Prediction of influence scores of scientific journals

Saad Elnaem

2024-04-12

Data Description

- **File 1: api_journal11-13-17.csv:**

1.Issn: The International Standard Serial Number of the publication.

2.Journal-name: The name of the scientific journal.

3.Pub_name: The name of the publisher.

4.Is_hybrid: Electronic and printed versions of journal (1); only electronic version of journal (0).

5.category: The category or scientific field of the journal.

6.URL: The web page address of the journal.

- **File 2: api_price11-13-17.csv:**

1.id: Observation id.

2.price: The subscription's price.

3.date_stamp: The date in which in the information was collected.

4.Journal_id: The International Standard Serial Number of the publication.

5.Influence_id: The influence Id.

6.URL: The web page address of the journal.

7.license: Rights for publication, distribution, and use of research.

- **File 3: estimated-article-influence-scores-201.csv:**

1.Journal_name: The name of the scientific journal.

2.issn: The International Standard Serial Number of the publication.

3.Citation_count_sum: The total number of citations of journal.

4.Paper_count_sum: The total number of papers published by the journal.

5.Avg_cites_per_paper: The average number of citations per paper.

6.Proj_ai: The projected article influence. The higher the influence, the better the scientific credibility of the journal.

7.Proj_ai_year: The year of projected article influence.

Loading Libraries

```
library(dplyr)

library(Hmisc)

library(tidymodels)

library(caret)

library(bruceR)

library(randomForest)

library(rpart)
```

Reading the data files

```
journalDB <- read.csv("api_journal11-13-17.csv")

head(journalDB)
```

```
##        issn
## 1 0001-527X
## 2 0002-0397
## 3 0003-0090
## 4 0003-5521
## 5 0004-1254
## 6 0004-282X
##                                                                             journal_name
## 1                                                              Acta Biochimica Polonica
## 2                                                                       Africa Spectrum
## 3                                    Bulletin of the American Museum of Natural History
## 4                                                                       L’anthropologie
## 5 Arhiv Za Higijenu Rada I Toksikologiju-Archives of Industrial Hygiene and Toxicology
## 6                                                         Arquivos De Neuro-Psiquiatria
##                       pub_name is_hybrid                   category  url
## 1    ACTA BIOCHIMICA POLONICA         0 MOLECULAR AND CELL BIOLOGY
## 2                                     0                       NULL
## 3 AMER MUSEUM NATURAL HISTORY         0      ECOLOGY AND EVOLUTION
## 4                    Elsevier         1               Anthropology NULL
## 5                                     0                       NULL
## 6                                     0                       NULL
```

```
priceDB <- read.csv("api_price11-13-17.csv")

head(priceDB)
```

```
##     id price date_stamp journal_id influence_id  url license
## 1 8691  1400 2016-08-11  2051-5960         NULL NULL      NA
## 2 8692  2175 2016-08-11  1758-9193         NULL NULL      NA
## 3 8693  2145 2016-08-11  1476-0711         NULL NULL      NA
## 4 8694  2145 2016-08-11  2047-2994         NULL NULL      NA
## 5 8695  2145 2016-08-11  1744-9081         NULL NULL      NA
## 6 8696  2450 2016-08-11  1480-9222         NULL NULL      NA
```

```
scoresDB <- read.csv("estimated-article-influence-scores-2015.csv")

head(scoresDB)
```

```
##   X                        journal_name      issn citation_count_sum
## 1 0                         3d research 2092-6731                151
## 2 1                     aaps pharmscitech 1530-9932               2208
## 3 2       abstract and applied analysis 1687-0409               3005
## 4 3                 academic psychiatry 1545-7230                537
## 5 4                  academic questions 1936-4709                 40
## 6 5 accreditation and quality assurance 1432-0517                255
##   paper_count_sum avg_cites_per_paper proj_ai proj_ai_year
## 1             106           1.4245283   0.290         2015
## 2             801           2.7565543   0.665         2015
## 3            2923           1.0280534   0.192         2015
## 4             490           1.0959184   0.208         2015
## 5              67           0.5970149   0.097         2015
## 6             331           0.7703927   0.134         2015
```

Join the data files

```
describe(scoresDB$issn)
```

```
## scoresDB$issn
##        n  missing distinct
##     3615        0     3615
##
## lowest : 0001-527X 0002-0397 0003-0090 0005-1098 0007-215X
## highest: 4570-6535 6776-9071 7258-7266 8619-2016 8756-3282
```

```
describe(priceDB$journal_id)
```

```
## priceDB$journal_id
##        n  missing distinct
##     7795        0     5720
##
## lowest : 0001-527X 0003-0090 0003-5521 0005-1098 0008-6223
## highest: 2537-6276 2540-7767 2540-7929 2540-8232 8756-3282
```

```
describe(journalDB$issn)
```

```
## journalDB$issn
##        n  missing distinct
##    13149        0    13149
##
## lowest : 0001-527X 0002-0397 0003-0090 0003-5521 0004-1254
## highest: 9680-5667 9681-2016 9714-1800 9826-4263 9933-2016
```

```
priceDB <- priceDB %>% distinct(journal_id, .keep_all = TRUE)

joinDB <- inner_join(scoresDB, journalDB, by="issn")

joinDB <- inner_join(joinDB, priceDB, by=c("issn"= "journal_id"))
```

Understand the final data after join

```
glimpse(joinDB)
```

```
## Rows: 2,698
## Columns: 19
## $ X                  <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 15, 16, ~
## $ journal_name.x     <chr> "3d research", "aaps pharmscitech", "abstract and ~
## $ issn               <chr> "2092-6731", "1530-9932", "1687-0409", "1545-7230"~
## $ citation_count_sum <dbl> 151, 2208, 3005, 537, 40, 255, 30, 9, 28, 71, 512,~
## $ paper_count_sum    <dbl> 106, 801, 2923, 490, 67, 331, 25, 15, 37, 97, 447,~
## $ avg_cites_per_paper <dbl> 1.4245283, 2.7565543, 1.0280534, 1.0959184, 0.5970~
## $ proj_ai            <dbl> 0.290, 0.665, 0.192, 0.208, 0.097, 0.134, 0.234, 0~
## $ proj_ai_year       <int> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 20~
## $ journal_name.y     <chr> "3d Research", "AAPS PharmSciTech", "Abstract and ~
## $ pub_name           <chr> "Springer", "Springer", "Hindawi Publishing Corpor~
## $ is_hybrid          <int> 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,~
## $ category           <chr> "NULL", "PHARMACOLOGY", "Science", "NULL", "NULL",~
## $ url.x              <chr> "NULL", "", "https://www.hindawi.com/journals/aaa"~
## $ id                 <int> 11162, 11165, 9167, 11166, 11167, 11168, 11169, 13~
## $ price              <dbl> 3000, 3000, 1000, 3000, 3000, 3000, 3000, 0, 0, 0,~
## $ date_stamp         <chr> "2016-08-11", "2016-08-11", "2012-01-01", "2016-08~
## $ influence_id       <chr> "NULL", "NULL", "NULL", "NULL", "NULL", "NULL", "N~
## $ url.y              <chr> "NULL", "NULL", "NULL", "NULL", "NULL", "NULL", "N~
## $ license            <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

Handling the data columns

```
# Delete column x

joinDB <- joinDB[ ,-1]


# The column has one value, 2015, it is not effective in the ML model

joinDB <- select(joinDB, -("proj_ai_year"))
```

```r
# The columns of journal names have 3169 and 3174 unique categorical values, it is very challenging to
joinDB <- select(joinDB, -("journal_name.x"))
joinDB <- select(joinDB, -("journal_name.y"))

# The publisher name has no missing values but with 601 unique categorical values it is very challengin
joinDB <- select(joinDB, -("pub_name"))

# The column contains 545 missing value and cant be imputed.
joinDB <- select(joinDB, -("category"))

# The columns contain 2172 and 927 missing values plus it can't be converted to interpretative values f
joinDB <- select(joinDB, -("url.x"))
joinDB <- select(joinDB, -("url.y"))

# The issn column used as our id column to join the tables, so they will be dropped.
joinDB <- select(joinDB, -("id"))
joinDB <- select(joinDB, -("influence_id"))

# The column contains 917 missing values
joinDB <- select(joinDB, -("date_stamp"))

# Most of the column is missed, 5155.
joinDB <- select(joinDB, -("license"))

# Delete column issn
joinDB <- select(joinDB, -("issn"))

# The column will be converted to factor as it represent categorical values, printed and electronic.
joinDB$is_hybrid <- factor(joinDB$is_hybrid)
```

Data checking

Finding if there is duplicates or missed values

```
any(is.na(joinDB))
```

```
## [1] TRUE
```

```
#  Deleting the missing 4 rows from proj_ai

joinDB <- na.omit(joinDB)

any(is.na(joinDB))
```

```
## [1] FALSE
```

Moving is_hybrid to the end

Move the is_hybrid column to the end to gather all numeric columns together for transforming.

```
joinDB <- joinDB %>% relocate("is_hybrid", .after = "price")

glimpse(joinDB)
```

```
## Rows: 2,694
## Columns: 6
## $ citation_count_sum  <dbl> 151, 2208, 3005, 537, 40, 255, 30, 9, 28, 71, 512,~
## $ paper_count_sum     <dbl> 106, 801, 2923, 490, 67, 331, 25, 15, 37, 97, 447,~
## $ avg_cites_per_paper <dbl> 1.4245283, 2.7565543, 1.0280534, 1.0959184, 0.5970~
## $ proj_ai             <dbl> 0.290, 0.665, 0.192, 0.208, 0.097, 0.134, 0.234, 0~
## $ price               <dbl> 3000, 3000, 1000, 3000, 3000, 3000, 3000, 0, 0, 0,~
## $ is_hybrid           <fct> 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,~
```

Data Transforming - Scaler

Transform all numeric values using min-max method.

```
joinDB[ ,1:5] <- scaler(joinDB[ ,1:5])

head(joinDB)
```

```
##   citation_count_sum paper_count_sum avg_cites_per_paper      proj_ai     price
## 1       0.0003536722    0.0002482732        3.336530e-06 6.792380e-07 0.0070266
## 2       0.0051715779    0.0018761023        6.456402e-06 1.557563e-06 0.0070266
## 3       0.0070383114    0.0068462510        2.407907e-06 4.497024e-07 0.0023422
## 4       0.0012577615    0.0011476781        2.566860e-06 4.871776e-07 0.0070266
## 5       0.0000936880    0.0001569274        1.398328e-06 2.271934e-07 0.0070266
## 6       0.0005972610    0.0007752682        1.804414e-06 3.138548e-07 0.0070266
##   is_hybrid
## 1         1
## 2         0
## 3         0
## 4         1
## 5         1
## 6         1
```

Final data summary

```
dim(joinDB)
```

```
## [1] 2694    6
```

```
glimpse(joinDB)
```

```
## Rows: 2,694
## Columns: 6
## $ citation_count_sum  <dbl> 0.0003536722, 0.0051715779, 0.0070383114, 0.001257~
## $ paper_count_sum     <dbl> 0.0002482732, 0.0018761023, 0.0068462510, 0.001147~
## $ avg_cites_per_paper <dbl> 3.336530e-06, 6.456402e-06, 2.407907e-06, 2.566860~
## $ proj_ai             <dbl> 6.792380e-07, 1.557563e-06, 4.497024e-07, 4.871776~
## $ price               <dbl> 0.0070266, 0.0070266, 0.0023422, 0.0070266, 0.0070~
## $ is_hybrid           <fct> 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,~
```

Split Data

```
set.seed(123)

joinDB_split <- initial_split(joinDB, prop = 0.80, strata = proj_ai)

joinDB_train <- joinDB_split %>% training()

joinDB_test <- joinDB_split %>% testing()
```

Model_01 : Linear Model

```
model_01 <- lm(proj_ai ~ ., data = joinDB_train)

model_01
```

```
##
## Call:
## lm(formula = proj_ai ~ ., data = joinDB_train)
##
## Coefficients:
##         (Intercept)   citation_count_sum      paper_count_sum
##          -3.471e-07            2.437e-05           -1.043e-04
## avg_cites_per_paper                price           is_hybrid1
##           3.311e-01           -2.499e-05            9.012e-08
```

```
predictions_01 <- predict(model_01, joinDB_test)

RMSE_value_01 <- RMSE(joinDB_test$proj_ai, predictions_01)

RMSE_value_01
```

```
## [1] 1.719275e-07
```

Model_02 : logistic Regression using cross validation

- Using 10 folds to train the model.

```
set.seed(123)

training_parameter <- trainControl(method = "cv", number = 10)

model_02 <- train(proj_ai ~ ., data = joinDB_train, family = binomial, method = "glm", trControl = trai

model_02
```

```
## Generalized Linear Model
##
## 2154 samples
##    5 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1939, 1938, 1938, 1939, 1938, 1938, ...
## Resampling results:
##
##   RMSE          Rsquared   MAE
##   2.682205e-06  0.7614669  5.442886e-07
```

```
predictions_02 <- predict(model_02, joinDB_test)

RMSE_value_02 <- RMSE(joinDB_test$proj_ai, predictions_02)

RMSE_value_02
```

```
## [1] 0.00034701
```

Model_03 : Random Forest using hyper-parameter tuning

- Find the best result for different parameters, in this case parameter "mtry" controls the number of variables randomly sampled as candidates at each split when building each tree in the forest. Then apply it to the model training process.

```r
grid_tuning <- expand.grid(mtry = c(2, 5))

model_fit <- train(proj_ai ~ ., data = joinDB_train, method = "rf", tuneGrid = grid_tuning)

model_fit
```

```
## Random Forest
##
## 2154 samples
##    5 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 2154, 2154, 2154, 2154, 2154, 2154, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE          Rsquared   MAE
##   2     4.268777e-07  0.9201609  5.511830e-08
##   5     2.633361e-07  0.9643478  1.848347e-08
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 5.
```

```r
best_mtry <- model_fit$bestTune$mtry

best_mtry
```

```
## [1] 5
```

```r
model_03 <- train(proj_ai ~ ., data = joinDB_train, method = "rf", tuneGrid = expand.grid(mtry = best_m

model_03
```

```
## Random Forest
##
## 2154 samples
##    5 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 2154, 2154, 2154, 2154, 2154, 2154, ...
## Resampling results:
##
##   RMSE          Rsquared   MAE
##   2.562087e-07  0.9663116  1.845702e-08
##
## Tuning parameter 'mtry' was held constant at a value of 5
```

```r
predictions_03 <- predict(model_03, joinDB_test)

RMSE_value_03 <- RMSE(joinDB_test$proj_ai, predictions_03)

RMSE_value_03
```

```
## [1] 2.844273e-08
```

Model_04 : Random Forest without hyper-parameter tuning

```
model_04 <- randomForest(proj_ai ~ ., data = joinDB_train)

model_04
```

```
##
## Call:
##  randomForest(formula = proj_ai ~ ., data = joinDB_train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 4.833721e-13
##                     % Var explained: 77.05
```

```
predictions_04 <- predict(model_04, joinDB_test)

RMSE_value_04 <- RMSE(joinDB_test$proj_ai, predictions_04)

RMSE_value_04
```

```
## [1] 4.857536e-07
```

Model_05 : Decision Tree

```
model_05 <- rpart(proj_ai ~ ., data = joinDB_train)

model_05
```

```
## n= 2154
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 2154 4.537396e-09 1.333868e-06
##    2) avg_cites_per_paper< 1.392463e-05 2091 1.441573e-09 1.161339e-06
##      4) avg_cites_per_paper< 6.229121e-06 1495 2.095075e-10 7.255556e-07
##        8) avg_cites_per_paper< 3.695939e-06 854 2.997000e-11 4.476235e-07 *
##        9) avg_cites_per_paper>=3.695939e-06 641 2.568001e-11 1.095843e-06 *
##      5) avg_cites_per_paper>=6.229121e-06 596 2.359940e-10 2.254454e-06
##       10) avg_cites_per_paper< 9.287384e-06 405 2.782716e-11 1.880579e-06 *
##       11) avg_cites_per_paper>=9.287384e-06 191 3.151343e-11 3.047227e-06 *
##    3) avg_cites_per_paper>=1.392463e-05 63 9.677763e-10 7.060172e-06
##      6) avg_cites_per_paper< 2.54875e-05 49 6.771771e-11 5.451973e-06
##       12) avg_cites_per_paper< 1.899589e-05 37 1.093805e-11 4.869561e-06 *
##       13) avg_cites_per_paper>=1.899589e-05 12 5.531626e-12 7.247743e-06 *
##      7) avg_cites_per_paper>=2.54875e-05 14 3.297784e-10 1.268887e-05 *
```

```r
predictions_05 <- predict(model_04, joinDB_test)

RMSE_value_05 <- RMSE(joinDB_test$proj_ai, predictions_04)

RMSE_value_05
```

```
## [1] 4.857536e-07
```

Analysis and results

- The hyper-parameter tuning improved the Random forest model.
- The result from the Random forest and the Decision tree algorithms is the same in case of no hyper-parameter tuning. this indicate how much they are related. Random forest is an ensemble learning method that builds multiple decision trees.
- Based on the results, the model with best performance is: Random Forest using hyper-parameter tuning, with the lowest RMSE.

```
## RMSE for Linear regression model is: 1.719275e-07

## RMSE for Logistic Regression using cross validation is: 0.00034701

## RMSE for Random Forest using hyper-parameter tuning is: 2.844273e-08

## RMSE for Random Forest without hyper-parameter tuning is: 4.857536e-07

## RMSE for Decision Tree is: 4.857536e-07
```