remainder when  $23^{24}$  is divided by 24,23?
## 5                      Which fish would survive in salt water?
## 6 I'm a triple Capricorn (Sun, Moon and ascendant in Capricorn) What does this say about me?
##   is_duplicate
```

```
## 1      0
## 2      0
## 3      0
## 4      0
## 5      0
## 6      1
```

Part 3: How we did it

Constructed feature

We constructed four features:

Similarity: counted the number of words in each question, measured the similarity between each pair of questions, including the number of same verbs and nouns.

Interrogative word: which pair of interrogative words showed up in each pair of questions.

Parsing: measured the difference of parsing result in each pair of questions.

Basic properties: determine the amount of punctuations, modals and negative words, such as “non” and “not”.

Sentiment analysis: analysis the sentiment components in the sentences (based on the “bing” vocabulary).

```
feature <- read.csv("../output/allfeatures.csv", header = TRUE)
head(feature)
```

```
##   X len.1 len.2 rem.len.1 rem.len.2 num.same len.ratio len.v.1 len.v.2
## 1 1    14    12         7         6 0.8571429 0.8571429      2      2
## 2 2     8    13         4         8 0.3333333 0.6153846      1      2
## 3 3    14    10         7         6 0.3000000 0.7142857      2      3
## 4 4    11     9         4         5 0.0000000 0.8181818      2      3
## 5 5    13     7        10         4 0.1666667 0.5384615      1      1
## 6 6    16    16         8         7 0.5000000 1.0000000      4      2
##   len.n.1 len.n.2 rem.len.v.1 rem.len.v.2 rem.len.n.1 rem.len.n.2 v.ratio
## 1       7       6           1           1           6           5      1.0
## 2       3       3           0           2           3           3      0.0
## 3       3       3           2           2           3           3      0.0
## 4       1       1           1           2           1           1      0.0
## 5       9       4           1           1           8           3      0.0
## 6       6       5           2           1           6           5      0.5
##   n.ratio V1.1 V2.1 V3.1 V4.1 V5.1 V6.1 V7.1 V8.1 V9.1 V10.1 V11.1 V12.1
## 1 0.8333333 1 0 0 0 0 0 0 0 0 0 0 0
## 2 0.5000000 1 0 0 0 0 0 0 0 0 0 0 0
## 3 0.2000000 0 0 0 0 0 0 0 0 0 1 0 0
## 4 0.0000000 0 0 0 0 0 0 0 0 0 1 1 0
## 5 0.2222222 0 0 0 0 0 0 0 0 0 0 0 0
## 6 0.5714286 0 0 0 0 0 0 0 0 0 0 0 0
##   V13.1 V14.1 V15.1 V16.1 V17 V18 V19 V20 V21 V22 V23 V24 V25 V26 V27 V28
## 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 4 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
## 5 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
```

```

## 6      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
##      V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40 V41 V42 V43 V44 V45 punc
## 1      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
## 2      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
## 3      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
## 4      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
## 5      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      -2
## 6      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      -3
##      modal      length neg sentiment positive negative CC CD DT EX FW IN JJ
## 1      0 -0.07692308      0      0      0      0      0 0 0 0 0 0 0 1 0
## 2      0  0.23809524      0      0      1      1      0 0 0 -1 0 0 0 -2
## 3      0 -0.16666667      0      0      0      0      0 0 0 0 2 0 0 0 1
## 4      0 -0.10000000      0      1      0      -1      0 -1 0 -1 -1 0 0 -1 0
## 5      0 -0.30000000      0      1      1      0      1 0 0 0 0 0 0 0 0
## 6      0  0.00000000      0      0      0      0      0 0 0 0 -1 0 0 0 -1
##      JJR JJS LS MD NN NNS NNP NNPS PDT POS PRP PRP. RB RBR RBS RP SYM TO UH
## 1      0      0      0      0      1      0      0      0      0      0      0      0      0      0      0      0      0
## 2      0      0      0      -1      -1      0      2      0      0      0      0      0      -1      0      0      0      0
## 3      0      0      0      0      1      -1      0      0      0      0      1      1      0      0      0      0      0
## 4      0      0      0      1      -1      0      0      0      0      0      3      0      2      0      0      0      -1
## 5      0      0      0      -1      5      0      0      0      0      0      0      0      1      0      0      0      0
## 6      0      0      0      0      2      0      0      0      0      0      0      0      0      0      0      0      0
##      VB VBD VBG VBN VBP VBZ WDT WP WP. WRB . X.LRB. X.RRB. X. X..1 X.. X...1
## 1      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
## 2     -1     -1      0      0      0      1      0      0      0      0      0      0      0      0      0      0
## 3      0      0      0      -1      0      0      0      0      0      0      0      0      0      0      0      0      0
## 4      0      0      0      -1      1     -1      0      0      0      1      1      0      0      0      0      0      0
## 5      0      0      0      0      0      0      0      0      0      0      0      0      0      2      0      0      0
## 6      0      0      1      0      0      0      0     -1      0      0      0      -1      -1     -1      1      0      0

```

```
label <- Data$is_duplicate
```

Model training and results

We used several methods to train our models, including random forest, GBM, adaboost, SVM, ANN. Among all these methods, random forest has the best result.

Random forest

```

load("../output/err_rf_test.RData")
load("../output/rf_pred.RData")
err_rf_test

```

```
## [1] 0.245
```

GBM

```

load("../output/gbmresults.RData")
err_gbm_test <- results[[1]]
gbm_pred <- results[[2]]
err_gbm_test

```

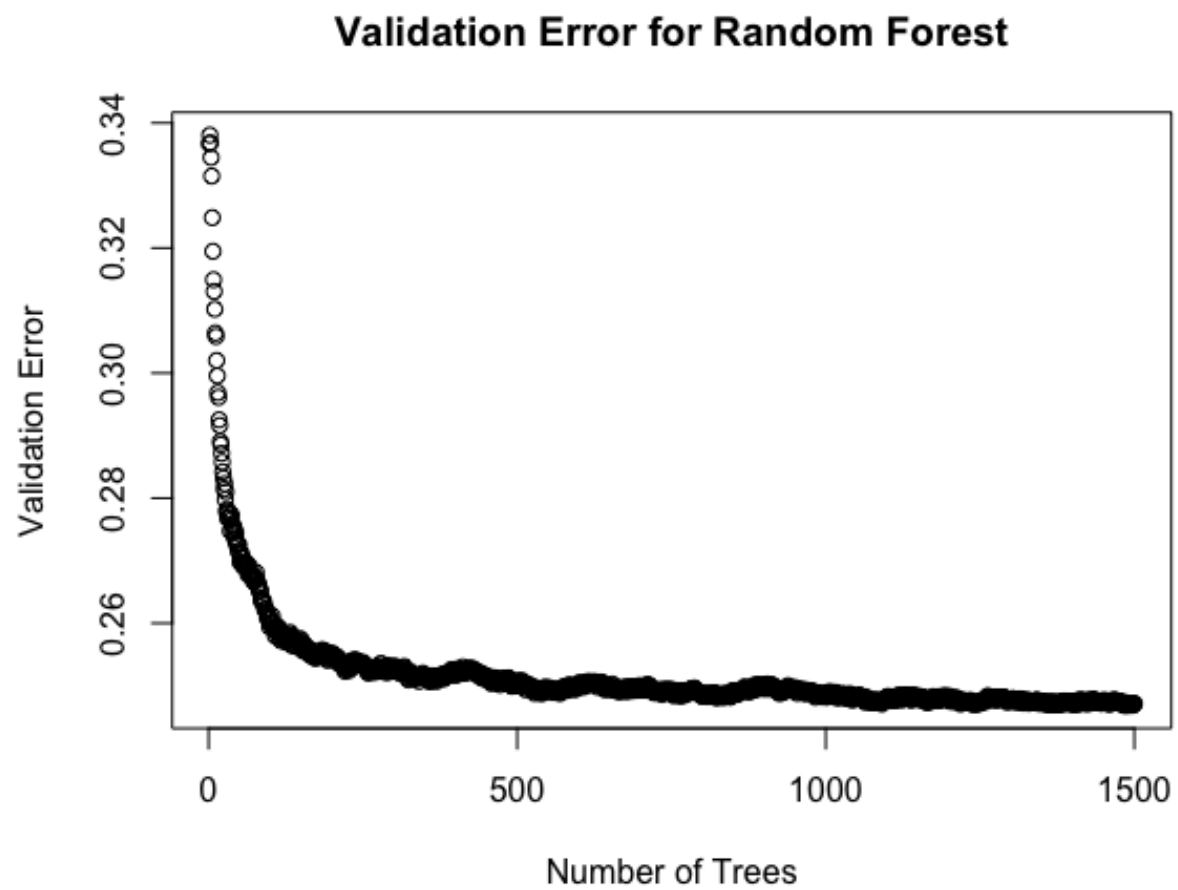


Figure 1: Figure 1: GBM Cross Validation Results

```
## [1] 0.2947
```

Part 4: Comparison

```
test.label <- label[40001:50000]  
result.rf <- table(rf_predict, test.label)  
result.rf
```

```
##           test.label  
## rf_predict    0     1  
##           0 5092 1300  
##           1 1150 2458
```

```
result.gbm <- result.gbm <- table(gbm_pred, test.label)  
result.gbm
```

```
##           test.label  
## gbm_pred    0     1  
##           0 5581 2286  
##           1  661 1472
```