Cousera - Data Science Capstone Project

Lior Ginzberg

November 11, 2015

Title

This project was developed as part of the Coursera and Johns Hopkins University Data Science certification and more specifically it was prepared as part of the Capstone project which is the final requirement for completing the certification. The project requirement was to download a YELP dataset which consisted of 5 data files, understand the data model, develop a research question, clean and analyze the data and respond with results to the research question. It was critical that the research question would be answerable using the given data and that there will be personal and preferably a wider interest in the analysis results. This challenge was presented to a wider audience as part of round 5 of the YELP challenge..

Introduction

My research question is whether there is a direct correlation between the closure of a restaurant to specific attributes such as 1) Is it good for kids? 2) Are dogs allowed? 3) Is alcohol served? 4) How high the Noise levels is? (Loud and Very Loud) 5) Is smoking allowed inside the restaurant and 6) Is parking only available in the street? These attributes were selected by polling people what would bother them the most about a restaurant from the list of attributes we have in the business file table. The exploratory analysis results indicate that there is a direct correlation between restaurant closure and whether it is good for kids, whether alcohol is served and whether parking is only available in the street. The results of this research question can be of interest to existing restaurant owners and future investors who are planning to open a new restaurant. The analysis results can help them to avoid pitfalls in running their business. The actual analysis will look for additional support to the influence of these attributes in the reviews people provided about these restaurants.

Methods and Data

Step 1: Library preparation (not printed on screen) & data download.

```
## Loading required package: jsonlite
##
## Attaching package: 'jsonlite'
##
## The following object is masked from 'package:utils':
##
##     View
##
## Loading required package: caret
## Loading required package: lattice
## Loading required package: ggplot2
## Loading required package: randomForest
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
## Loading required package: mice
## Loading required package: Rcpp
## mice 2.22 2014-06-10
```

Note: I’ve commented the stream_in() download command for all JSON files that will not be in use during this analysis due to the time it takes to load the files. All these files were downloaded and analyzed in the test environment.

```
business_df <- stream_in(file("C:/Personal/Coursera/Capstone project/Project data/yelp_academic_dataset_business.json"))
```

```
## opening file input connection.
## closing file input connection.
```

```
# checkin_df <- stream_in(file("C:/Personal/Coursera/Capstone project/Project data/yelp_academic_dataset_checkin.json"))
# review_df <- stream_in(file("C:/Personal/Coursera/Capstone project/Project data/yelp_academic_dataset_review.json"))
# tip_df <- stream_in(file("C:/Personal/Coursera/Capstone project/Project data/yelp_academic_dataset_tip.json"))
# user_df <- stream_in(file("C:/Personal/Coursera/Capstone project/Project data/yelp_academic_dataset_user.json"))
```

step 2: Data Preparation
1. Filter the business dataset to include businesses that are defined as Restaurants. This has reduced the dataset
from 61,184 to 21,892 records.
2. Build a dataset that will include only the attributes that are of interest to the research question and convert all attributes to factors.
3. Evaluate the completeness of the data by checking all possible permutations of all 6 predictors + the value NA. I’m
only presenting the first 6 rows of the evaluation due to the big number of permutations. Since there are only 432 entries with all values populated and since this is a relatively small data sample I will keep all records and will handle the NA values at a later point.

```
#See #1 above
rest_only_df <- business_df[grep("[Rr]estaurants",business_df$categories),]

#See #2 above
research_df = data.frame(rest_only_df$business_id)
colnames(research_df)[1] <- c("business_id")
research_df$open <- as.factor(rest_only_df$open)
research_df$good_for_kids <-  as.factor(rest_only_df$attributes$`Good for Kids`)
research_df$dogs_allowed <-   as.factor(rest_only_df$attributes$`Dogs Allowed`)
research_df$alcohol <-        as.factor(rest_only_df$attributes$Alcohol)
research_df$noise_level <-    as.factor(rest_only_df$attributes$`Noise Level`)
research_df$street_parking <- as.factor(rest_only_df$attributes$Parking$street)
research_df$smoking <-        as.factor(rest_only_df$attributes$Smoking)

head(md.pattern(research_df))
```

```
##       business_id open good_for_kids street_parking alcohol noise_level
##   432           1    1             1              1       1           1
##    14           1    1             0              1       1           1
##  1204           1    1             1              1       1           1
##     2           1    1             1              1       0           1
##    42           1    1             1              1       1           0
##     1           1    1             1              0       1           1
##       dogs_allowed smoking
##   432            1       1 0
##    14            1       1 1
##  1204            0       1 1
##     2            1       1 1
##    42            1       1 1
##     1            1       1 1
```

step 3: build train, train_test & test datasets.
1. train dataset: Used for building the prediction model
2. train_test dataset: Used for testing the prediction model without worrying about over fitting
3. test dataset: Used once for “production” run of the final prediction model

```
inTrain = createDataPartition(y= research_df$open  ,p=0.5, list=FALSE)
research_df_train = research_df[inTrain,]
research_df_test = research_df[-inTrain,]

inTest = createDataPartition(y= research_df_test$open  ,p=0.2, list=FALSE)
research_df_train_test = research_df_test[inTest,]
research_df_test_final = research_df_test[-inTest,]
```

step 4: Check for imbalanced data
When the predicted field ($open attribute) has disproportion in its values distribution (# of FALSE and TRUE values) it can severely influence the Random Forest results since the tendency of this prediction model is to take the most common value (TRUE in our case) and predict most of the values base on this value. In order to avoid this issue I’m ensuring that there is an equal presence of the TRUE and FALSE values in the $open field.

```
research_df_train_split <- split(research_df_train, research_df_train$open)
split_test_final <- rbind(research_df_train_split[[1]][1:2167,],
                          research_df_train_split[[2]][1:2167,])
```

step 5: Handle NA values (rfImpute) & run the random forest prediction model

```
split_test_final_imputed <- rfImpute(split_test_final$open ~
                                     split_test_final$good_for_kids+
                                     split_test_final$dogs_allowed+
                                     split_test_final$alcohol+
                                     split_test_final$noise_level+
                                     split_test_final$smoking+
                                     split_test_final$street_parking,
                                     data=split_test_final)
```

```
## ntree      OOB      1      2
##   300:  44.97% 60.59% 29.35%
## ntree      OOB      1      2
##   300:  44.25% 61.61% 26.90%
## ntree      OOB      1      2
##   300:  44.49% 61.33% 27.64%
## ntree      OOB      1      2
##   300:  44.53% 60.41% 28.66%
## ntree      OOB      1      2
##   300:  44.99% 61.42% 28.56%
```

```
research_df_modFit <- randomForest(
      split_test_final_imputed$`split_test_final$open` ~
      split_test_final_imputed$`split_test_final$good_for_kids`+
      split_test_final_imputed$`split_test_final$dogs_allowed`+
      split_test_final_imputed$`split_test_final$alcohol`+
      split_test_final_imputed$`split_test_final$noise_level`+
      split_test_final_imputed$`split_test_final$smoking`+
      split_test_final_imputed$`split_test_final$street_parking`,
      data=split_test_final_imputed, importance = TRUE, ntree = 125)
```

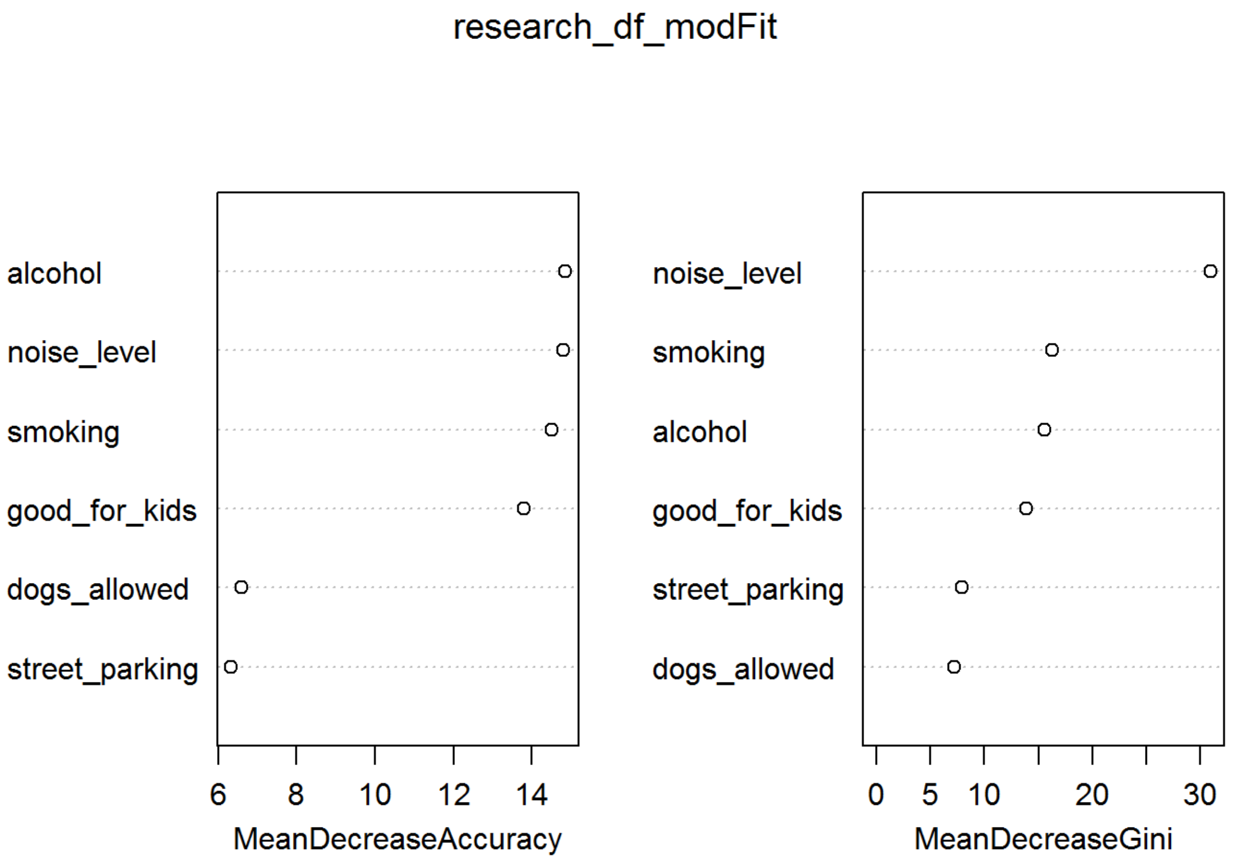
step 6: Visualize the results of the random forest prediction model

As we can see from the below results the prediction model we have designed has a relatively high error rate of 44.39% and more specifically an error rate of 60% in predicting FALSE values (restaurant is closed) and 28% in predicting TRUE values (restaurant is open). In addition we can see that the attributed alcohol, noise level, smoking and good for kids have higher correlation to the prediction results comparing with dogs allowed and street parking.

```
research_df_modFit
```

```
##
## Call:
##  randomForest(formula = split_test_final_imputed$`split_test_final$open` ~      s
plit_test_final_imputed$`split_test_final$good_for_kids` +          split_test_final
_imputed$`split_test_final$dogs_allowed` +          split_test_final_imputed$`split_
test_final$alcohol` +          split_test_final_imputed$`split_test_final$noise_leve
l` +          split_test_final_imputed$`split_test_final$smoking` +          split_t
est_final_imputed$`split_test_final$street_parking`,      data = split_test_final_im
puted, importance = TRUE, ntree = 125)
##                Type of random forest: classification
##                      Number of trees: 125
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 44.39%
## Confusion matrix:
##       FALSE TRUE class.error
## FALSE   865 1302   0.6008306
## TRUE    622 1545   0.2870328
```

```
rownames_backup <- rownames(research_df_modFit$importance)
rownames(research_df_modFit$importance) <-
  c("good_for_kids", "dogs_allowed", "alcohol","noise_level","smoking", "street_park
ing")
varImpPlot(research_df_modFit)
```



```
rownames(research_df_modFit$importance) <- rownames_backup
```

step 7: Run the prediction model against the train test dataset prior to the final run against the real test dataset

We can see from the below results that the accuracy level (how often is the classifier correct?) in the train test dataset is 57%. more specifically we can see that the accurate prediction of closed restaurants is only 20.9% (Pos Pred Value: FALSE predicted as FALSE) which is very low and that the prediction of opened restaurants is 80% (Neg Pred Value: TRUE predicted as TRUE) which is high. This is all leading us to believe that the 6 selected attributes are not good enough predictors to the reason a restaurant is closed and not open.

```
train_test_df <- research_df_train_test[1:4334,]
train_test_pred_results <- predict(research_df_modFit,train_test_df)
train_test_df$predRight <- train_test_pred_results == train_test_df$open
confusionMatrix(table(train_test_pred_results,train_test_df$open))
```

```
## Confusion Matrix and Statistics
## 
## 
## train_test_pred_results FALSE TRUE
##                   FALSE   181  685
##                   TRUE    253 1071
## 
##                Accuracy : 0.5717
##                  95% CI : (0.5507, 0.5925)
##     No Information Rate : 0.8018
##     P-Value [Acc > NIR] : 1
## 
##                   Kappa : 0.0196
##  Mcnemar's Test P-Value : <2e-16
## 
##             Sensitivity : 0.41705
##             Specificity : 0.60991
##          Pos Pred Value : 0.20901
##          Neg Pred Value : 0.80891
##              Prevalence : 0.19817
##          Detection Rate : 0.08265
##    Detection Prevalence : 0.39543
##       Balanced Accuracy : 0.51348
## 
##        'Positive' Class : FALSE
## 
```

Results

step 1: Run the prediction model against the test (production) dataset

Similar to the run of the prediction model against the train test dataset we can see consistent results with the test (production) dataset. The overall level of accuracy is 60%. closed restaurants are predicted as closed only in 22% of the cases and opened restaurants are predicted as opened in 78% of the cases.

```
test_df <- research_df_test_final[1:4334,]
pred_test <- predict(research_df_modFit,test_df)
confusionMatrix(table(pred_test,test_df$open))
```

```
## Confusion Matrix and Statistics
##
##
## pred_test FALSE TRUE
##     FALSE   311 1090
##     TRUE    620 2313
##
##                Accuracy : 0.6054
##                  95% CI : (0.5907, 0.62)
##     No Information Rate : 0.7852
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0116
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.33405
##             Specificity : 0.67969
##          Pos Pred Value : 0.22198
##          Neg Pred Value : 0.78861
##              Prevalence : 0.21481
##          Detection Rate : 0.07176
##    Detection Prevalence : 0.32326
##       Balanced Accuracy : 0.50687
##
##        'Positive' Class : FALSE
##
```

step 2: Research result interpertation
1. The overall accuracy of the prediction model is ~60% which is not accurate enough to provide a conclusive answers if applied against other datasets. 2. The prediction model is better predicting whether a restaurant is open (Neg Pred Value : 0.78861 (78%)) than closed (Pos Pred Value : 0.22198 (22%)) given the selected attributes.
3. The selected attributed can better predict whether a restaurant will stay open vs. whether it will be closed which was not the purpose of this research question.
4. Some bias was introduced into the model by imputing missing data for the selected attributed (predictors) which might have influenced the accuracy of the prediction model. Other imputation models might have been better but much more complex to implement.

Discussion

Even though initially it seemed like the six selected predictors (1. Restaurant is not good for kids 2. Dogs are allowed 3. Alcohol is not served 4. High Noise levels (Loud and Very Loud) 5. Smoking is allowed inside the restaurant and 6. Parking is only available in the street.) would be sufficient to predict whether a restaurant might get closed in the future the Random Forest prediction model showed different results and provided a relatively low accuracy (~60%). Even though we were not able to build a good prediction model using these six attributes we do see that 4 of the attributes (Alcohol, noise level, smoking & good for kids) have more correlation or influence on whether a restaurant will be closed. Since it is critical for a business to know the attributes that might influence on its success I would suggest investing more time in improving this prediction model until optimized results can be achieved. The success of a business is determined by many parameters and can be hard to predict but a good prediction model can be valuable for both business owners and investors.

Thank you for taking the time to look at my research.

Lior Ginzberg