# Love in The First Four Minutes: A Second Date Study

Sara Vanovac, Dakota Howard, Luke Eldredge, Jared Korht

## 1 Introduction

"What is most important to people when they are picking a partner?" is a question as old as time. As our society changes, so does the dating culture. In recent years, technological advancements and social media have changed the way in which young, single people socialize and date. We often hear that technology has made us more connected as a global society, but on an individual level people are ever more isolated and lonely. Finding a romantic partner can be a difficult and stressful endeavor. It is no wonder that so many have turned to companies that provide speed dating services, which allow participants to briefly meet and interact with around nine to twenty different potential matches. Meeting this many people in one night may significantly improve the chances of finding the right match. Between 2002 and 2004, Columbia Business School professors Ray Fisman and Sheena conducted an experiment in Speed Dating that they used for their paper titled *"Gender Differences in Mate Selection: Evidence From Speed Dating Experiment."* The data they collected yielded several interesting but disheartening findings that we will investigate and discuss in this project.

Our goal in this project was to build models and determine which algorithms were most successful in predicting the outcomes of speed dating pairings. In evaluating the success of the algorithms, we aim to determine which attributes are most predictive of whether or not a 'match' is made between two participants, and, ultimately, whether or not two participants are likely to be a match. We made use of a wide variety of models, including classification, association learning, numerical estimation, and ensemble learning. We compared the accuracy of different models in predicting Male versus Female decisions. We found that different classification models indeed had different accuracy in predicting decisions made by participants of the opposite gender, underscoring the complexity of the factors that contribute to whether or not a speed date between two individuals leads to a date or not. We also used association learning to explore the relationships between attributes in the dataset, uncovering several interesting correlations that might be helpful to an individual using an online dating site to determine whether or not they might be compatible with another user based on that user's stated preferences.

## 2 The Dataset

The Speed Dating Experiment Dataset we used was collected by professors Ray Fishman and Sheena Lyengar of the Columbia Business School. This is an experimental type of data gathered for research purposes. The data and a description of the variables

are freely available on Kaggle's website at https://www.kaggle.com/ annavictoria/speed-dating-experiment. The dataset contains approximately 8000 instances and 190 attributes. Participants were asked to provide information about themselves, such as their demographics, dating habits, career, and self-perception. They were also asked to review their matches, providing information about their attractiveness, sincerity, intelligence, etc. One interesting aspect of the data is that the there is not only data on each of the participants as individuals, but also on each of the pairings that occurred, such as whether those two people were of the same race or had any shared interests.

The experiment was conducted in 10 waves. For most waves, the participants were given the same instructions, but in some waves, the scoring process was modified, making it necessary to perform some cleaning on the data. A particular issue was that for waves 6-9 scoring was done on a 10 point scale, while for the rest of the waves the scoring was done on a 100 point scale, where the points were distributed across the 6 attributes in question. The participants were asked to submit several surveys: before, during and after the experiment. The 6 main attributes are attractiveness, sincerity, intelligence, fun, ambition, and shared interests. Our work primarily focused on these six to build the classification model, although we also used association learning to explore relationships between matches and other demographic information.

We worked with a mix of nominal, numeric and ordinal attributes. This required a lot of thought and care to be put into their handling, such that we could make use of several different classification algorithms. We describe these efforts in the following section.

# 3 Data Pre-Processing

The dataset that was downloaded from Kaggle website was a result of aggregation of results obtained from 10 waves of speed dating experiments that consisted of 9-21 rounds. Some waves had different questionnaires than others. Thus, several problems were encountered: missing values, different scales, redundant attributes and about 6 different identifier attributes. Furthermore, 190 attributes is far too many, thus we had to perform feature selection. We also had to remove all of the spaces between nominal values and make them all single word.

## 3.1 Feature Selection

We started our data cleaning process by reducing the number of attributes. We explored attributes one by one and we found that several attributes were missing in over 75% of the instances. These were the first attributes that we eliminated. Once these attributes were removed, the remaining attributes had a maximum of 200-300 values missing across 8000 instances. We removed all of the redundant identifier attributes such as subject or partner id. Some attributes were removed upon visualization. For example, we plotted the age distribution and found that everyone was in the same 10-year range, thus, this attribute was removed as it was not very interesting. Likewise, zip code, university, and median income (which was calculated based on the zip code) had very skewed distributions, and were also removed. We also decided to only use the pre- and post-completion surveys, disregarding the several intermediate surveys that were conducted. Attributes such as a does a subject

like partner, decision of partner, decision of subject and match were redundant as well. Thus, we conducted several types of analysis predicting only one of them as a class at a time. Some models looked at what contributed to subject liking the partner, other models looked at what partners liked in subjects, and finally we built models to predict the match. This was the most difficult model to build as 83% of instances were match=No.

### 3.2 Missing Values

As the extent of missing values was not too severe but the reasons behind them were heterogeneous we decided to simply remove them. Most of the missing values were there because users did not answer a question or did not rate one of the dates. This could be for several reasons: the date never showed up or they were not matched with anyone for that round, or they decided to leave before rating their date. Either way, trying to guess what scores they would have given could make data biased. Removing those instances seemed like the best choice given this particular dataset. Also, after we finished with feature selection, only 200-300 instances were missing overall out of 8000. Thus, we decided that the best solution was to simply remove them.

### 3.3 Unbalanced Data

As it turns out, ZeroR revealed that 82% of participants did not have a match by the end of the experiment. As we are trying to build a model that predicts if there will be a match, we run into the issue of an unbalanced data. To remedy this issue we used Weka's ClassBalancer filter to give instances that matched the same weight as instances that did not. We also built separate models for subject and partner. Decision of partner and decision of subject were in fact better balanced about 50-50. Thus, we looked at what contributes to people liking the other persons as well as what contributes to the match.

### 3.4 Dataset Types

As we studied several questions, we worked with several datasets. Note we will refer to the attractiveness, sincerity, intelligence, shared interest, fun, and ambition when used together as "main-six". We used the following 4 dataset versions (which were further subdivided by gender and/or partner/subject) :

- **Dataset 1.** What contributes to a match? This dataset had three versions. The first version contained the "main-six" for the partner and subject, as well as the nominal "Match" attribute. Thus 13 attributes in total. This was the dataset used to build the general model that predicts if a match will happen. We also subdivided this data into subject only and partner only where the class attributes were subject decision and partner decision.

- **Dataset 2.** What attributes are related to each other? This dataset includes 46 attributes. The most of the nominal attributes are self explanatory: race_of_Subject is the the ethnicity of the subject, gender is the gender of the subject, and so on. The attributes that are basic hobbies or activities, like movies and theater, are numeric rankings between 1 and 10 inclusive, where 10 was "I really enjoy this activity" and 1 was "I strongly dislike this activity". The Main Six for both partners and subjects were included as well.

All instances that contained missing values were removed, leaving 2440 from the original dataset; this allowed for analysis of the attributes that were often left blank by participants. While some missing values in a dataset can actually be predictive and yield interesting results, there were simply too many (just less than half) for it to be more interesting than destructive.

- **Dataset 3.** What do women think? This is the subset of the first dataset that focuses only on female subjects, and it has two parts to it: the partner data and subject data. Each of these contains 13 attributes: decision, preference of main-six, and rating of main six.
- **Dataset 4.** What do men think? This is the subset of the first dataset that focuses only on female subjects, and it has two parts to it: the partner data and subject data. Each of these contains 13 attributes: decision, preference of main-six, and rating of main six.

Datasets 3. and 4. were used to test the hypothesis that the differences between the genders could be significant and we might be able to build more predictive models for each gender separately rather than a general model.

# 4 Data Analysis

In the following section we will describe the analysis we conducted on the aforementioned datasets.

## 4.1 Association Learning

Unsupervised learning offered us a glimpse into the more subtle relationships between the attributes in the dataset. We utilized the Apriori algorithm to perform association learning. Apriori keeps track of item sets that appear a sufficient number of times in the training data (a value known as 'support') and then uses them to build rules, keeping those with a high enough level of confidence (the ratio of correctly predicted to incorrectly predicted instances within relevant item sets). The rules built by the Apriori algorithm could be useful for determine what attributes are related to one another, predicting what a new partner will enjoy based on their other attributes, and even if a person is the gender they claim to be.

One limitation of the Apriori algorithm is that it cannot handle numerical attributes. Because of this, we had to discretize each of the numerical attributes. Numerical values of attributes, such as "opinion of partner," were placed into five bins. A preliminary run of Apriori showed that the unbalanced nature of the match attribute (far more speed dates did not lead to a second date than did) led to an overwhelming number of rules that predicted 'match = no'. To remedy this, we applied a resample filter with a 1.0 value for biasToUniformClass to the data. This resampled the data such that there were an even number of instances with match = yes and match = no. This was used as an alternative to ClassRebalancer, which was incompatible with association learning algorithms.

Apriori was now capable of processing our dataset. With a minimum support of .1, 244

instances were necessary to build an item set, and with a minimum confidence of .9, only the rules that were correct 90% of the time were kept. With these parameters, Apriori created the following item sets:

```
Apriori
=======

Minimum support: 0.1 (244 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 153

Size of set of large itemsets L(2): 2805

Size of set of large itemsets L(3): 4939

Size of set of large itemsets L(4): 535

Size of set of large itemsets L(5): 10
```

**Fig 1:** Item sets from AssociationFinal.arff produced by the Apriori algorithm.

More than 60 rules were built from these item sets. The following are a few of the more interesting rules:

- gender=Female; art=(6.4-8.2]; theater=(6.4-8.2] ==> museums=(6.4-8.2] (conf: 0.93 )

Women who indicated that they enjoyed art and theater were also likely to indicate enjoying museums. This could be put to good use on a man's first date – if their partner's profile states that they're into art and theater, then taking them on a date to a nearby museum might be a good idea. This concept is fairly intuitive, as art is typically displayed in museums, and theater is another creative art, and having data to back up this notion has positive implications for planning out dates. Filtering out the instances in Association-Final.arff that were female and enjoyed art and theater (rated >= 6) provided data for a histogram that confirms the validity of this rule.

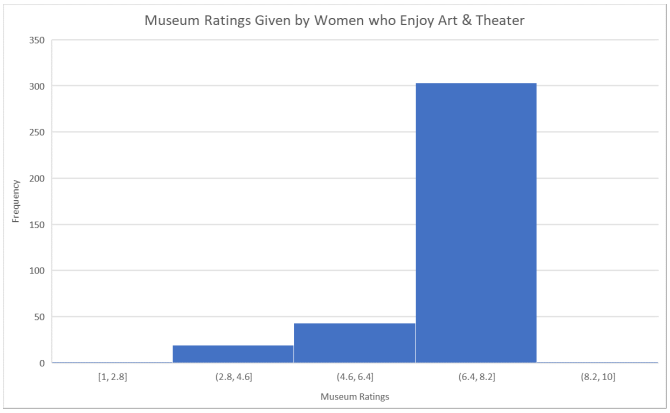


**Fig 2:** The frequency of certain ratings for the attribute "Enjoyment of Museums". Based on a subset of data that included only women who rated "Enjoyment of Art" and "Enjoyment of Theater" above 6.4.

- field = Real Estate; yoga = (-inf-2.8]; ==> gender = Male (conf: 0.94)

According to this rule, individuals who work in real estate but have little-to-no interest in yoga are overwhelmingly likely to be male. On its own, disliking yoga alone is insufficient to predict that someone is Male (about 20.5% of the subjects who gave Yoga a rating between 1 and 3 were Female), and the gender gap in real estate is actually quite small (about 40.9% of realtors were Female). As this rule shows, however, male realtors particularly dislike yoga, making it this a useful rule for predicting gender. Searching online dating profiles for sex-predictive attributes such as this one may be a way to detect men and women who are masquerading as the opposite sex – a growing problem in the industry and in online dating culture.

- Subject opin sinc= (8-inf) Subject opin amb= (8-inf) ==> Subject opin intel = (8-inf) (conf: 0.92)

According to this rule, subjects who perceived their partner as sincere and ambitious also perceived them as highly intelligent. To check the validity of this rule, we needed to ensure that higher scores in general do not boost other scores, but rather that high ratings for sincerity and ambition are uniquely tied to high intelligence. A visualization of the individual relationships between intelligence and sincerity, ambition, and attractiveness (an attribute not included in the rule) helped us explore this.

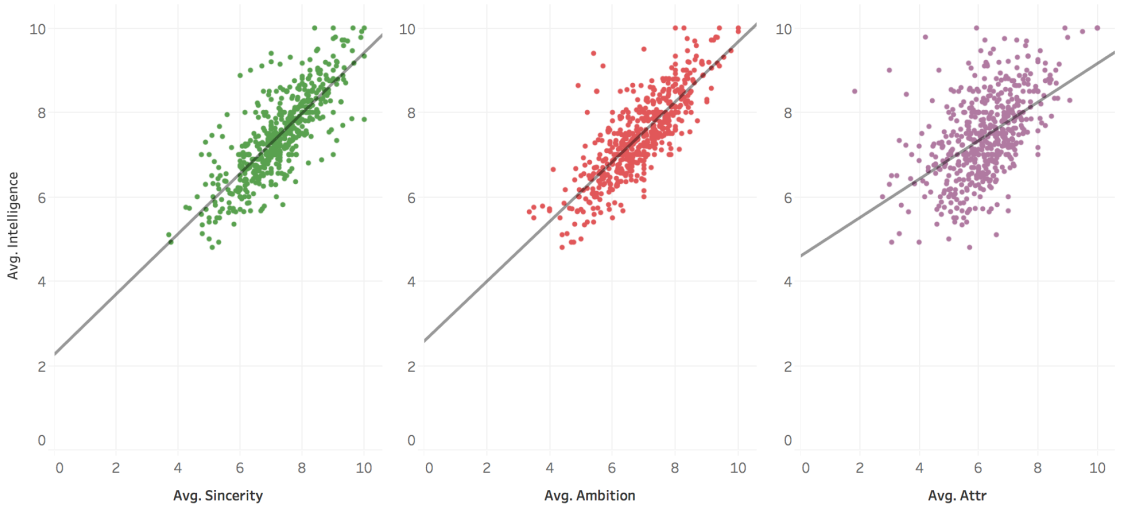


**Fig 3:** Correlations between each Subject's intelligence scores and their sincerity, ambition, and attractiveness scores. Plot created in Tableau.

The relationship between sincerity and intelligence was strong and positive; A Pearson's correlation test in Tableau reported an R squared value of .63. The relationship between sincerity and ambition was also strong and positive, with an R squared value of .64. On the other hand, attractiveness did not correlate as strongly, having an R squared value of .27. This validates the rule, demonstrating that it is not simply the result of higher scores in one area always raising scores in other areas. According to this rule, individuals who are hoping to be perceived as intelligent by their partner could do so by appearing more sincere

and ambitious. Additionally, although attractiveness may contribute more positively to matching, it does not necessarily make a subject appear more intelligent.

## 4.2 Numeric Estimation

Because the majority of the data in our dataset was numeric, we decided to employ numeric estimation to build a model that predicts the match on a scale from 0 to 1 (where 1 means a match occurred). For this purpose we had to convert the match, decision-of-partner, and decision-of-subject attributes to numeric attributes. This then enabled us to utilize Weka's linear regression algorithm to develop a model.

Numeric estimation is a learning strategy similar to classification learning but which works on numerical data and gives a numerical output value. In numeric estimation, a regression function is calculated from the instances in the training set in which each attribute is given a weighting, and the output value is the sum of those weights plus or minus any constants that the model must also include. The regression function is an equation that can be applied to predict the output value for instances in the test set, by simply modifying the values of the individual attribute variables according to their values in that test instance.

We started by using R to build Figures 4. and 5. which show the correlation between the main-six and the decision attributed. We notice immediately that attractiveness and fun show significant correlation, as well as the fact that correlations across genders are different.

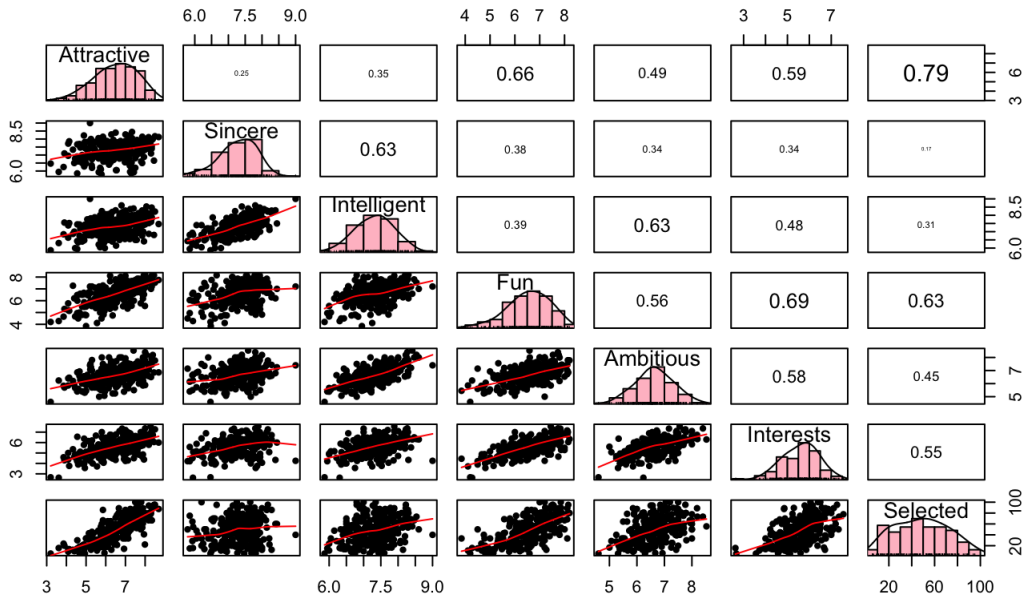


**Fig 4:** Correlations between main-six for males. The plot was developed using R software. The correlation coefficients indicate correlations between single attribute combinations only.

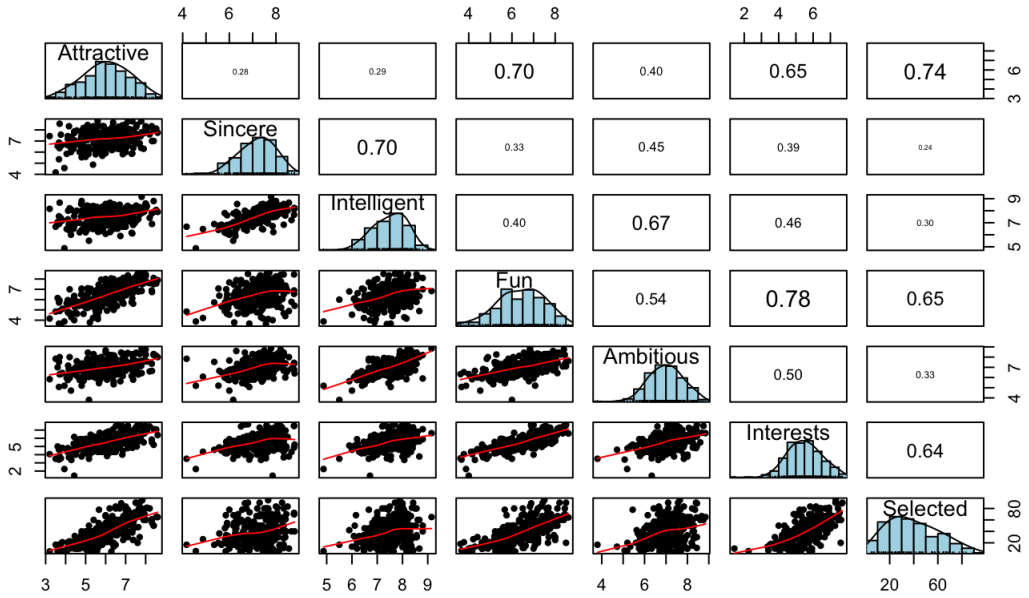


**Fig 5:** Correlations between main-six for females. The plot was developed using R software.The correlation coefficients indicate correlations between single attribute combinations only.

Weka results for Linear Regression:

| Linear Regression Results (percentiles) | | | | | |
|---|---|---|---|---|---|
| | **Dataset 1.** | | **Dataset 3.** | | **Dataset 4.** |
| Algorithm | Subject | Partner | Subject | Partner | Match |
| Linear Regression | 0.488 | 0.541 | 0.538 | 0.507 | 0.5408 |

```
Linear Regression Model

dec=Yes =

     0.0679 * attr +
    -0.0086 * sinc +
     0.0156 * intel +
     0.0395 * fun +
    -0.0223 * amb +
     0.0416 * shar +
    -0.4091
```

**(a)** Linear Regression Equation for Female Subject.

```
Linear Regression Model

dec=Yes =

     0.1137 * attr +
    -0.0257 * sinc +
     0.0429 * fun +
    -0.0237 * amb +
     0.0411 * shar +
    -0.4251
```

**(b)** Linear Regression Equation for Male Subject.

The correlation coefficient is very close across each variation of the data. Also, attractiveness is always the most significant attribute, i.e. it contributes the most positively to

the equation.

## 4.3 Classification Learning

We used an array of different classification learning techniques including Zero-Rule, One-Rule, K-Nearest Neighbor, Naive Bayes, and J48. Each of these techniques had slightly different requirements and enabled us to examine the dataset in different ways. Both Zero-Rule and One-Rule (hereafter, 0R and 1R) are considered "baseline" algorithms because they actually carry out very little computation on the dataset. 0R, for example, builds a model that considers only the class attribute, and no other inputs. It then predicts the majority outcome as seen in the training data. 1R is only slightly more complex in that it predicts the majority outcome for each individual input attribute. 1R and 0R are considered baseline algorithms because they are so simple that if a more advanced algorithm does not have a higher predictive accuracy than 1R and/or 0R, then that algorithm is probably being applied incorrectly or unnecessarily.

The more advanced algorithms we used included J48, K-Nearest Neighbor and Naive Bayes. J48 is a tree-building algorithm similar to 1R in the sense that it finds a sequence of attributes that "best divide" the training data – in other words, the most predictive attribute is found, and then instances are classified accordingly. Next, the attribute that best divides the resulting groups of instances is found, and further splits are made in the same fashion until either there are no more attributes to use or the accuracy may no longer be improved. J48 has several added benefits, in that it can deal with missing values and automatically generate its own rules.

K-Nearest Neighbor classifies instances in the test data based on their similarity (or "nearness") to instances in the training data. For example, the nearest possible neighbor to an instance in the test data would be an instance in the test data for which all attribute values were identical to those of the instance in the training data. The more differences in attributes, the "further" the neighbor is. In K-Nearest Neighbor, the amount of neighbors is variable and can be specified by the individual carrying out the analysis. Generally, the larger the dataset, the more the neighbors that will be used. In our use of the KNN algorithm, we set the number of neighbors to 15. Regardless of the number of neighbors, the final classification of the test instance is the majority classification of all the neighbor instances.

Finally, the Naive Bayes algorithm is a statistical method for classification that determines the probability of each possible classification, using instances as evidence. The rules generated by Naive Bayes read like this: "The probability that the classification is <A> given that the value of <attribute X> is <Y> is: <probability>." In generating these rules, Naive Bayes makes three major assumptions about the attributes: That they are independent, equally predictive, and normally distributed.

| Classification Algorithms Results (percentiles) | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Dataset 1.** | | **Dataset 3.** | | **Dataset 4.** | | |
| Algorithm | Subject | Partner | Subject | Partner | Subject | Partner | Match |
| ZeroR | 83.12 | 83.12 | 60 | 51.37 | 53.57 | 64.65 | 83.12 |
| OneR (1st) | 73.25 | 73.04 | 69.22 | 73. 36 | 73.352 | 74.14 | 49.23 |
| OneR (2nd) | 68.34 | 68.58 | 68.52 | 69.59 | 69.57 | 68.59 | 49.23 |
| N. Bayes | 70.8 | 74.44 | 70.48 | 72.38 | 72.65 | 78.62 | 72.26 |
| IBk (k=15) | 62.8 | 73.43 | 71.96 | 75.12 | 72.0 | 77.42 | 62.77 |
| J48 | 66.7 | 64.81 | 74.12 | 76.93 | 72.80 | 77.72 | 69.58 |

## 4.4 Ensemble Learning

Ensemble learning is a strategy which trains multiple algorithms, comparing and combining their results to achieve a better predictive strength than any of the individual algorithms involved. The ensemble learning algorithm we selected was called Voting, which is used for classification. The Voting algorithm is intuitively appealing: all of the pre-specified sub-models are run, and the classifications made by each model for each instance are compared. Each of the classifications is considered a "vote," and the classification that received the majority of the votes (i.e., more than half of the sub-models gave this classification) is selected as the final result.

For out ensemble learning, we selected the following sub-models: J48, Naive Bayes, and 1R. We chose these three because we had experience working with them in the labs, and because they are all very different, being a tree algorithm, a classification algorithm, and a rule-generating algorithm, respectively. We anticipated that using a diverse selection of algorithms would be beneficial because it's unlikely that combining a bunch of very similar algorithms could offer much improvement. Another algorithm that we considered using was tried the instance-based nearest neighbor algorithm, IBk, although we found that including this in our model reduced its accuracy.

Before running our ensemble learning paradigm, we applied two filtering steps. The first was to use the Class Balancer filter in Weka, which is a component of the pre-processing toolkit. We used this to accommodate for the fact that, in our dataset, there were far more (roughly 9-10x) more non-matches than matches, and we did not want this to skew the results such that the algorithms could simply predict "no" with a high success rate. The Class Balancer filter achieves this by adjusting the weights of the instances in the dataset so that all classes are equally weighted. The second filtering step we used was simply to run our Voting algorithm through a Filtered Classifier, which is done in the classification tab of Weka.

The results of our ensemble learning efforts are given below:

| Ensamble Learning Results (percentiles) | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Dataset 1.** | | **Dataset 3.** | | **Dataset 4.** | | |
| Algorithm | Subject | Partner | Subject | Partner | Subject | Partner | Match |
| Ensemble (J48, OneR, N.Bayes, IBk) | 76.50 | 77.28 | 74.63 | 78.15 | 76.83 | 74.55 | 73.17 |

# 5 Conclusions

Overall, we can compare our algorithms in terms of their ability to predict decisions made by females versus males, further sub-divided into partner versus subject ratings. Our baseline algorithm, ZeroR, had a predictive accuracy between 51.37 and 64.65% accuracy across all four of these possible subdivisions, indicating that there was significant potential for the application of more complex models for more accurate results. According to our other baseline algorithm, OneR, the most predictive attribute was Attractiveness, followed by Fun. Predictions made on the sole basis of these two attributes alone were much more successful than those of ZeroR, improving the accuracy range to 69.22-74.14% (for Attractiveness) and 68.52-69.57% (for Fun).

In general, our algorithms predicted Male decisions more accurately than they did Female decisions, implying that the reasons why men like a partner are more consistent than those for females. Furthermore, predictions made on the basis of partner ratings were more accurate than predictions made based on subject ratings, and this effect appeared magnified somewhat for Male decisions. One interesting result of our analysis was that the most predictive classification algorithm for predicting Female decisions was Nearest Neighbor (subject: 71.96%; partner: 75.12%), whereas for predicting Male decisions, it was Naive Bayes (subject: 72.65%; partner: 78.62%). This underscores the complexity of predicting the results of speed dating pairings, and indicates that women and men making decisions through slightly different processes, even though it was true that for both Males and Females, Attractiveness and Fun were the most predictive attributes. Linear regression picked out Attractiveness and Fun as most predictive attributes as well.

Different from the gender based data, the general dataset where the class attribute was match was not both very unbalanced, i.e. ZeroR accuracy of 83.12% and OneR predicted less than 50% instances correctly. This indicates that unlike predicting single direction decisions, such as those for subject and partner, to predict if a match happened more than one attribute had to be used in combination, i.e. the model is more complex. Naive Bayes performed best as the general model giving the accuracy of 72.26%.

The results of our ensemble learning model further recapitulate the importance of separating Males versus Females in conducting this analysis. It is interesting to consider that, because a large number of our results were linked to differences in gender, a separate study based on (or at least including) same-sex pairings might have very different results in terms of which attributes it finds predictive of successful matching. However, the results show that in fact predicting the match had lower accuracy, or was more difficult than predicting the decisions based on gender.

Finally, while the preferences of both males and females indicated that attributes such as sincerity, intelligence and shared interest are most important, the data shows that in really attractiveness and fun are the attributes that decided whether or not individual wanted to go on a second date. The human behavior that this experiment revealed is that common "heart versus head" battle. While we know that we value virtues that are internal,when looking for a partner on a speed date we go for the external factors such as attractiveness.