Does Yelp reviews influence user ratings?

sivarome

Sunday, November 22, 2015

Title

Does Yelp reviews influence user ratings?

Introduction

Yelp is a popular website where users share their feedback about their experience with any business. There are thousands of businesses that are currently being rated by millions of customers. Yelp has become a key influencer for customers.

Imagine you visited a restaurant and wants to rate their business. When you login to Yelp you see your friends' reviews. Would you be influenced by your friends' reviews?

In this analysis we use Yelp data to answer following question:
- While rating a business on Yelp, does the user gets influenced by their friends' reviews.

Yelp dataset contains business, user, review, checkin and tip data. For this analysis we use only business, review and user data.

Environment

The analysis is performed on R programming language on Windows PC. This report is generated using the knitr package in RStudio.

- OS: Windows 8.1 64-bit
- R version: 3.1.3
- RStudio version: 0.98.1103

The random seed value is set to "123".

```
set.seed(123)
```

Methods

Loading the data

Source data is loaded into data tables. For this analysis user, business and review data are used.

```
business_data <- stream_in(file("yelp_academic_dataset_business.json"))
user_data <- stream_in(file("yelp_academic_dataset_user.json"))
review_data <- stream_in(file("yelp_academic_dataset_review.json"))
```

Subsetting the data

Only the businesses in Pennsylvania state are used for this analysis.

```
#Select the businesses in Pennsylvania State (PA)
business_ids <- business_data[which(business_data$state=="PA"), business_id]

# get sample reviews, tips & checkin for the selected businesses
business_sample <- business_data[business_id %in% business_ids]
review_sample <- review_data[business_id %in% business_ids]

# get sample users for the seleted business
user_ids <- unique(review_sample$user_id)
user_sample <- user_data[user_id %in% user_ids]
```

Cleaning the data

The puspose of this report is to see if there is any influence of friends' reviews. So it is important to identify the sentiment of each review. Sentiment analysis is applied on the review text to identify the total positive and negative words in the review. The difference between positive and negative counts will give the sentiment score for the review which will be later used in the prediction model.

Date filter is applied to systematically include only the past reviews (for example, if a user is rating a business on 01-Jan-2015, they cannot see the review posted on 05-Jan-2015. such reviews will be ignored).

Here is the final structure of the cleaned data. **stars** is the outcome variable, which represents the actual rating given by the user. Other fields are calculated using the aggregated sentiment scores of the user's friends.

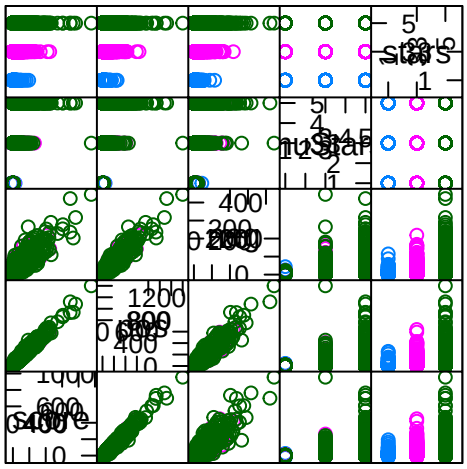
```
str(clean_data)
```

```
## 'data.frame':    16171 obs. of  13 variables:
##  $ nFriends       : int  2 73 27 833 833 833 833 77 77 77 ...
##  $ muFriendFans   : num  1 50 3 36 36 36 36 11.5 11.5 11.5 ...
##  $ muFriendsFunny : num  12 1170 40 854 854 854 854 160 160 160 ...
##  $ muFriendsUseful: num  38 2019 172 3788 3788 ...
##  $ muFriendsCool  : num  8 1225 41 2447 2447 ...
##  $ score          : int  11 0 14 5 5 5 5 6 6 6 ...
##  $ muscore        : num  11 0 7 5 5 5 5 3 3 3 ...
##  $ pos            : int  16 0 18 6 6 6 6 9 9 9 ...
##  $ neg            : int  5 0 4 1 1 1 1 3 3 3 ...
##  $ mupos          : num  16 0 9 6 6 6 6 4.5 4.5 4.5 ...
##  $ muneg          : num  5 0 2 1 1 1 1 1.5 1.5 1.5 ...
##  $ muStars        : num  5 3 5 5 5 5 5 5 5 5 ...
##  $ stars          : Factor w/ 3 levels "1","3","5": 3 2 3 3 2 3 3 3 3 3 ...
```

Exploratory analysis

Here is a pairs plot that shows the correlation of User rating with their friends' reviews (sentiment scores).

```
plotData <-select(clean_data,c(score, pos, neg, muStars, stars))
featurePlot(x=plotData, y=plotData$stars, plot="pairs")
```



Scatter Plot Matrix

These seems to be some correlation between the sentiment scores and the user rating (stars).

Prediction model

Classification models are used for predicting the user rating based on the aggregated sentiment score of their friends. **rpart** (Decision Tree) and **rf** (Random Forest) algorithms are tested as prediction models. The most accurate model is selected based on the results of these tests.

For this analysis the ratings are classified as 1-negative, 3-neutral and 5-positive to show the polarity of the review. These categories are used as prediction outcomes.

Results

Partition

The data is partitioned into training and test datasets with 60:40 ratio based on the outcome variable, stars.

```
set.seed(123)
inTrain <- createDataPartition(clean_data$stars, p=0.6, list=FALSE)

trainData <- clean_data[inTrain,]
testData <- clean_data[-inTrain,]
```

Decision Tree Algorithm

First we apply Decision Tree algorithm to fit the model using the training data set.

```
f <- as.formula("stars ~ nFriends + muStars + score + muscore + pos + neg")

rpartFit <- train(f, data=trainData, method="rpart",
                  preProcess=c("center", "scale"),
                  trControl=trainControl(method="cv", number=5))
```

We use this model to predict the training and test data sets and compare the actual values using the confusion matrix to get the accuracy.

```
rpart_predictions <- predict(rpartFit, newdata=trainData)
rpart_cMatrix_train <- confusionMatrix(rpart_predictions, trainData$stars)
rpart_cMatrix_train$overall[1]
```

```
##  Accuracy
## 0.6845306
```

```
rpart_predictions <- predict(rpartFit, newdata=testData)
rpart_cMatrix <- confusionMatrix(rpart_predictions, testData$stars)
rpart_cMatrix$overall[1]
```

```
##  Accuracy
## 0.6764069
```

This model gives an accuracy of 0.6764069 on the test data.

Random Forest Algorithm (rf)

Next model is based on Random Forest algrithm. Below code applies Random Forest algorithm to fit the model using the training data set.

```
set.seed(123)
f <- as.formula("stars ~ nFriends + muStars + score + muscore + pos + neg")

fitControl <- trainControl(method = "none")
tgrid <- expand.grid(mtry=c(3))
rfFit <- train (f, data = trainData, method = "rf",
                trControl = fitControl,
                tuneGrid = tgrid)
```

We use this model to predict the training and test data sets and compare the actual values using the confusion matrix to get the accuracy.

```
rf_predictions <- predict(rfFit, newdata=trainData)
rf_cMatrix_train <- confusionMatrix(rf_predictions, trainData$stars)
rf_cMatrix_train$overall[1]
```

```
##  Accuracy
## 0.8918891
```

```
rf_predictions <- predict(rfFit, newdata=testData)
rf_cMatrix <- confusionMatrix(rf_predictions, testData$stars)
rf_cMatrix$overall[1]
```

```
##  Accuracy
## 0.6495053
```

This model gives an accuracy of 0.6495053 on the test dataset.

Discussion

Decision tree and Random forest algorithms both resulted in similar accuracy rate with the test data. But for the training dataset, Random forest has much better accuracy. Hence Random forest model is chosen as the better model.

The prediction model resulted in 65 percent accuracy. Although this is not a significant fraction, we can see a considerable correlation between user rating and their friends' reviews.

The correlation is strong for positive reviews compared to the negative reviews. So we can interpret that the positive friend reviews may influence user ratings strongly, however the negative reviews may have less influence.

Following are the confusion matrices for the predictions on training and test datasets. We can observe that the positive reviews have more influence than the negarive reviews.

```
# Confusion matrix for training data
rf_cMatrix_train$table
```

```
##            Reference
## Prediction    1    3    5
##          1  166    8    4
##          3   36 2149  193
##          5   89  719 6339
```

```
# Confusion matrix for test data
rf_cMatrix$table
```

```
##            Reference
## Prediction    1    3    5
##          1   10   25   14
##          3   71  538  690
##          5  113 1354 3653
```