CS573 HOMEWORK 2

Mohit Gupta

Feb 13, 2019

1 Preprocessing

The code is given in preprocess.py file.

2 Visualizing interesting trends in data

2.1 Observation - Male Female Difference

From Figure 1, we observe that in general attributes like attractiveness, sincerity, intelligence and funny are much more important compared to other qualities like ambition and shared interests. For example, attraction has a mean score of around 0.26 and 0.18 for males and females respectively compared to the mean score of shared interests which is just 0.11 and 0.12 for males and females respectively.

Males prefer attraction to be the most important attribute and has a fraction of 0.26 which is very high even when compared to the second highest attribute of intelligence which has a fractional score of just 0.2. This might mean that males in general need their partners to be very attractive and dont normally care for them to be ambitious which has a fractional score of just 0.09. They also care less if their partners have shared interests since shared interests attribute has a fractional score of 0.11. Males care for sincerity and intelligence with their fractional scores being 0.17 and 0.2 respectively.

On the other hand, females prefer intelligence with a score of 0.21 to be the most important attribute closely followed by attractiveness and sincerity both with scores of 0.18 each. This shows that females might be more thoughtful when selecting their partners as they give more weightage to intelligence attribute and dont have a very large difference between the 3 attributes of intelligence, attractiveness and sincerity. Partners which have a mix of these 3 attributes are likely to appeal more to females. Females also care about their partners being funny which has a score of 0.17. The fractional score for ambitition and shared interests is 0.13 and 0.12 respectively.

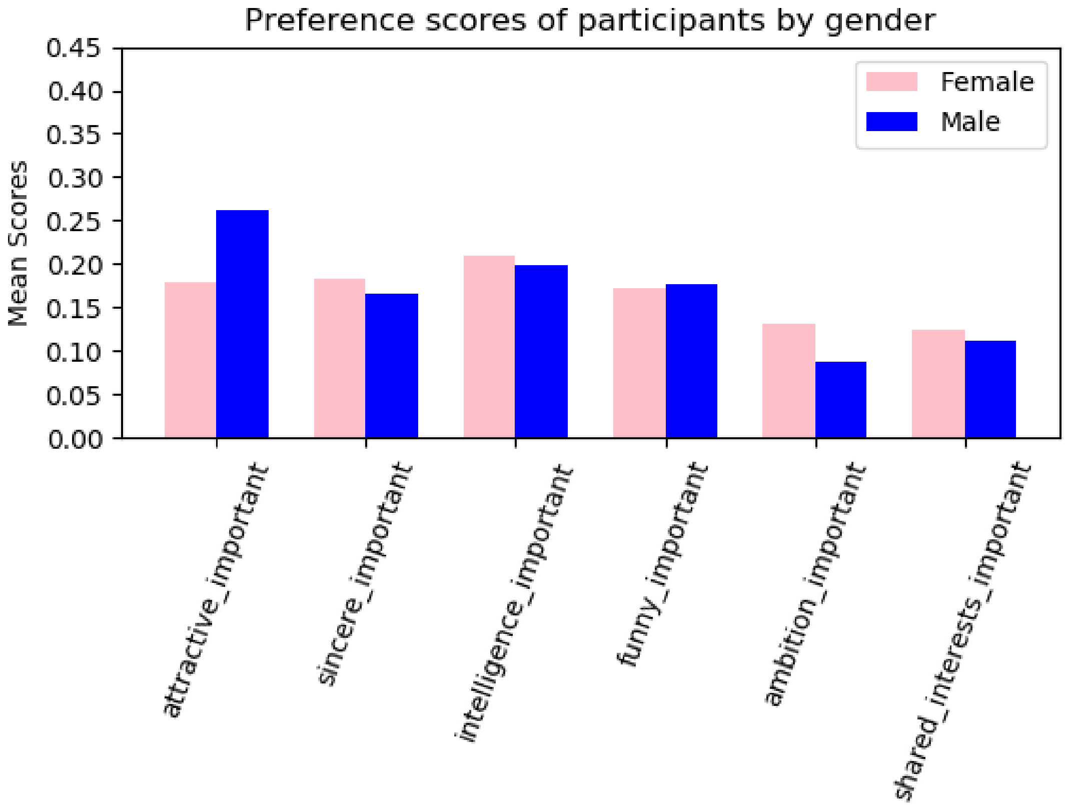

Figure 1:

2.2 Observation from Scatter Plots

From Figure 2, we can observe that the success rate for partners which are rated low i.e. less than 5 is very low i.e. around 0.1. The success rate increases drastically when value for attractive becomes greater than 6 and for success rate for 8 and higher values is around 0.9. There is an outlier for value 9.5 which is just a single data point. In general, the success rate for attractive partners becomes very high once the value becomes greater than 6.

From Figure 3, we can observe that the success rate for ambition is relatively low compared to for example the attribute of attractiveness. For attribute value greater than 6, there is an almost even split of some values having success rate of around 0.5 and some around 1. We observe that the partner having a very high ambitious value is likely to have a higher success rate in around half the

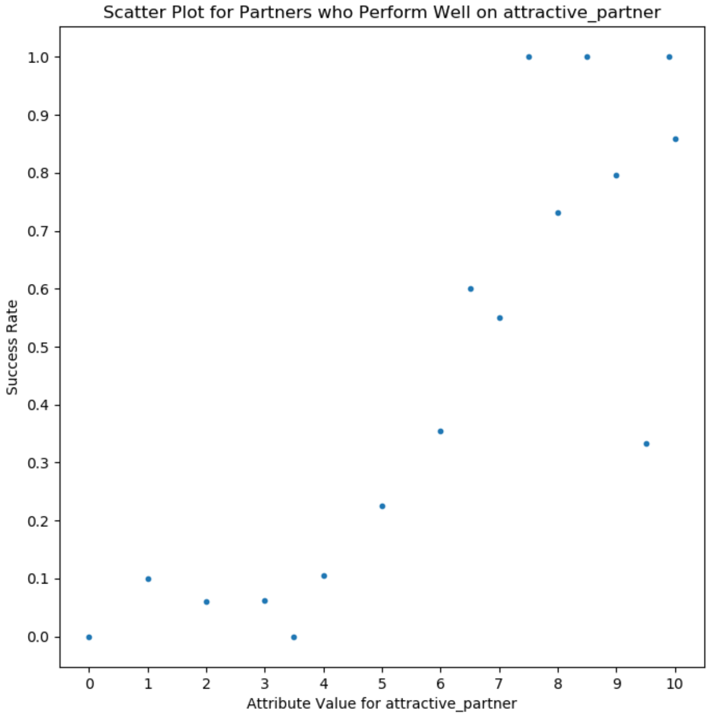

Figure 2:

cases.

From figure 4, we observe that funny partner attribute has an almost flat success rate of 0.1 till the attribute value is less than 5. After that there is an increase in the success rate which becomes around 0.7 or larger once the attribute value reaches 8.

From figure 5, we observe that intelligence partner attribute has a relatively low success rate even for very high attributes values i.e. greater than 7. The success rate hovers around 0.5 which is low. Thus, we find that having a very high intelligence does not guarantee a high success rate.

From figure 6, we observe that shared interests attribute has an almost linear increase in success rate with increase in attribute value starting from around 0.1 for attribute value 1 and goes to around 0.8 for attribute value 10. The attribute has relatively high success rate for high values which is good sign for

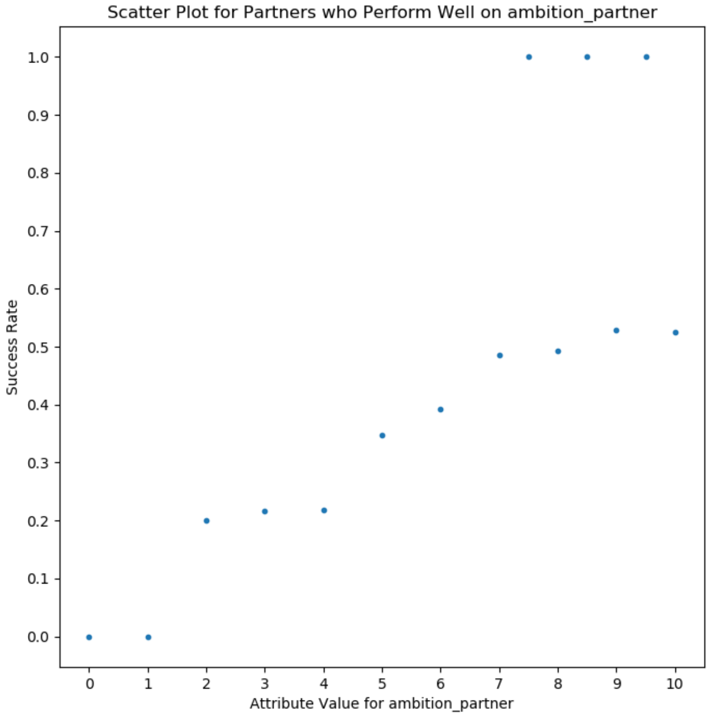

Figure 3:

partners with shared interests.

From figure 7, we observe that sincere attribute has relatively low success rate even for high attribute values of around 7 apart from 2 outliers at 7.5 and 8.5. The success rate hovers around 0.5 for values 8, 9 and 10. This shows that having a very high value of sincerity alone is likely to have a lower success rate compared to other attributes for example attractiveness.

3 Convert continuous attributes to categorical attributes

The code is in discretize.py file. I have a do_discretize method which has the core logic for discretization.

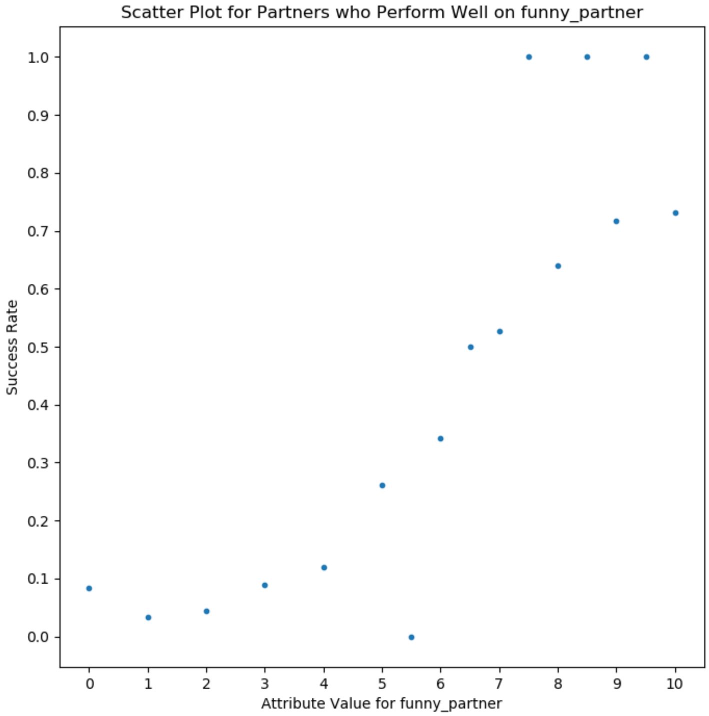

Figure 4:

4 Training-Test Split

The corresponding file is split.py.

5 Implement a Naive Bayes Classifier

5.1 Use all the attributes and all training examples in trainingSet.csv to train the NBC.

The core logic is in nbc(t_frac) method. For bin size 5 and t_frac value of 1, we have training accuracy score of around 0.78 and test accuracy score of 0.75.

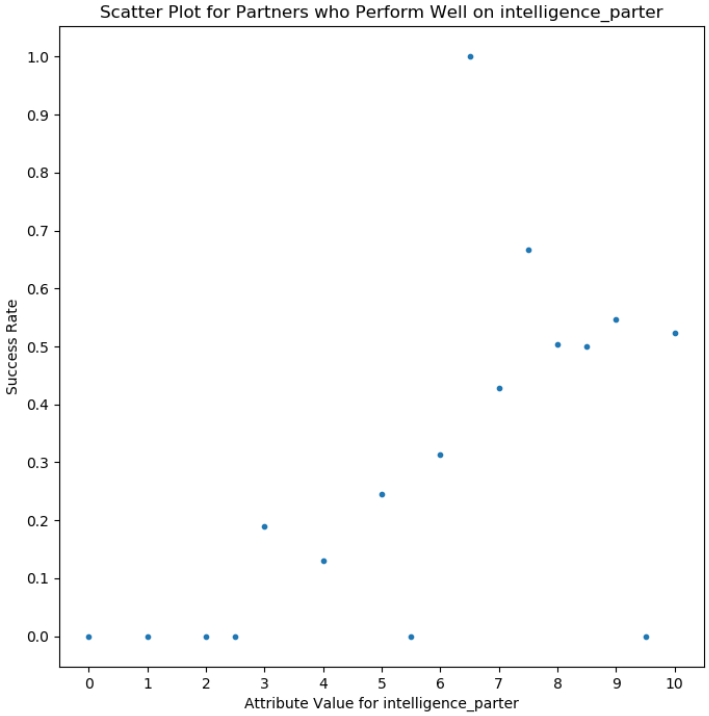

Figure 5:

5.2 Examine the effects of varying the number of bins for continuous attributes during the discretization step.

As we can observe from Figure 8, the accuracy for both training set and test set is relatively low when number of bins is very small i.e. 2. The main reason for this is that with very small number of bins, we have the size of a single bin to be very large. This simply means that we are not able to approximate the continuous valued attributes properly.

For example, if the range of an attribute is from 0 to 10, we are simply dividing it into 2 bins 0-5 and 5-10. Here, we are losing a lot of information since a value of 6 and value of 10 basically would mean the same thing in our model with num_bins = 2. This clearly is untrue.

Now, with increase in number of bins, we see an increase in both the training accuracy as well as the test accuracy. The training accuracy becomes arond 0.80

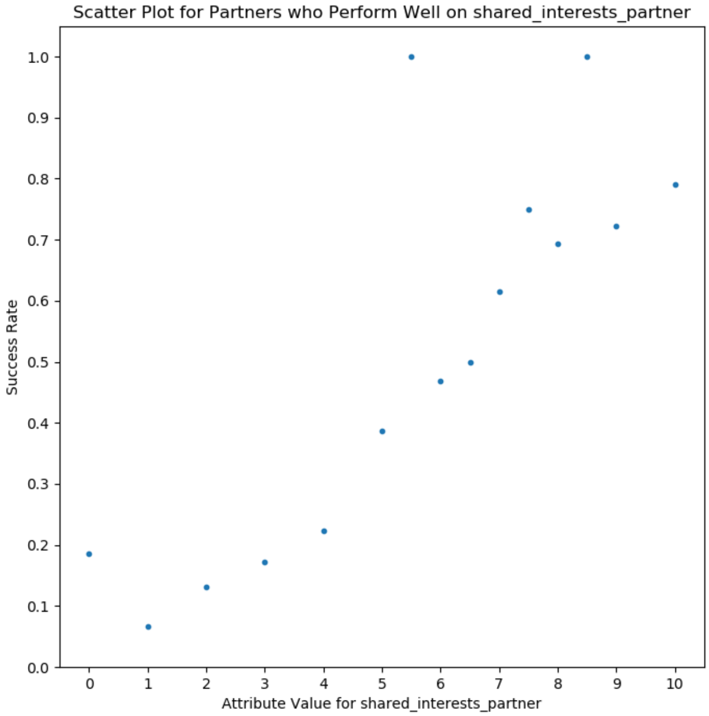

Figure 6:

and the test accuracy increases to around 0.755.

We also note that with further increase in number of bins from 100 to 200, we see a marginal drop in test accuracy. This is because we are simply discretizing it into continuous values and are not able to generalize as well to unseen test data.

5.3 Plot the learning curve

As we can observe from learning curve as seen in Figure 9, we start with a very high training score of around 0.93 and very low test score of around 0.67 for value of f being 0.01. The primary reason for this is that when the value of f is 0.01, we have very few samples in our training data set.

This means that Naive Bayes Classifier is overfitting on that very small training data and is not able to generalize on unseen test data.

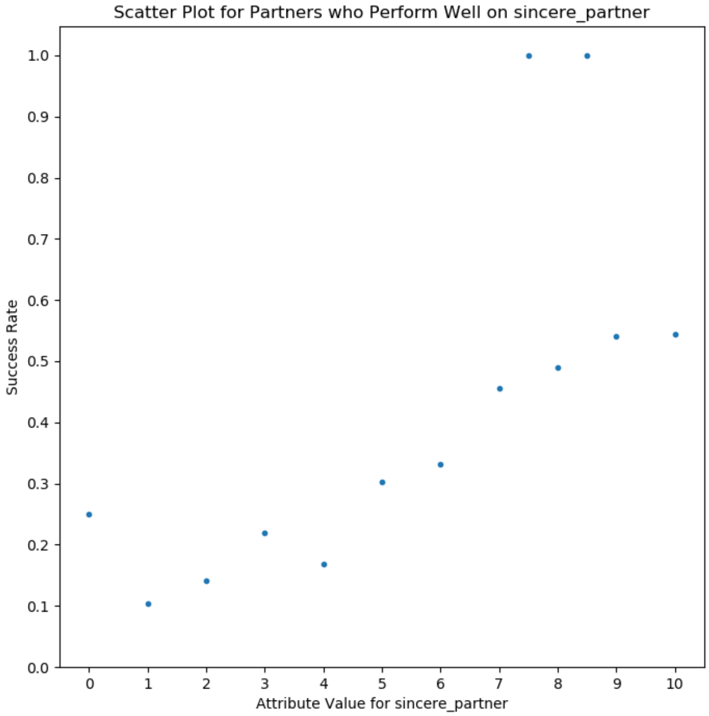

Figure 7:

With increase in value of f, we observe a drop in training accuracy to around 0.78 and increase in test accuracy to 0.75. This is because as value of f increases, we have much larger training data size. The Naive Bayes Classifier is fitting well on the training data set and is able to generalize on the unseen test data.

As the value of f becomes around 0.75, the difference between training accuracy and test accuracy becomes much smaller, this means that Naive Bayes Classifier is not overfitting on training data.

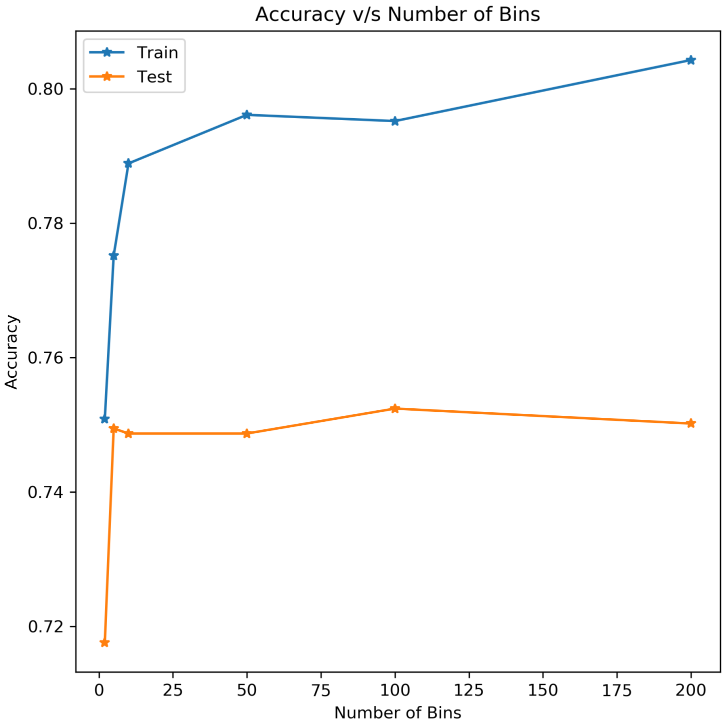
Accuracy v/s Number of Bins
Train
Test
0.80
0.78
0.76
0.74
0.72
Accuracy
0
25
50
75
100
125
150
175
200
Number of Bins

Figure 8:

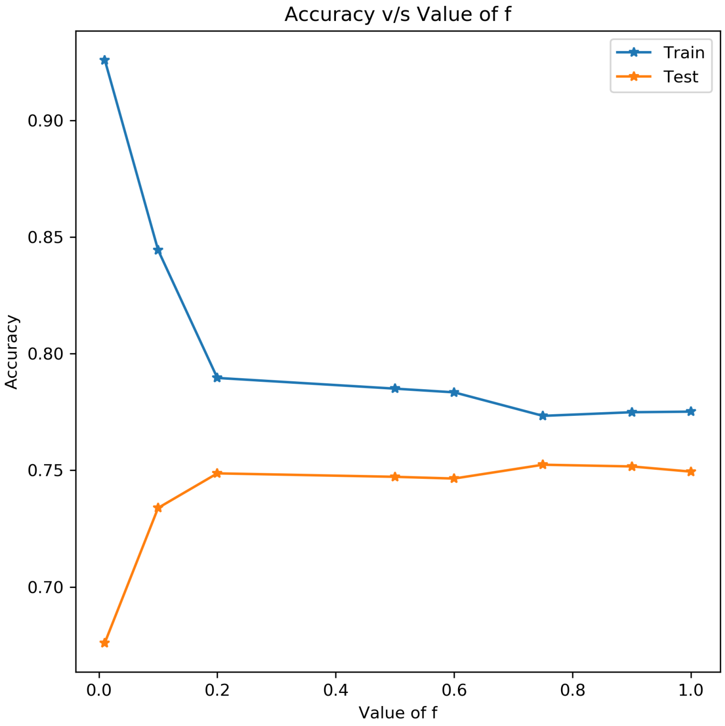
Accuracy v/s Value of f
Train
Test
Accuracy
0.70
0.75
0.80
0.85
0.90
Value of f
0.0
0.2
0.4
0.6
0.8
1.0

Figure 9: