Nathan Lin, Andrew Ton, Mansoor Syed
Speed Dating Project (DS 4559)
10/31/2016

Data Log

Cleaning

1. Imported data with relative path in a project
2. Replaced blank strings with NA
3. Viewed and removed attributes with more than 3,000 missing values
    a. [1] "undergra" "mn_sat" "tuition" "income" "expnum" "attr5_1" "sinc5_1" "intel5_1" "fun5_1" "amb5_1"
    b. [11] "attr1_s" "sinc1_s" "intel1_s" "fun1_s" "amb1_s" "shar1_s" "attr3_s" "sinc3_s" "intel3_s" "fun3_s"
    c. [21] "amb3_s" "attr7_2" "sinc7_2" "intel7_2" "fun7_2" "amb7_2" "shar7_2" "attr5_2" "sinc5_2" "intel5_2"
    d. [31] "fun5_2" "amb5_2" "you_call" "them_cal" "date_3" "numdat_3" "num_in_3" "attr1_3" "sinc1_3" "intel1_3"
    e. [41] "fun1_3" "amb1_3" "shar1_3" "attr7_3" "sinc7_3" "intel7_3" "fun7_3" "amb7_3" "shar7_3" "attr4_3"
    f. [51] "sinc4_3" "intel4_3" "fun4_3" "amb4_3" "shar4_3" "attr2_3" "sinc2_3" "intel2_3" "fun2_3" "amb2_3"
    g. [61] "shar2_3" "attr3_3" "sinc3_3" "intel3_3" "fun3_3" "amb3_3" "attr5_3" "sinc5_3" "intel5_3" "fun5_3"
    h. [71] "amb5_3"
4. Removed additional attributes that did not appear useful in generating models
    a. Condtn (categorical variable for the size of the round), position, positin1 (order matters more than table number), undergrd, field (field_cd categorizes similar fields), career (similar to field_cd), zipcode, goal, date, go_out
5. Created different_scale to attempt to scale the preferences ranked from 1 to 10. Upon viewing the data, appeared that the results were already re-scaled (given the decimal points and sums to 100)
6. Created dataframe/subset date2 which grouped relevant/similar attributes between participants and partners, while removing most of the post-date follow-ups (anything with _2). We chose to perform this since we were more concerned with factors contributing to date performance that were measured during the date or within very close proximity. Additionally, up to ¼ of the observations within these attributes were NA. Attributes removed were as follows:
    a. [1] "attr4_2" "sinc4_2" "intel4_2" "fun4_2" "amb4_2" "shar4_2" "attr2_2"
    b. [8] "sinc2_2" "intel2_2" "fun2_2" "amb2_2" "shar2_2" "attr3_2" "sinc3_2"
    c. [15] "intel3_2" "fun3_2" "amb3_2"
7. Created dataframe date3 which removed observations/rows with more than 20% of the data missing (with ‘manyNAs’)
    a. Removed attributes with suffix 4_1 because of the prevalence of NAs

Data Exploration and Insight/Graph Generation

8. Viewed correlations greater than 0.75
    a. Naturally, there was a correlation between museums and art
9. Created a subset with a collection of rating factors (corrrelation_subset) and visualized correlation with the corrplot library (this library adds some colors and gives a more nuanced view of correlation)
    a. This allowed us to gain a better understanding of the data and the potential "tradeoffs" the participants are making when indicating their preferences (negative correlation indicative of opportunity cost)
10. Created dataframes 'male' and 'female' (subsetted from date3 by 'gender') to begin exploring preferences between males and females
11. **The following relationships were explored** (the process for all these relationships was similar, and is detailed below each relationship)
    a. Difference in preferences between males and females for the six measured attributes (Attractiveness, Sincerity, Intelligence, Fun, Ambition, and Shared Interests); compared the values between males and females for 'attr1_1'
        i. Removed duplicate values with 'sqldf' to create 'means1'
        ii. Aggregated 'means1' by gender
        iii. Subsetted the aggregate data set by the desired attributes
        iv. Renamed the desired attributes to eliminate confusion in the graphs
        v. Created a diff table by subtracting the '1' values from the '0' values
        vi. Melted/transposed the data to prepare for graphing/visualization
        vii. Used 'ggplot' to plot the resulting bar graph
    b. Perceptions of the opposite sex for the six attributes (what males think females want and vice versa); compared the values for males' 'attr2_1' (what individuals think the opposite sex wants) vs. females' 'attr1_1' and vice versa; *note, the below was performed twice*
        i. Removed duplicate values with 'sqldf' to create 'means2m'
        ii. Aggregated 'means2m' by gender
        iii. Subsetted the aggregate data set by the desired attributes ('attr2_1')
        iv. Subsetted again to retain only males
        v. Renamed the desired attributes to eliminate confusion in the graphs
        vi. Performed the above procedure for females with 'attr1_1' and data frame 'means2f'
        vii. Used 'rbind' to stack the 2 generated dataframes, 'means2m' and 'means2f', to create 'means2c'
        viii. Created a diff table by subtracting the '1' values from the '0' values
        ix. Melted/transposed the data to prepare for graphing/visualization
        x. Used 'ggplot' to plot the resulting bar graph
    c. Individuals who marked 100 for desired attractiveness
        i. Subsetted 'date3' by individuals who marked '100' for 'attr1_1'
        ii. Plotted the result on a bar graph to compare male/female responses

iii. Plotted the count of the variable ‘match’ to view the outcome of the individuals who marked 100 for ‘attr1_1’
iv. Plotted the count of ‘dec’ to see if the result was because the individuals who marked 100 had particularly lofty standards

12. Explored additional relationships between the number of matches (‘match’) and the order/timing in which the date occurred in the night (‘order’)
    a. General scatterplot of date order in the night and matches
        i. Created ‘match_time’ by subsetting ‘date3’ by ‘wave’, ‘order’, and ‘match’
        ii. Aggregated based on match
        iii. Created a scatterplot, with grouping based on wave
    b. Which order in each wave had the most number of matches
        i. Used ‘sqldf’ to select the max number of matches and their associated order from each wave and graphed the count of order using ‘ggplot’
    c. Average matches per order for all waves
        i. Used ‘sqldf’ to select the average number of matches for each order and all waves and graphed using ‘ggplot’

Machine Learning Review

1. First, we explored the data and the different binary attributes that can be used as a classifier
    a. “match”: If, after the end of the whole speed-dating process, both individuals in the pair decide with one another.
    b. “dec”: The person’s decision on his or her partner
    c. “dec_o”: For each pair in the data, the partner or other person’s decision on him/her.
2. The central question that we set out to answer was, then, Can one predict a partner’s decision on a person given a few attributes *of* that person?
    a. Attribute “dec_o” is the output and the input attributes are attractiveness, sincerity, intelligence, fun capacity, ambition, and shared interests.
3. First, the data was prepared for randomly sorted and split with 80% of the data designated as training and the remaining 20% as test-data.
    a. All rows with NAs were removed - using the na.omit() command
    b. The binary decision data was converted from an integer type to a factor
4. The next step was to compare the different attributes and see how well of a predictor an attribute is
    a. A LVQ model was created and its variable importance was printed and explored
        i. Attractiveness was by far the most important variable, and ambition was the least important attribute in this model
5. Formally, feature selection was then performed, and the tool of choice lied in recursive feature elimination.
    a. This was chosen since the optimal number of variables to use is easy to obtain from rfe data.

    b. The accuracy of the model on the test data was plotted against the number of variables used to make the model
    c. It is clear from the plot that 2 variables, attractiveness and shared interests, rank highly, or fit according to the data well in some fashion
6. A simple C5.0 tree was then created based on the 6 attributes.
    a. The crosstable showed that false positives and false negatives were of approximately the same value, and the total accuracy was around 74.5%
7. Another C5.0 tree was created with the two most important variables: attractiveness, and shared interests
    a. True to the rfe prediction, the accuracy was greater, at 75.5% accuracy
8. A conditional inference tree was then modeled with the same 2 top attributes.
    a. The accuracy was also 75.5%
9. And finally, a JRIP rule algorithm was executed on the data to achieve a mere 4 rule model with the highest accuracy seen so far 76.0%:
   (attr_o >= 6.5) and (shar_o >= 7) => dec_o=1 (1252.0/262.0)
   (attr_o >= 6.5) and (shar_o >= 5) => dec_o=1 (840.0/315.0)
   (attr_o >= 6) and (shar_o >= 4) and (attr_o >= 8) => dec_o=1 (75.0/33.0)
    => dec_o=0 (3333.0/808.0)

   Number of Rules : 4
10. The Jrip model was run on the 6 attributes as a baseline, and the accuracy was also 76.0%, but with an additional 2 rules.
11. In all these models, the accuracy does not much exceed 75%. This means that at least 25% of a partner's decision cannot be explained well by these 6 attributes.

Text Mining: Observations and Results

1. NYC Trends: "dating"
    a. We analyzed the "dating" keyword in NYC.
        i. Sentiment analysis showed a very negative skew. The overall sum of 2000 tweets was -536.
        ii. Emotion category was mostly uncategorized but the runner up was joy, curiously. Uncategorized was an order of magnitude greater than the emotions. A bayes algorithm was used to classify the emotion and polarity.
        iii. Polarity was split between positive and negative. There were more positive tweets somehow, despite the negative sentiment analysis.
        iv. There was some skew from Taylor Swift - Drake drama. They are possibly dating and they showed up on the wordle. But their effect on the tweets is negligible (from skimming the contents of multiple tweets).
2. Charlottesville Trends
    a. There wasn't enough data to significantly analyze trends. When we used "speed dating," the returned tweets were all bots advertising sex websites. When we used "date," there were mostly just messy data.

3. USA Trends: “Speed Dating”
    a. We analyzed the “Speed dating” keyword for all of Twitter, since that is what returned the most pertinent results.
        i. Sentiment analysis yielded a less negative result for the National Average: -444. This is more positive than the sentiment in NYC, which is unsurprising.

A note on imputation*: Since the data consisted of survey responses (regarding love, which can elicit unique responses from individuals), we made a decision to not impute missing values. As such, our graphics represent statistics where the NAs have been removed, and during our machine learning process, we removed observations which contained NAs.*