Social Media Analytics - Group 6 - Prediction of online engagement : the case of Club Bruges K.V.

1) Introduction

Recently, **social media** has become a phenomenon being used by consumers around the world. With the emergence of social media, it is now possible for one person to communicate with hundreds or even thousands of other people about products and the companies that provide them.

According to Kuzma et al. (2014), football clubs are realizing that the use of social media is the future in terms of improving and expanding their business, whether this be for marketing purposes or as a medium for directly communicating with their fans. As in a classical business, interactivity between fans and a club allows to build a **long term relationships** between them.

This project is studying the case of a Belgian football club: **Club Bruges KV**. The final objective is to evaluate what elements drive the **engagement** on the Facebook page of the club. In our case, two metrics are used: the **number of like** of a posts and the **number of comments**. These metrics will be evaluated separately so that we can know if the elements that drive comments are the same as the elements driving likes.

Two approaches were used to determine the elements driving engagement. The first one is a **classification model** that determine according to the characteristics of the post, if the number of likes/comments for this post will be higher than a certain threshold. The second one is a **regression model** that predict, according to these characteristics the number of likes and comments. For both, the regression and the classification model, the **Random Forest** model is used.

The **R packages** needed for this project can be found here below:

```
library(dplyr)
library(tm)
library(lubridate)
library(plyr)
library(textstem)
library(leaps)
library(pROC)
library(caret)
library(ggplot2)
library(stringr)


for (i in c('SnowballC','slam','tm','RWeka','Matrix', 'lubridate', 'plyr',
'dplyr')){
```

```
  if (!require(i, character.only=TRUE)) install.packages(i, repos =
"http://cran.us.r-project.org")
  require(i, character.only=TRUE)
}

## package 'RWeka' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\rdoyen\AppData\Local\Temp\RtmpcJx5mU\downloaded_packages

if (!require("randomForest")) install.packages("randomForest", quiet=TRUE) ;
require("randomForest")
```

2) The Data

For this project we received a data sets of **6029 observations** and **9 variables**. The time range of the data is from the 11th April 2010 to the 30th August 2015. The interesting columns in this raw dataset are:

- feed_message: it contains the content of the post
- like_count: Number of like for each post
- comments_count: Number of comment for each post
- feed_created_time: Date and hour of the publication of the post

The shares_count column could also be interesting to predict the online engagement but unfortunately, the column only contains 0 and then, this column has been deleted. The other columns have been deleted.

The average number of like is equal to 661 while the average number of comments is equal to 44.Moreover, since a log transformation has been performed, as explained in the next point, the posts with 0 comments has been deleted.

Moreover, we noticed that the dataset contains some rows with a blank message, it probably comes from an error during the exportation of the dataset, these lines have been deleted.

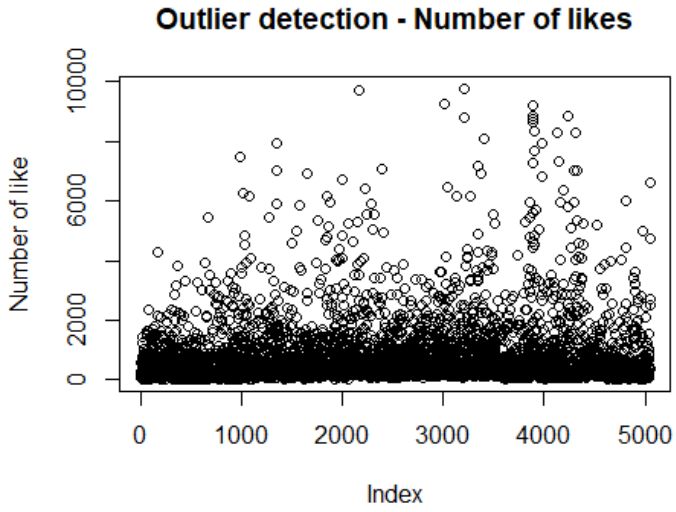
```
str(postsBASE)

## Classes 'tbl_df', 'tbl' and 'data.frame':    6209 obs. of  9 variables:
##  $ ID               : chr  "3572" "1779" "1778" "1777" ...
##  $ feed_id          : chr  "356654128253_10150501890358254"
"356654128253_10150514517828254" "356654128253_10150514971618254"
"356654128253_10150514982453254" ...
##  $ feed_message     : chr  "" "De Club-delegatie is geland in Marbella,
klaar om te starten met de voorbereidingen op een prachtig 2012!" "Ons shirt
hangt hier in het museum naast dat van CSKA Moskou, Sparta Praag, Rode Ster
Belgrado, de Oostenrijkse"| __truncated__ "En zoals beloofd, een beeld van op
het trainingsveld: One touch play met Vadis en Donk!" ...
##  $ like_count       : chr  "89" "227" "246" "282" ...
```

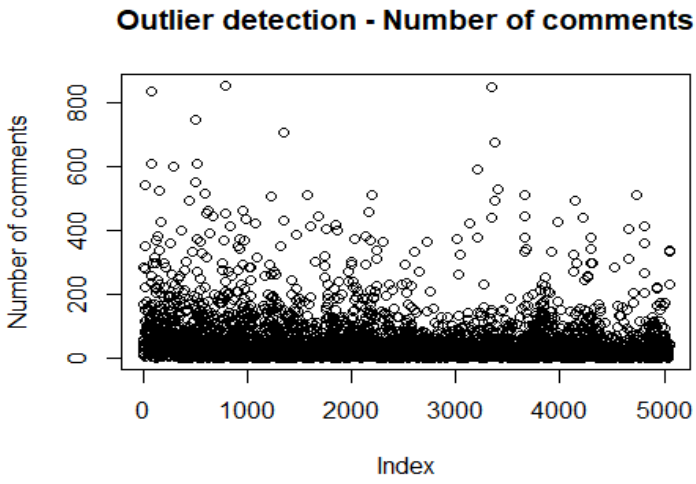
```
##  $ comments_count   : chr  "6" "20" "26" "41" ...
##  $ shares_count     : chr  "0" "0" "0" "0" ...
##  $ page_name        : chr  "clubbrugge" "clubbrugge" "clubbrugge"
"clubbrugge" ...
##  $ extracted_on     : chr  "2015-08-25 14:43:30 UTC" "2015-08-24 12:18:27
UTC" "2015-08-24 12:18:27 UTC" "2015-08-24 12:18:27 UTC" ...
##  $ feed_created_time: POSIXct, format: "2011-12-28 10:02:58" "2012-01-03
11:13:30" ...
```

```
posts$id <- posts$feed_id <- posts$shares_count <- posts$page_name <-
posts$extracted_on <- posts$ID <-NULL
```

2.1) Outliers

The followed graphs allowed us to detect the presence of **outliers** in the dataset:





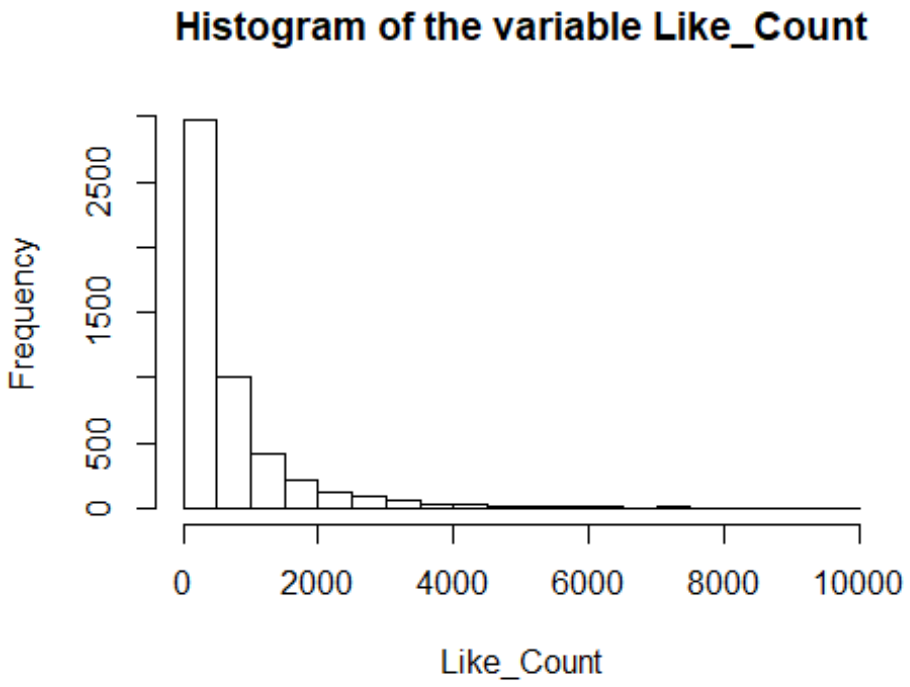
After some researches, we discovered that the number of likes we particularly high on the 22nd of March 2015 because the club won the Belgian Cup. Since this high number of likes are due to a particularly rare event, we decided to remove the outliers to avoid that these outliers influence too much our results. The same action has been taken for the comments. At the end, we removed the posts with more than 10.000 likes or with more than 900 comments.

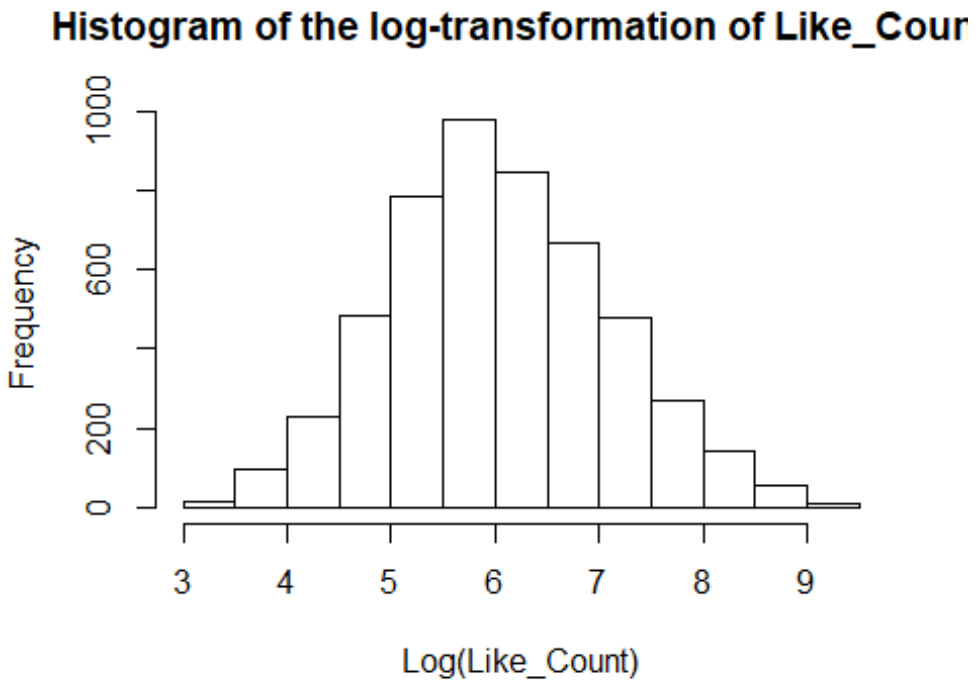
After removing the lines containing lines and the outliers, the data set is containing **5048 observations**.

2.2) Pre processing of the data

With the following histograms, we noticed that our data was **positively skewed** and the data was really imbalanced. To correct this, we decided to apply a log-transformation on the number of likes and comments. This log-transformation will be used for the regression model but not for the classification model since the way Random Forests are built is invariant to monotonic transformation of the independent variables.

Here below one can find the histogram of the number of likes before and after the log transformation.





2.3) Variables creation

Based on the literature and the information we had, we decided to create several variables, some are linked to the **content** of the posts, other are linked to the **'context'** of the publication of this posts.

2.3.1) Variables linked to the Context

- **week_day**: factor variable containing the day of the week in which the post has been written.
- **Hour**: variable containing the hour in which the post has been written.
- **Month**: variable indicating the month in which the post has been created.
- **countperday**: indicates the number of posts until now for this day. For instance, if the countperday is equal to 3, it means that two posts have been written by the club on this day before this particular post. The goal of this variables is to detect a potential spamming effect.
- **d_Matchday**: dummy variable equal to 1 if the club played a match on the day of the post creation.

2.3.2) Variables linked to the post itself

- **Length**: the number of character that the post contains.
- **numberhas**: the number of hashtag used in a post.
- **numberexl**: the number of exclamation point used in a post.
- **numberinter**: the number of question mark used in a post.

- **numberdot**: the number of dot used in a post.
- **numbercapitalletters**: the number of capital letter used in a post
- **link**: dummy variable equals to 1 if the post contains a link, 0 otherwise.
- **d_opponent**: dumy variables have been created according to the opponent Club Bruges faced. The creation of the variable is explained later in this report.
- **d_Goal**: dummy variable equals to 1 if the world goal appears in the posts. We made the assumption that this represents a moment when the team scored.
- **d_hashtagused**: We created dummy variables for the most used hashtag. The creation of the variable is explained later in this report
- **d_wordused**: We created dummy variables for the most used word. The creation of this variable is explained later in the report.
- **sentimentScore**: indicates the sentiment score of each score. Again, the way we created this variable is explained later in the report.

2.3.2.1) Creation of the variable opponent

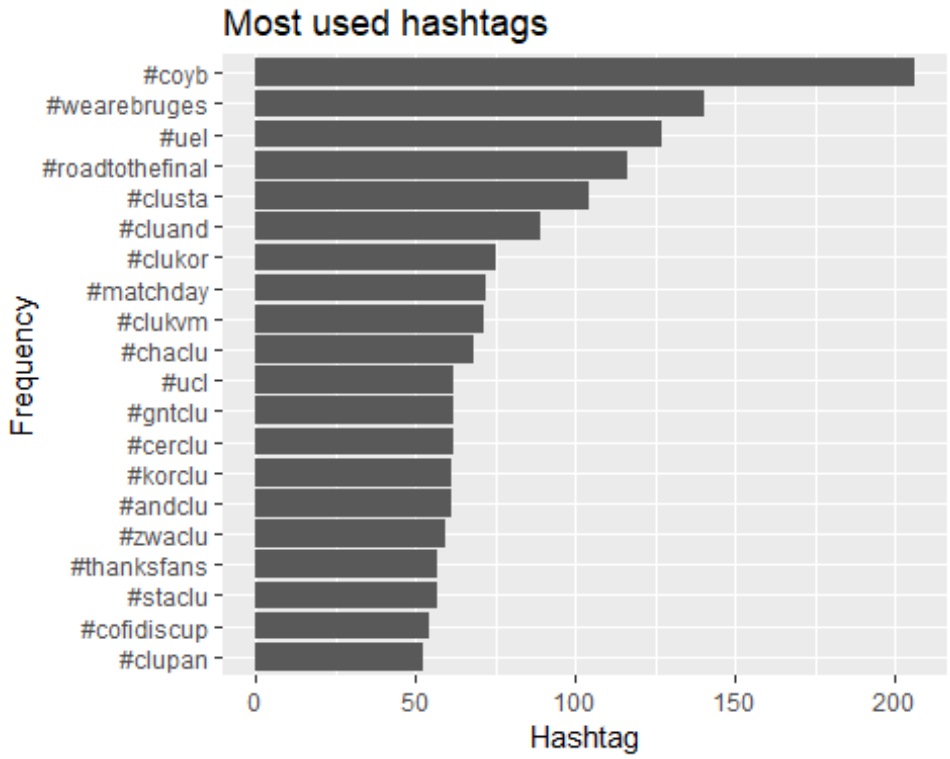
We noticed that most of the time, the club was using the same form of hashtag to comment something about a match. Indeed, most of the time, the club use either **#CluXXX** for a home match, and **#XXXClu** for a away match. Then, by using this pattern, we were able to determine which team the club faced. After some manipulations, we were able to determine all the team they faced, and we created dummy variables for the team that are mentioned at least in 20 posts.

```
library(stringr)
#Creation of the variable opponent
#We noticed that the club used all the time the same form of hashtag during
the match. #CluXXX or #XXXClu
posts$hashtags <- sapply(str_extract_all(posts$feed_message,
"#[cC]lu[^[:space:]]{3}"), paste, collapse=", ")
posts$hashtags2 <- sapply(str_extract_all(posts$feed_message,
"#[^[:space:]]{3}[cC]lu"), paste, collapse=", ")
#Removing the hashtag
posts$hashtags <- substr(posts$hashtags,2,7)
posts$hashtags2 <- substr(posts$hashtags2,2,7)
#Determining the opponent
posts$hashtags <- tolower(paste(posts$hashtags,posts$hashtags2))
posts$opponent <- ifelse(posts$hashtags2!="",posts$hashtags2,
                         ifelse(posts$hashtags=="","",posts$hashtags))
posts$hashtags2 <- posts$hashtags <- NULL
posts$opponent <- tolower(posts$opponent)
posts$opponent <-
ifelse(startsWith(posts$opponent,'clu'),substr(posts$opponent,4,6),substr(pos
ts$opponent,1,3))

count <- count(posts$opponent)
count <- count[order(count$freq,decreasing = TRUE),]

print(count[2:6,])
```

```
##      x freq
## 44 sta  160
## 3  and  148
## 28 kor  133
## 16 cha  113
## 30 kvm  105

posts$d_Standard[posts$opponent=='sta']<-1
posts$d_Standard[posts$opponent!='sta']<-0
```

2.3.2.2) Creation of the variable hashtag used

The **hastags** used may also be an important variable to determine the number of likes or comments. To do so, we wanted to know which are the most hashtag used. Here are our results:



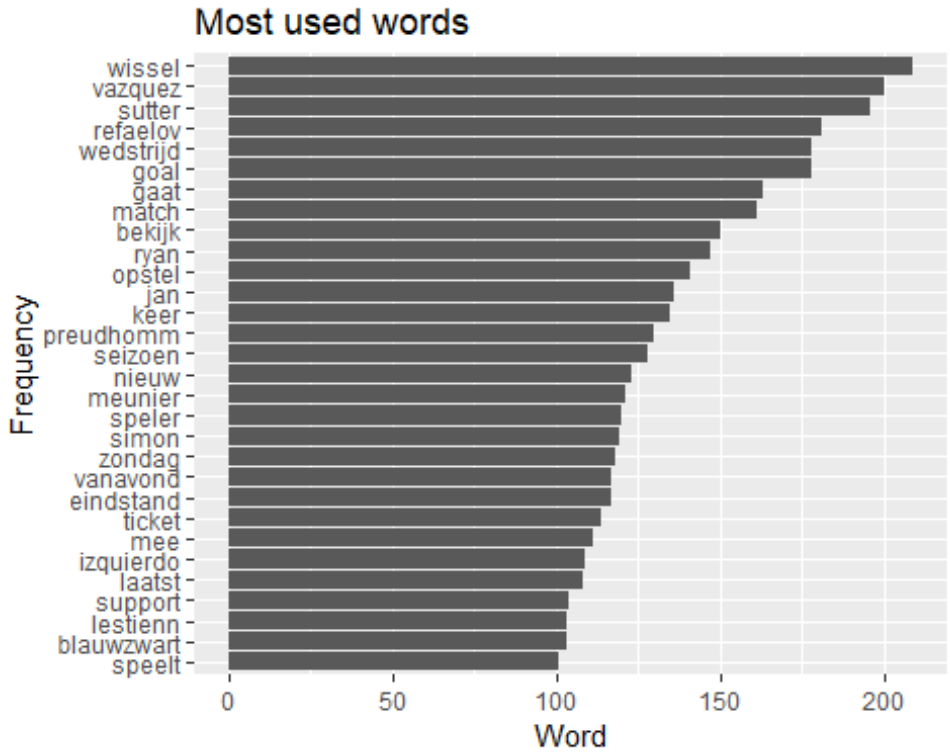
We can notice that expecting the hashtag related to a match (format #CluXXX or #XXXClu), the hashtags #COYB, #WeAreBruges, #UEL and #Roadtothefinal are widely used.

2.3.2.3) Creation of the variable words used

We also wanted to know if the use of **specific words** impact the online engagement or not. To do so, after creating a corpus, we cleaned it. We remove the links and the # used, we also remove the punctuation, the numbers and the extra white space. We also remove the stopwords and the words that did not make much sens for the analysis. Finally we also applied a stemming function.

Thanks to that we were able to create a document-term matrix that allowed us to determine the most used words. Notice that we used a sparse coefficient of 0.995 and we considered unigram, bigram and trigram.

```
## [1] "data.frame"
```



The most used word are 'wissel' (change), Vazquez (player), Sutter (player), Refaelov (player) and wedstrijd (match). Again, for the must used words, a dummy variable has been created.

2.3.2.4) Sentiment analysis

Finally, we also wanted to perform a simple **sentiment analysis** to understand if the global sentiment of the post can impact the online engagement of this post. To do so, we used the dictionary-based method and we used a dutch dictionary that informed us of the polarity of several words. The idea was to compute the average sentiment of each posts but taking into consideration the negative word that can change the meaning of a sentence (niet, geen, nee).

```
for (i in 1:length(PostsText)){
  text <-tolower(PostsText)
  split <- strsplit(text[i],split=" ")[[1]]
  m <- match(split, dictionary$ï..form)
  present <- !is.na(m)
  presentMinusone <- lead(present,default = FALSE)
  wordpolarity <- dictionary$polarity[m[present]]
  negators <- c("nee","niet","geen")
```

```
  wordnegator <-  split[presentMinusone] %in% negators
  wordpolarity <- ifelse(wordnegator == TRUE, wordpolarity*(-1),
wordpolarity)
  posts$sentimentScore[i] <- mean(wordpolarity, na.rm=TRUE)
}
```

3) Modelisation

As discussed in the introduction, this project has been separated into 2 types of model: one **classification model** and one **regression model**. In both cases a **random forest model** has been build. Furthermore, between each RF model, a **forward stepwise variable selection** has been performed.

The target data set has been separated into a training and a testing set following the following distribution: * Training: 3533 observations - 70% * Test: 1515 observations - 30%

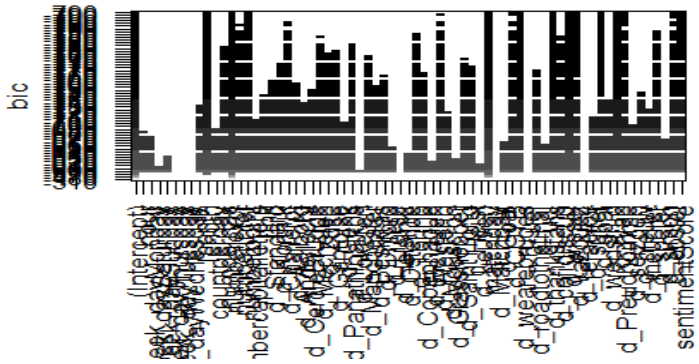
Moreover, for each a the Random Forest model, 1000 trees are explored and the number of variables selected at each split is compute with the formula tuneRF.

3.1) Classification

3.1.1) Like

The classification model is a 2-class classification problem. A posts is considering as a 1 if it has more than 500 likes. This threshold has been selected to avoid to have imbalanced dataset and thus to avoid inaccuracy due to this problem.

The forward variable selection and with the following BIC graph, we decided to keep 14 variables.



```
##  [1] "length"         "numberexcl"     "numberpoint"    "matchday"
##  [5] "d_COYB"         "d_wearebruges"  "d_thanksfans"   "d_wissel"
##  [9] "d_vazquez"      "d_goal"         "d_ryan"         "d_Preudhomme"
## [13] "d_opstel"       "sentimentScore"
```

To construct the model, we used the following way:

```
set.seed(125)
train_ind <- sample(seq_len(nrow(postsLIKE2)), size = smp_size)

train3 <- postsLIKE2[train_ind, ]
test3 <- postsLIKE2[-train_ind, ]

RF3 <- randomForest(CAT ~ ., data=train3,ntree = 1000, mtry=3)
RF3

##
## Call:
##  randomForest(formula = CAT ~ ., data = train3, ntree = 1000,      mtry =
3)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 29.61%
## Confusion matrix:
##      0   1 class.error
## 0 1836 253   0.1211106
## 1  793 651   0.5491690

predicted <- predict(RF3, test3)
```

3.1.2) Comments

The classification model is also a 2-class classification problem. A comment is considering as a 1 if it has more than 40 comments, otherwise it is considering as a 0. This threshold has been selected to avoid to have imbalanced dataset and thus to avoid inaccuracy due to this problem.

After the forward variable selection, 16 variables have been kept:

```
##  [1] "length"         "numberexcl"     "numberpoint"    "d_Charleroi"
##  [5] "d_Westerlo"     "matchday"       "d_COYB"         "d_wearebruges"
##  [9] "d_thanksfans"   "d_wissel"       "d_vazquez"      "d_goal"
## [13] "d_ryan"         "d_Preudhomme"   "d_opstel"       "sentimentScore"
```

The model has been constructed using:

```
set.seed(125)
train_ind2 <- sample(seq_len(nrow(postsCOMMENT2)), size = smp_size)
```

```
train4 <- postsCOMMENT2[train_ind2, ]
test4 <- postsCOMMENT2[-train_ind2, ]

RF4 <- randomForest(CATCOMMENT ~ ., data=train4,ntree = 1000, mtry=4)
```

3.2) Regression

The goal of the regression model is to predict the log transformation of the like and the comments. The log transformation has been chosen since the results of the prediction without the transformation were not sufficient.

3.2.1) Like

Again before doing the random forest regression, a forward variable selection has been performed. 28 variables have been kept.

```
##  [1] "hour"            "length"          "countperday"
##  [4] "numberhas"       "numberexcl"      "numberinter"
##  [7] "numberpoint"     "d_Standard"      "d_Kortrijk"
## [10] "d_Charleroi"     "d_Mechelen"      "d_Zulte"
## [13] "d_Gantoise"      "d_Panathinaikos" "d_Manchester"
## [16] "d_Dnipro"        "d_Westerlo"      "d_SaintTrond"
## [19] "matchday"        "d_COYB"          "d_wearebruges"
## [22] "d_thanksfans"    "d_wissel"        "d_vazquez"
## [25] "d_sutter"        "d_ryan"          "d_Preudhomme"
## [28] "sentimentScore"
```

Here is the model that we built:

```
RF <- randomForest(like_countLOG ~ ., data=train, importance=TRUE,ntree =
1000, mtry=5)
```

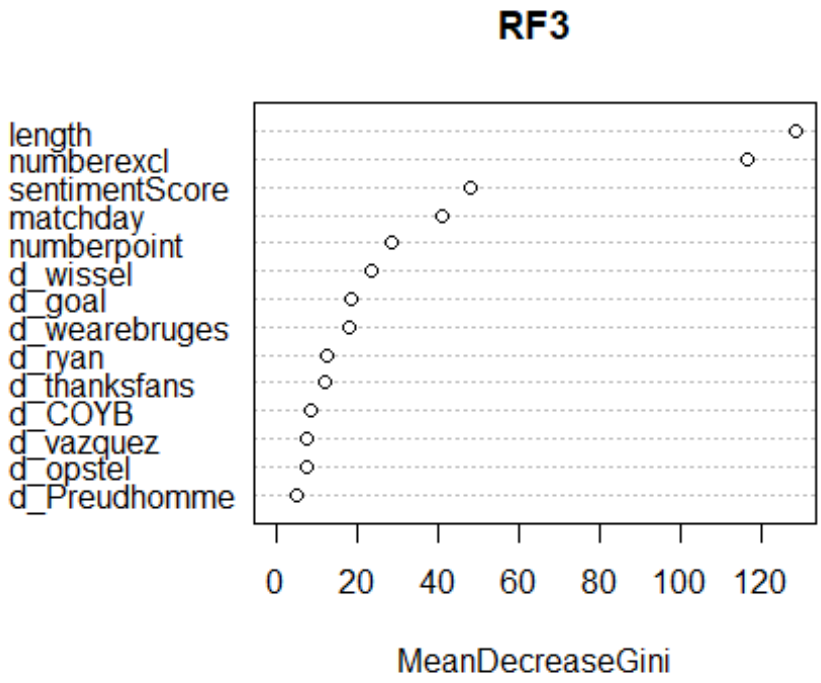
3.2.2) Comments

The same steps have been followed for the comments. Here below are the variables kept and the model built.

```
##  [1] "length"          "numberhas"       "numberexcl"
##  [4] "numberinter"     "numberpoint"     "d_Kortrijk"
##  [7] "d_Charleroi"     "d_Mechelen"      "d_Zulte"
## [10] "d_Gantoise"      "d_Panathinaikos" "d_Manchester"
## [13] "d_Dnipro"        "d_Westerlo"      "d_SaintTrond"
## [16] "matchday"        "d_COYB"          "d_wearebruges"
## [19] "d_thanksfans"    "d_wissel"        "d_vazquez"
## [22] "d_ryan"          "d_Preudhomme"    "sentimentScore"

RF2 <- randomForest(comments_countLOG ~ ., data=train2, importance=TRUE,ntree
= 1000, mtry=5)
```

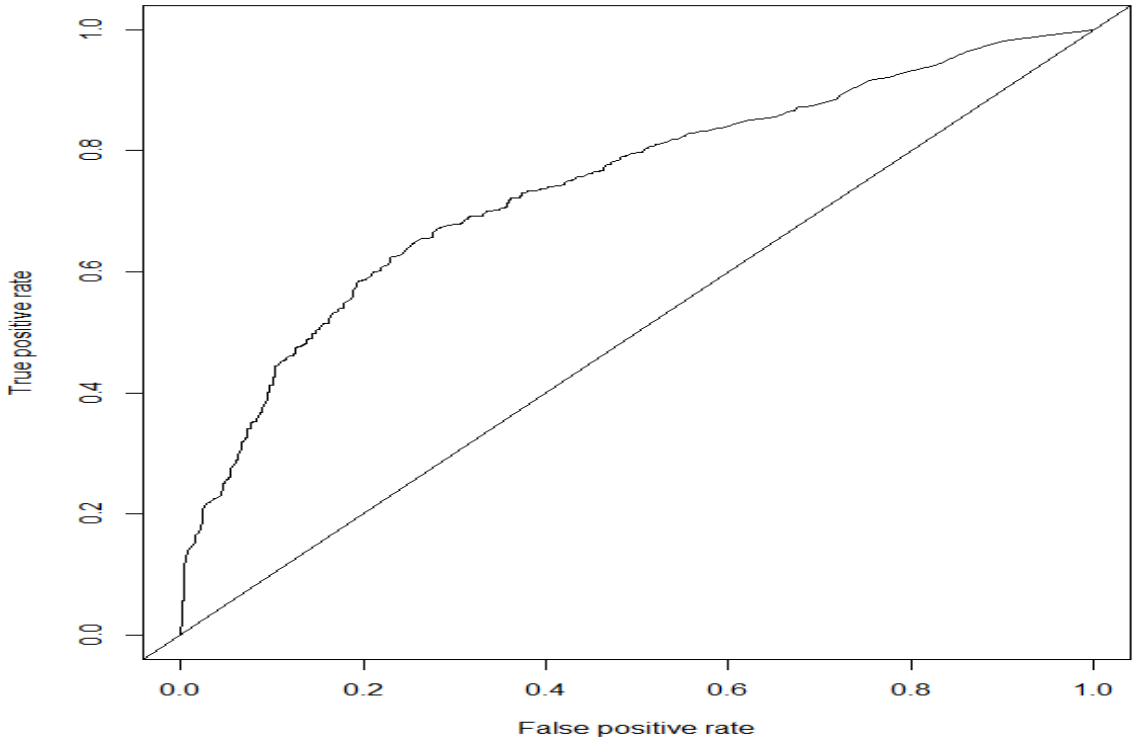
4) Results

This section will provide on overview on how the models perform. After knowing which are the most important variables for both likes comments, we will be able to compare them. ## 4.1) Classification ### 4.1.1) Like To understand which variables are important for the classification model of the likes, we used the following MeanDecreaseGini graph.



From this graph, one can see the two variables seem to be particularly important to predict the class of the post. These variables are the length of the post and the number of exclamation point. The sentiment score, the dummy variable about the match day and the number of dot in the sentence also seem to be important.

This model has a test AUC of 0.74.



Here below, one can find the confusion matrix of the test set plus some other statistics:

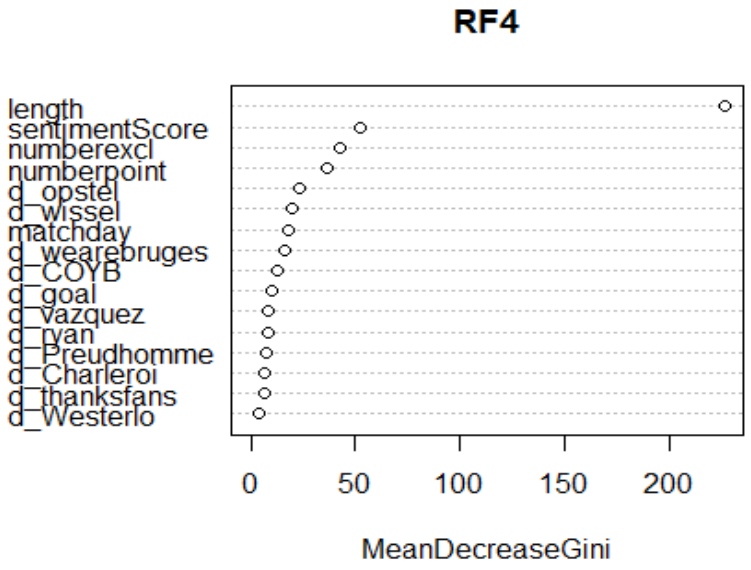
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 776 337
##          1 110 292
##
##                Accuracy : 0.705
##                  95% CI : (0.68139, 0.7362)
##     No Information Rate : 0.5848
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.29
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8758
##             Specificity : 0.4690
##          Pos Pred Value : 0.6991
##          Neg Pred Value : 0.7284
##              Prevalence : 0.5848
##          Detection Rate : 0.5122
##    Detection Prevalence : 0.7327
##       Balanced Accuracy : 0.6724
##
```

```
##        'Positive' Class : 0
##
```

The accuracy of the model is around 71%.

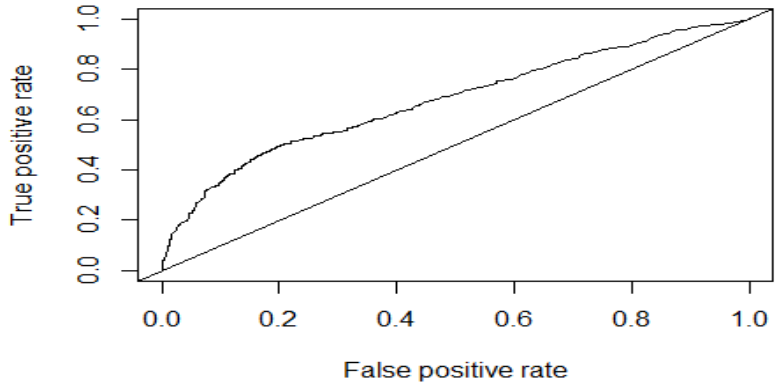
4.1.2) Comments

We are using the same graph to understand which variables have the biggest impact on the comment classification model.



We can observe that again, the length of the posts looks to be really important in the classification. Moreover, among the 5 most important variables, four a common between the comment and the like classification model.

This model has a test AUC of 0.68



Here below, one can find the confusion matrix of the test set plus some other statistics:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 887 324
##          1 111 193
##
##                Accuracy : 0.7129
##                  95% CI : (0.6894, 0.7356)
##     No Information Rate : 0.6587
##     P-Value [Acc > NIR] : 3.802e-06
##
##                   Kappa : 0.291
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8888
##             Specificity : 0.3733
##          Pos Pred Value : 0.7325
##          Neg Pred Value : 0.6349
##              Prevalence : 0.6587
##          Detection Rate : 0.5855
##    Detection Prevalence : 0.7993
##       Balanced Accuracy : 0.6310
##
##        'Positive' Class : 0
##
```

The accuracy is equal to 71%.

4.1.3) Conclusion

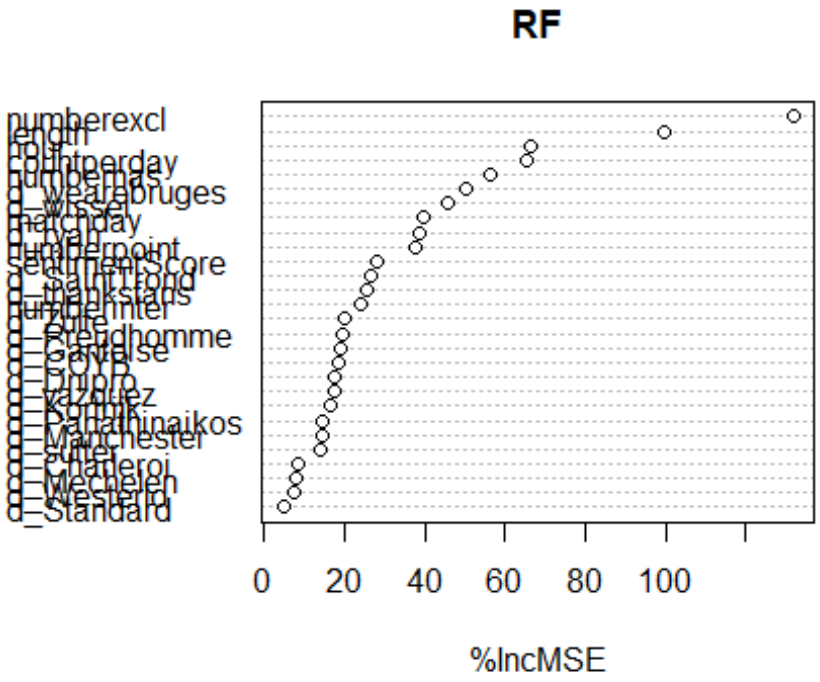
In conclusion, we can notice that the two performance model are equally performer. Moreover, it seems that the same variables drives online engagement for both the likes and the comments.

According to the model, to drive engagement, the posts should not be longer than 200 characters. It also seems that the ideal number of exclamation points is either 2 or 3.

4.2) Regression

4.2.1) Like

To understand the most important variables, we used the %IncMSE graph. This graph represent the increase in Mean Square Error if a specific variable is removed.



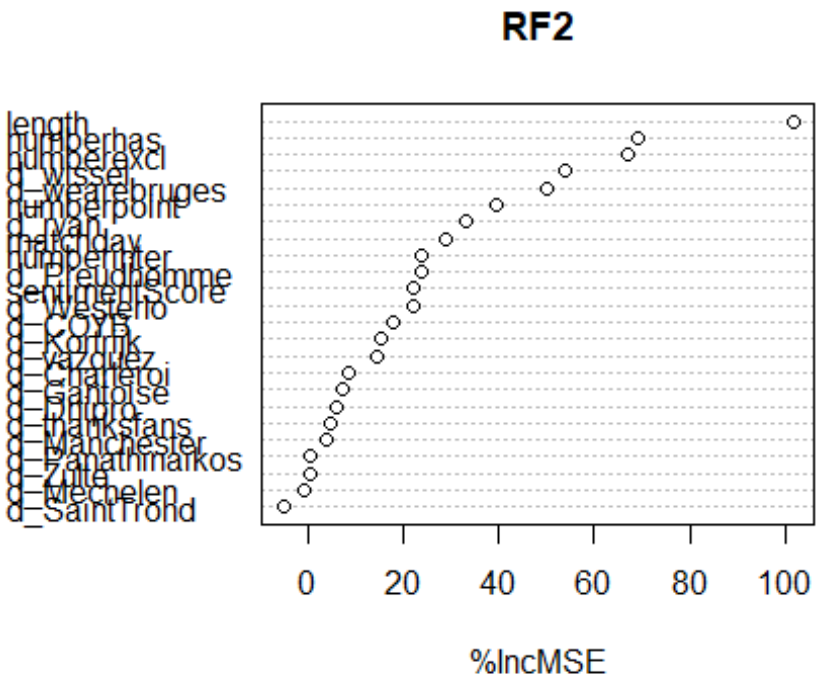
Again two variables looks to be particularly important: the number of exclamation point and the length of the post followed by the hour of the day and the countperday variable.

As you can see below, the model is explaining 32.2% of the total variance.

```
##
## Call:
##  randomForest(formula = like_countLOG ~ ., data = train, importance =
TRUE,      ntree = 1000, mtry = 5)
##                Type of random forest: regression
##                      Number of trees: 1000
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 0.7837909
##                     % Var explained: 32.23
```

4.2.2) Comment

The same actions have been performed for the comments.



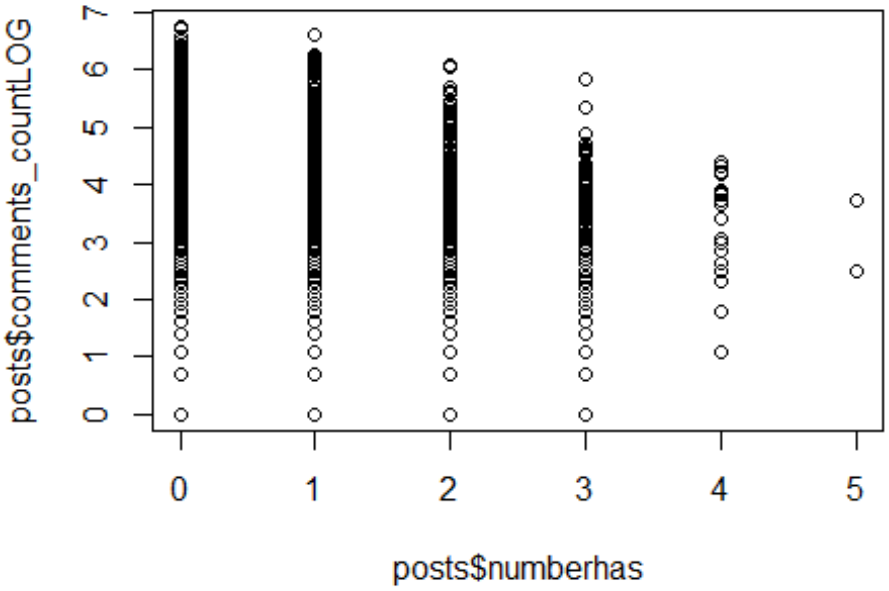
The length of the post is again at first position. Even if the number of exclamation point is important for the like, it looks like the number of hastags used is also important for the prediction of the number of comments.

The percentage of the variance explained is lower for the comments (16.57%)

```
##
## Call:
##  randomForest(formula = comments_countLOG ~ ., data = train2,
importance = TRUE, ntree = 1000, mtry = 5)
##                Type of random forest: regression
##                      Number of trees: 1000
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 1.18858
##                     % Var explained: 16.57
```

4.2.3) Conclusion

For the regression modelling, the model for the likes seems to give better prediction than the model for the comments. The number of hashtag used seems to be really important to predict the number of comments in a post. According to the following graph, we would advise to reduce the number of hashtag if the objective of the club is to get comments



However, in the four models build, the length and the number of exclamation point seem to be particularly important in order to drive engagement.

5) Limitation & future works

5.1) Limitation

The first problem we encountered was the fact that the datasets is imbalanced in the sense that the number of posts with a small number of likes or comments is way higher than the number of posts with a high number of likes or comments. That's why the threshold we selected to determine the class may seems a bit low but it allowed us to rebalance the dataset. However, after some researches in the literature and discussions we discover that this problem is well known in classification modeling. Another way to solve to issue would be to use a SMOTE techniques. According to Chawla et al (2002) is used to solve the problem of imbalanced data when one class in the training set dominates the other.

Secondly, according to Nielsen & Hughes (2015), photos and videos are really important variables to predict the future number of likes or comments. However, this information was not available in the database we received. We are really confident that adding this information to our model could be a nice improvement of ours different models

Third, a spell-check could also been used in this process. However, since we didn't find a nice spelling check dutch dictionary we decided to skip this step.

Finally the ambiguity is a problem in most of the text mining project. In our case, the accuracy of the definitions of several variables is not perfect, mainly for the dummies concerning the team faced. It was quite challenging to find the perfect text pattern so that we know which team Club Bruges faced.

5.2) Future works

We was thinking to integrate a Shiny App to this project. The idea is to provide this shiny app to the Community Manager of the Club Bruges so that he can, on the app, write the post he wants to publish and see directly what are the estimated number of likes and comments.

Moreover, it could be really interesting to use these models with the Tweet of Club Bruges. It could allow us to detect if their fans have different behavior if they are on Facebook or on Twitter.

6) Sources

* Nielsen, A.M., Hughes, B.D., (2015) Real-time suggestions for improving engagement of social media posts using machine learning, Technical University of Denmark

*Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, P.,(2002) SMOTE: Synthetics Minority Over-sampling Technique, Journal of Artifical Intelligence Research ,16, 321-357.

* James, G., Witten, D., Hastie, T., Tibshirani, R., (2017), An Introduction to Statistical Learning with Application in R, ed. Springer

* Moro, S., Rita, P., Vala, B. (2016), Predicting social media performance metrics and evaluation of the impact on brand building: A data mining approach, Journal of Business Reasearch

* Vale, L., Social media and sports: driving fan engagement with football clubs on Facebbok