CS573 - HW2

Wen-Ling Chi

Email: chi64@purdue.edu

1. Preprocessing

After the preprocessing, it will output a new file **dating.csv**.

(1) Remove the quotation marks from 8316 cells.
(2) Standardized 5707 cells to lower case.
(3) Use label encoding to convert the categorical values in columns gender, race, race o and field to numeric values start from 0.
(4) The mean values of each column of "pref_o_[six attributes]" and "[six attributes]_important".

```
Quotes removed from 8316 cells.
================================================
Standardized 5707 cells to lower case.
================================================
Value assigned for male in column gender: 1.
Value assigned for European/Caucasian-American in column race: 2.
Value assigned for Latino/Hispanic American in column race o: 3.
Value assigned for law in column field: 121.
================================================
Mean of attractive_important: 0.22.
Mean of sincere_important: 0.17.
Mean of intelligence_important: 0.20.
Mean of funny_important: 0.17.
Mean of ambition_important: 0.11.
Mean of shared_interests_important: 0.12.
Mean of pref_o_attractive: 0.22.
Mean of pref_o_sincere: 0.17.
Mean of pref_o_intelligence: 0.20.
Mean of pref_o_funny: 0.17.
Mean of pref_o_ambitious: 0.11.
Mean of pref_o_shared_interests: 0.12.
```

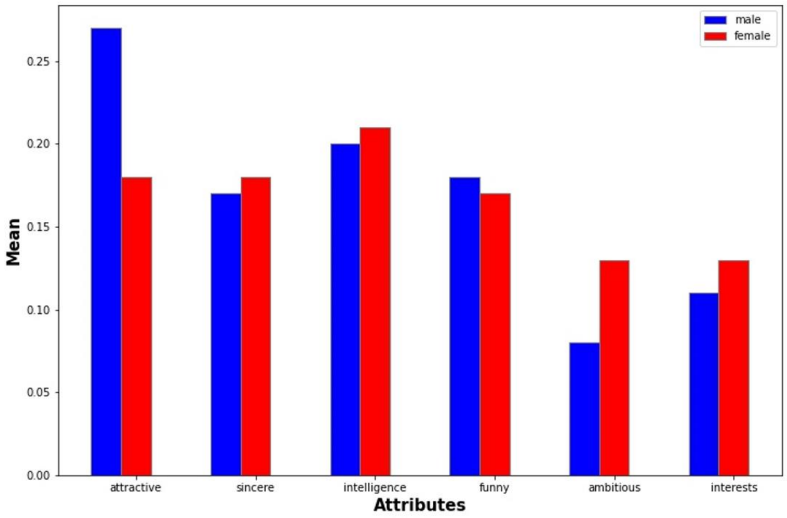
The output of compiling preprocessing.py

2. Visualizing interesting trend in data

(1) Contrast how females and males value the six attributes in their romantic partners differently.

Calculate the mean of the six characteristics of the subset of male and female.

Show a single barplot. The visualization code is in **2_1.py**.



six attributes of females and males

a. What do you observe from this visualization?

Female's ratings for each attribute are relatively average.

But men pay more attention to "attractive", and less attention to "ambition" and "shared_interests".

But between female and male, "sincere", "intelligence", "funny" are given high scores.

b. **What characteristics do males favor in their romantic partners?**

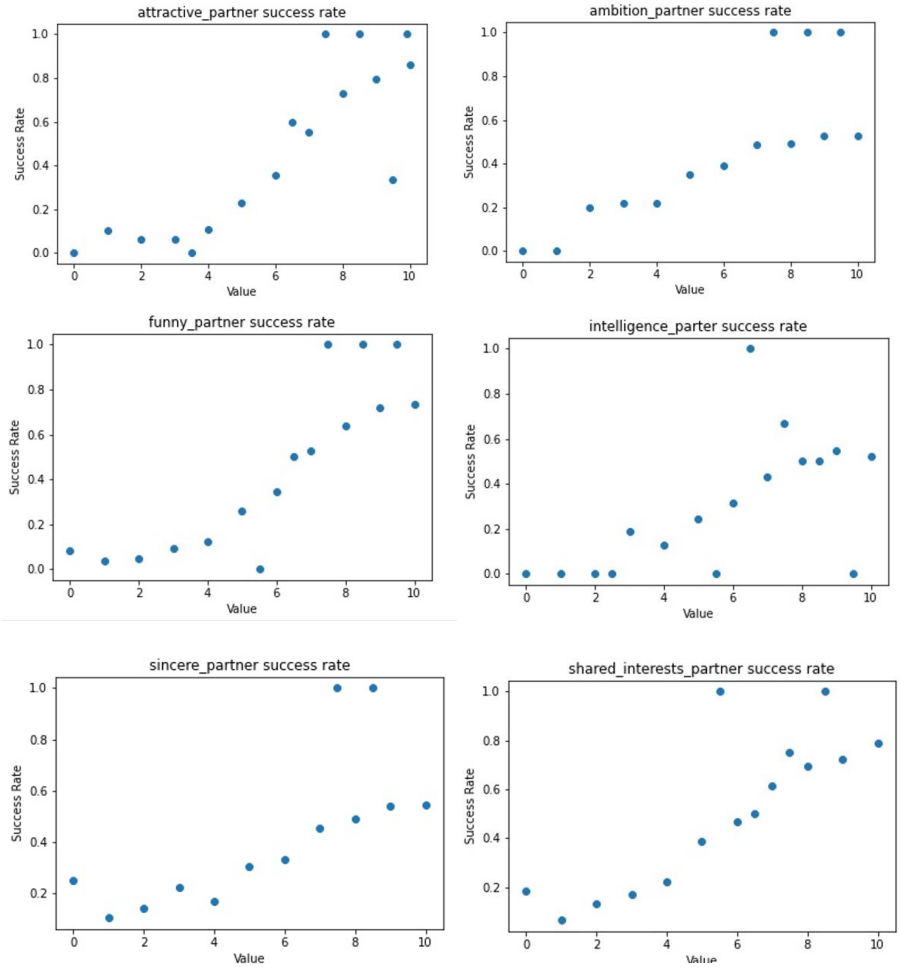
Male thinks "attractive" is more important, but they also give high scores to "sincere", "intelligence", and "funny".

c. How does this differ from what females prefer?

Female thinks "intelligence" is the most important, but they also give high scores to "attractive", "sincere", and "funny".

(2) How a participant's rating to their partner on each of the six attributes relate to how likely he/she will decide to give the partner a second date.

Compute the fraction of participants who decide to give the partner a second date among all participants whose rating of the partner on the chosen attribute (e.g., attractive partner) is the given value (e.g., 10). Show 6 scatter plots. The visualization code is in **2_2.py**.



participant's rating to their partner on each of the six attributes

a. What do you observe from these scatter plots?

Basically, every attribute is positively correlated, which means that the higher the score, the more successful the second date will be.

Attractive, funny, shared_interests: The slopes of these three categories are relatively high, which means that the higher the score, the easier it is to succeed. But there are a few exceptions

Intelligence, ambition, sincere: The slopes of these three categories are relatively low, which means that higher scores may not necessarily lead to success.

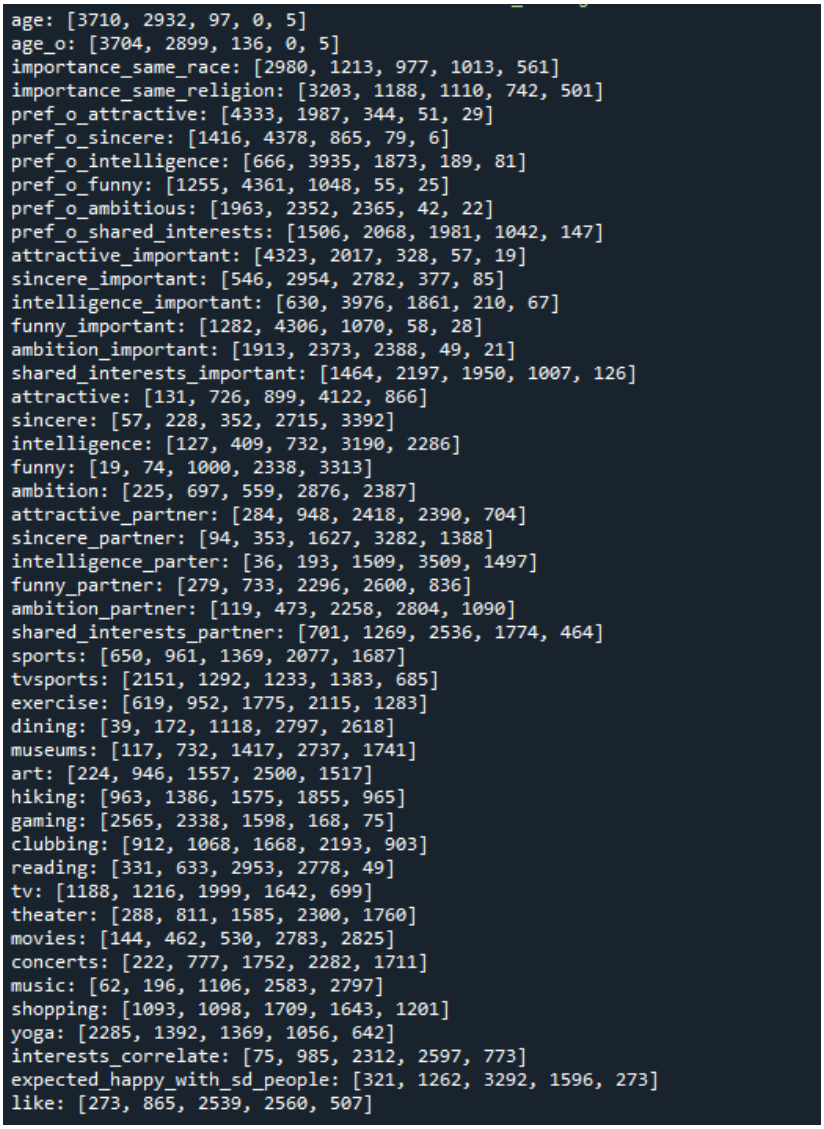
3. Convert continuous attributes to categorical attributes

Discretize all columns in *continuous valued columns* by splitting them into **5** bins of equal-width in the range of values for that column.

According **fieldmeaning.pdf**, there are 47 columns needing to discretize.

The script reads **dating.csv** as input and produces **dating-binned.csv** as output.

This image is the result of running **discretize.py**



4. Training-Test Split

Use the ***sample*** function from ***pandas*** with the parameters initialized as random state = 47, frac = 0.2 to take a random 20% sample from the entire dataset.

Create a new script called **split.py** that takes **dating-binned.csv** as input and outputs **trainingSet.csv** and **testSet.csv**.

- Total has 6744 rows
- trainingSet has 5395 rows (80%)
- testSet has 1349 rows (20%)

5. Implement a Naive Bayes Classifier

(1) Using Naïve Bayes Classifier to evaluate the accuracy of trainingSet.csv and testSet.csv

Follow NBC's formula to create probability tables.

After creating the table, I have used smoothing in order to solve the situation where the probability is zero.

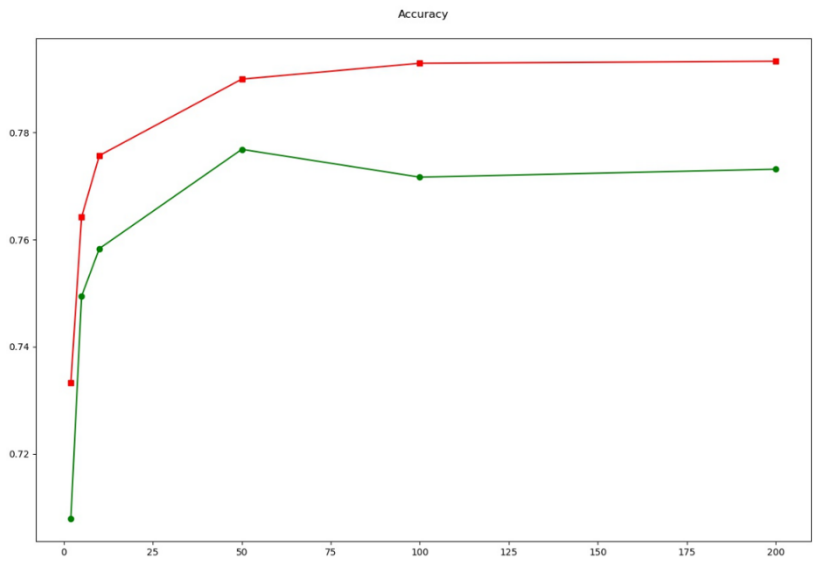
```
Training Accuracy: 0.76
Testing Accuracy: 0.75
```

5-1 output

(2) Examine the effects of varying the number of bins for continuous attributes during the discretization step

When the bin is smaller, the more difficult it is to correctly classify
However, the larger the bin is, the more time will be spent on training the model and the more overfitting problems will be caused.

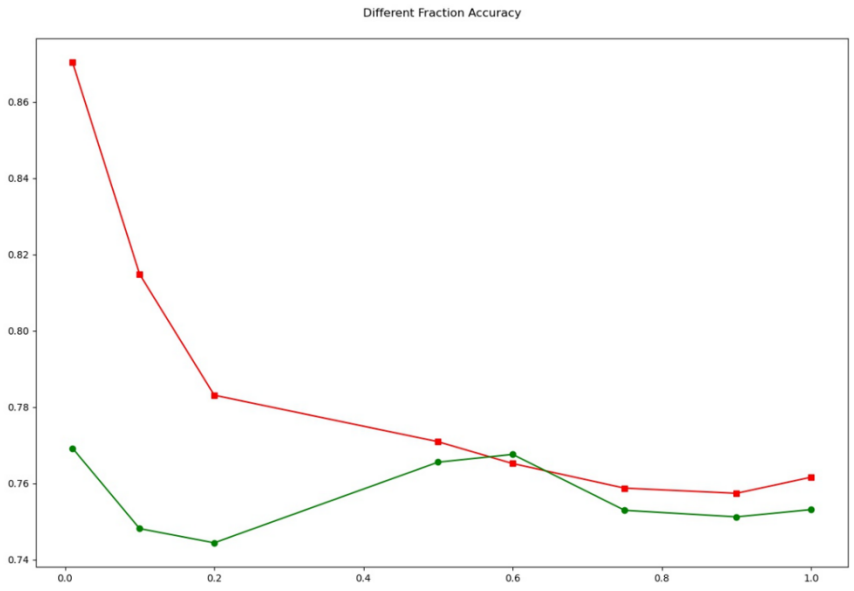
```
Bin size: 2
Training Accuracy: 0.73
Training Accuracy: 0.71
Bin size: 5
Training Accuracy: 0.76
Training Accuracy: 0.75
Bin size: 10
Training Accuracy: 0.78
Training Accuracy: 0.76
Bin size: 50
Training Accuracy: 0.79
Training Accuracy: 0.78
Bin size: 100
Training Accuracy: 0.79
Training Accuracy: 0.77
Bin size: 200
Training Accuracy: 0.79
Training Accuracy: 0.77
```



(3) Plot the learning curve for different fraction.

The proportion of the training data set used for training has a big impact on the accuracy of the validation set. If you use too small a fraction, it can lead to an overfitting situation. That is, even though you can achieve a high accuracy on the training set. You are only learning the pattern on this small data. The accuracy on the test set will be very small. However, if we increase the size of the training sample, it will lead to better model generation that presents good performance on both training and test data sets.

```
fraction = 0.01
Training Accuracy: 0.87
Testing Accuracy: 0.77
fraction = 0.1
Training Accuracy: 0.81
Testing Accuracy: 0.75
fraction = 0.2
Training Accuracy: 0.78
Testing Accuracy: 0.74
fraction = 0.5
Training Accuracy: 0.77
Testing Accuracy: 0.77
fraction = 0.6
Training Accuracy: 0.77
Testing Accuracy: 0.77
fraction = 0.75
Training Accuracy: 0.76
Testing Accuracy: 0.75
fraction = 0.9
Training Accuracy: 0.76
Testing Accuracy: 0.75
fraction = 1
Training Accuracy: 0.76
Testing Accuracy: 0.75
```



The output of different fractions