Data Science Capstone Project Report

1. Introduction

The is is the report for the capstone project in the Data Science specialization from Johns Hopkins University via Coursera. The data for the project come from Yelp, which is a business founded in 2004 to "help people find great local businesses like dentists, hair stylists and mechanics." As of the second quarter of 2015, Yelp had a monthly average of 83 million unique visitors who visited Yelp via their mobile device and written more than 83 million reviews. Within Yelp, users rate business with 1 to 5 stars, and also write their own text reviews and provide text tips. There is also a meta-review system from which users can vote on other user's reviews, like other users, be fans and friends.

Yelp collects many different forms of data for a particular business. In this project we analyze five data sets provided for download in JSON format. The data sets have a naming convention characterized by the following words describing the particular nature of the information provided: “business”, “checkin”, “review”, tip”, “user”

Reading the Data

The data are read into the R environment using the jsonlite package. They are structured as data frames with nested data frames and lists as elements. We use the stream_in() function to read the data and then the flatten() function to flatten nested data frames.

We now have five data sets with the following names and dimensions:

- business, 61184 observations of 105 variables
- checkin, 45166 observations of 170 variables
- review, 1569264 observations of 10 variables
- tip, 495107 observations of 6 variables
- user, 366715 observations of 23 variables

Data Description

business, holds business records in the form 'type': 'business', 'business_id': (encrypted business id), 'name': (business name), 'neighborhoods': [(hood names)], 'full_address': (localized address), 'city': (city), 'state': (state), 'latitude': latitude, 'longitude': longitude, 'stars': (star rating, rounded to half-stars), 'review_count': review count, 'categories': [(localized category names)] 'open': True / False ... etc.

checkin, holds checkin information

review, holds business reviews information in the form 'type': 'review', 'business_id': (encrypted business id), 'user_id': (encrypted user id), 'stars': (star rating, rounded to half-stars), 'text': (review text), 'date': (date, formatted like '2012-03-14'), 'votes': {(vote type): (count)}

tip: holds users tip information in the form 'type': ''tip', 'business_id': (encrypted business id), 'text': (tip text), 'likes': {(user likes): (count)} date': (date, formatted like '2012-03-14'), 'user_id': (encrypted user id)

user: holds user information ’type’: ’user’, ’user_id’: (encrypted user id), ’name’: (first name), ’review_count’: (review count), ’average_stars’: (floating point average, like 4.31), ’votes’: {(vote type): (count)}

We follow some data wrangling steps to get the data in more suitable format for our analysis. The data sets are transformed to data.table format. We use the libraries: data.table, dplyr, stringr, lubridate. The steps performed are: * drop from the data sets the variable 'type' as it provides no information. * covert all data frames to data.table objects in R. * simplify all variable names in the data sets to make them more readable * flatten some variables, that are lists to vectors * reorder columns * download US postal codes as a csv and identify based on that the non US locations

We will not use the checkin dataset in our analysis.

Primary Question Asked

In the context of this assignment we will deal with the business rating as reported in the business dataset by the variable "stars". Specifically we will try to build a predictive model for rating businesses based on data from the other datasets provided like "reviews" , "tips" and "users". We will use numerical data as given, construct new features, and also extract new information from the text data provided. The model will be trained and evaluated for the subset of businesses in Edinburgh city, in order to scale down the amount of data processed to a coherent dataset with reviews referring to a common geographical point.

We therefore isolate the observations related to Edinburgh from all our data sets by filtering related observations and we work further with the new data sets to extract features.

2. Methods and Data

Features manipulation and construction

We now work on four data sets refering to Edinburgh businesses, reviews for them, tips for them, and users referring to them in reviews and tips.

We perform several processing steps further in our data sets. Namely,

- We focus only on businesses characterized as 'open', so we remove 'closed' entries from the business data set.
- We drop meaningless variables like, full address, city and state as we are focusing on Edinburgh.
- In the reviews dataset we construct a new variable called 'votes' that is the sum of all votes categories (funny, useful and cool).
- We manipulate dates where present (with lubridate package) to bring them into the format: year/month/day.
- For the variable 'yelping since', the day is missing so we use the first day of each month as the day the user registered (makes no importance). The values are then transformed into a duration in years.
- Time variables yelping_since are transformed to number of years the user is yelping (duration).
- We consolidate all compliments in Edinburgh users, into one column and replace NAs with 0.
- We transform categories columns to a numeric vector with the number of the categories a business falls in.
- We reset variables names to reflect our changes and reorder the columns.

We have to group data together into a common data frame of features. For this we have to merge reviews texts that refer to the same business_id and also merge tips text that corresponds to the same business. We further use the mean value for deriving a single number from multiple observations referring to the same business. There is a number of businesses that have now observations; that means the observations are of value 0.

We thus, construct two datsets named dat1 and dat2, that have a number of variables and dat1 holds the reviews text and dat2 holds the tips text.

Extract Numerical Features from Text

The next step will be to extract numerical features from text fields and derive a final dataset in a form suitable for putting it through a prediction algorithm. We use the R **tm** package to do text processing. Specifically for the reviews text we remove special characters, punctuation, numbers, we transform all letters to lower and remove English stop words. We then stem the documents, strip the white spaces produced and keep it as a plain text document.

Next we use the **tm.lexicon.GeneralInquirer** library to construct numerical scores for positive and negative occurrences of words and we use the difference score (numerical) to replace the text as feature in our dataset.

For the tips text, we use a Document Term Matrix to measure frequencies of specific words like "good", "great", "nice", "like", "love", "best", "epic", "delicious", "mmm", "wow", "awesome", "cool" and use this frequency as the numerical fetaure to replace the plain text.

We finally join together the dat1 and dat2 data sets and produce a factor for our response variable Stars. The final dataset called *D*, has the following structure:

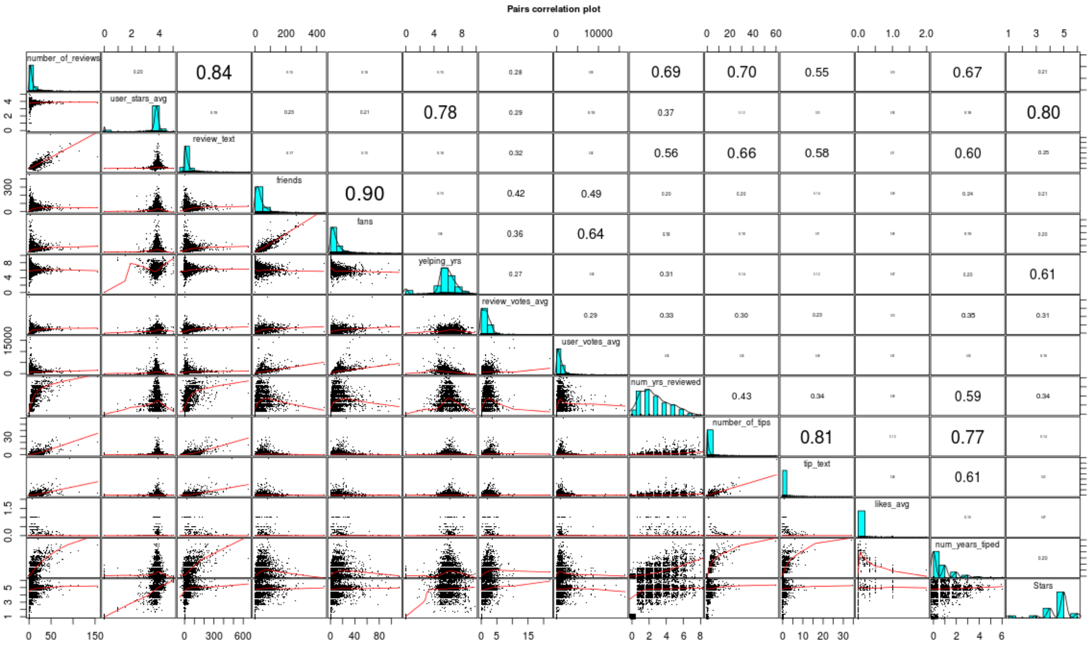
```
Classes 'data.table' and 'data.frame':   2686 obs. of  14 variables:
number_of_reviews:      num  4 5 3 3 3 3 3 4 3 1 ...
user_stars_avg   :              num  3.71 3.95 3.91 3.84 3.85 ...
review_text      :              int  13 2 11 30 4 13 -4 7 19 0 ...
friends          :      num  5.25 45.2 23 15 205 ...
fans             :      num  4.25 5.4 4.67 2.67 54 ...
yelping_yrs      :              num  5.4 6.66 5.73 5.31 5.98 ...
review_votes_avg :      num  0.5 2 1 0.667 3 ...
user_votes_avg   :              num  321 1063 960 392 2343 ...
num_yrs_reviewed :      num  2 4 2 2 3 3 2 2 2 1 ...
number_of_tips   :              num  0 0 0 1 2 3 0 1 1 0 ...
tip_text         :      num  0 0 0 0 0 1 0 1 0 0 ...
likes_avg        :      num  0 0 0 0 0 0 0 0 0 0 ...
num_years_tiped  :      num  0 0 0 1 2 2 0 1 1 0 ...
Stars            :      Factor w/ 6 levels "0","1","2","3",..: 5 5 5 5 4 5 6 4 5 6 ...
```

We will use this dataset to try and train a predictive model and evaluate the effectiveness of our approach.

Exploratory Analysis

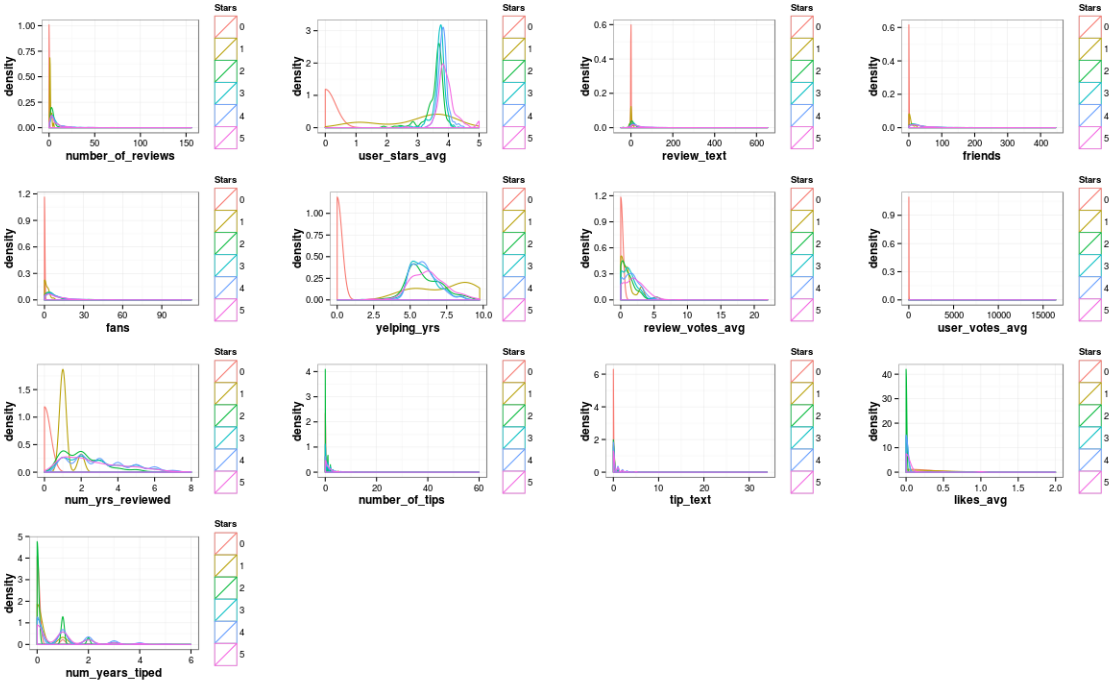
We have now a dataset with 2686 observations and 13 predictors. We will do some useful plots to get a grasp of the quality of our data and predictors.

We use the *psych* and *ggplot2* packages to do at first a correlation plot, shown next.



We see a high correlation between fans and friends predictors, but we will keep all predictors with correlation about so and less. The distributions of the predictors are also not very promising and this is expected because of the nature of the data.

Let's see the density plots of the predictors per response class.



We see that not all predictors have significant variations, so not all of them are promising for separating our classes. Predictors like, *number_of_tips*, *num_years_tiped*,*tip_text*, *likes_average* suffer from very high skewness because of the many 0 values introduced.

We could argue that we could do more value cleansing, deleting values about certain value levels, remove zero values and apply transformations to cope with skewness. However the data do not show to have some particular structure that would be promising for applying a predictive model and is expected as we would like to predict based on arbitrary human input about differing businesses.

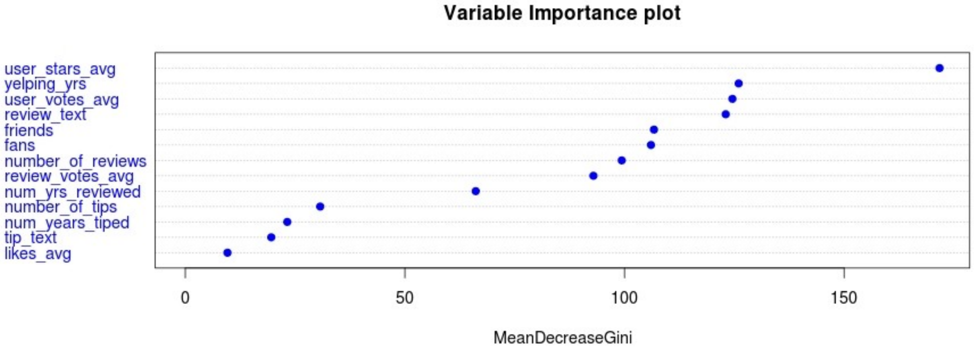
3. Results

We proceed to apply our predictive model and see what kind of accuracy we can get from this. We preprecess the data by centering by mean and scaling the data.

We train a random forest model using the package **randomForest** and we are able to get an accuracy of over 63% in our predictions which is not bad for this datset.

```
     Accuracy          Kappa        AccuracyLower    AccuracyUpper   AccuracyNull
    0.6358209      0.2403310      0.5981032          0.6723362      0.8820896
AccuracyPValue   McnemarPValue
    1.0000000          NaN
```

The variable importance plot agrees with the visual conclusions about our predictors.



4. Discussion

We used several feature transformations in the predictive process like log scale transformations of variables, different scoring functions and a number of predictors including rpart, gbm and ctree. There were minor accuracy differences from the described above so we stay with the random forest model. Possible generation of more features including more advanced techniques in extracting features from text will further improve accuracy and strengthen the model stability.

Please refer to https://github.com/mintzopg/CourseraDS_capstone

for code chunks and figures.