CS57300: Data Mining

Assignment 2

Name: Mohammad Haseeb
Purdue ID: mhaseeb@purdue.edu

Due: February 13, 2019

Note: Figures appear at different locations in the document due to position issues of LaTeX.

1 Preprocessing

The file 'dating.csv' is stored in the same directory from which the script 'preprocess.py' was run. The output from the script is shown in Figure 1.

2 Visualizing interesting trends in data

(i) The barplot showing how males and females differ in terms of what are the attributes they value the most in their romantic partners is shown in Figure 2. The plot is stored in the file named 'plot_2_1.png' in the same directory from which the script '2_1.py' was run.

We can observe that males prefer attractiveness as the most important characteristic in a partner as compared to intelligence for women. Females give more preference to characteristics like sincerity, intelligence, ambition and shared interests. On the other hand, being attractive, intelligent and funny is more important for males.

(ii) The scatter plots showing 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 is shown in Figures 3 - 8.

We can observe several interesting trends from these scatter plots. For example, generally, the higher the participant rates the partner for any attribute, the greater are the chances of them giving a second date. More interestingly, if the participant gave a higher rating to the partner on attributes like attractiveness, funny, and shared interests, then the partner is more likely to get a second date i.e. the probability is higher. In comparison to this, even if a partner received higher ratings for the other attributes, approx. 55%-60% of the participants opted for a second date.

```
data 157 $ python3 preprocess.py dating-full.csv dating.csv
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: 0.2.
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: 0.2.
Mean of pref_o_funny: 0.17.
Mean of pref_o_ambitious: 0.11.
Mean of pref_o_shared_interests: 0.12.
```

Figure 1: Output of preprocess.py

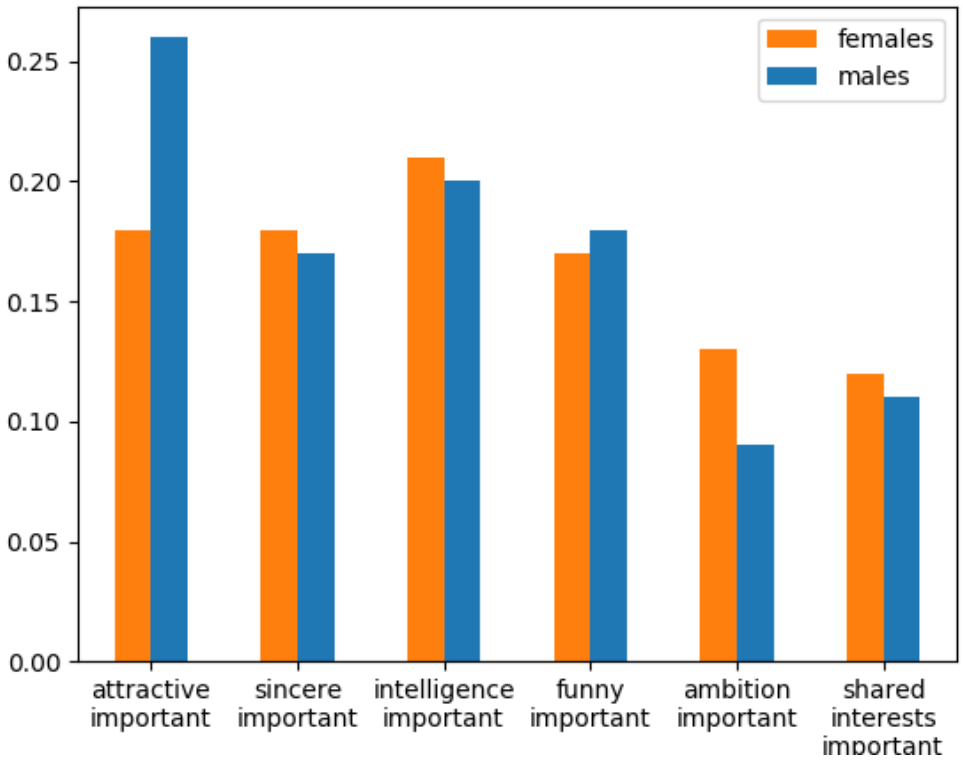

Figure 2: Output of 2_1.py

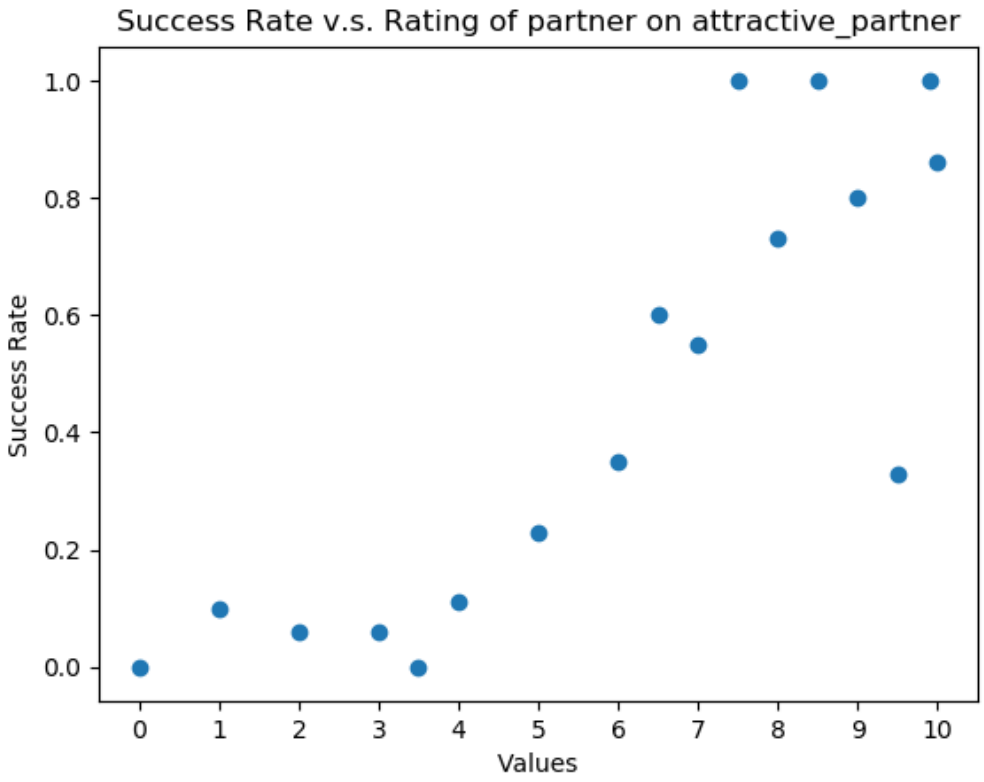

Figure 3: Output of 2_2.py

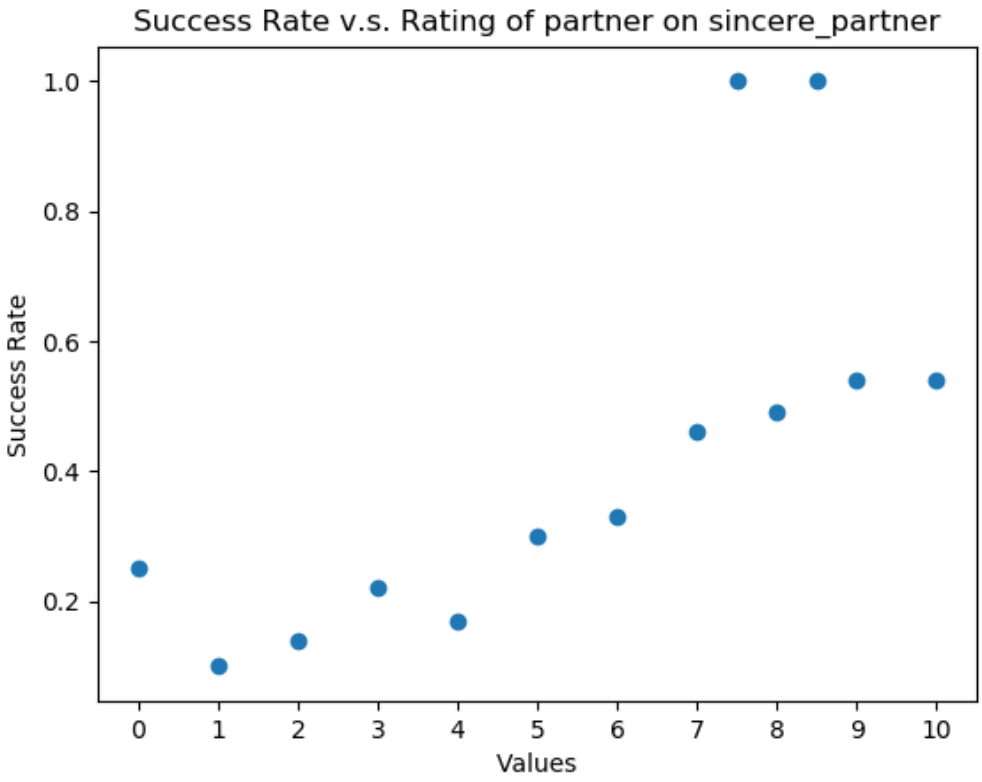

Figure 4: Output of 2_2.py

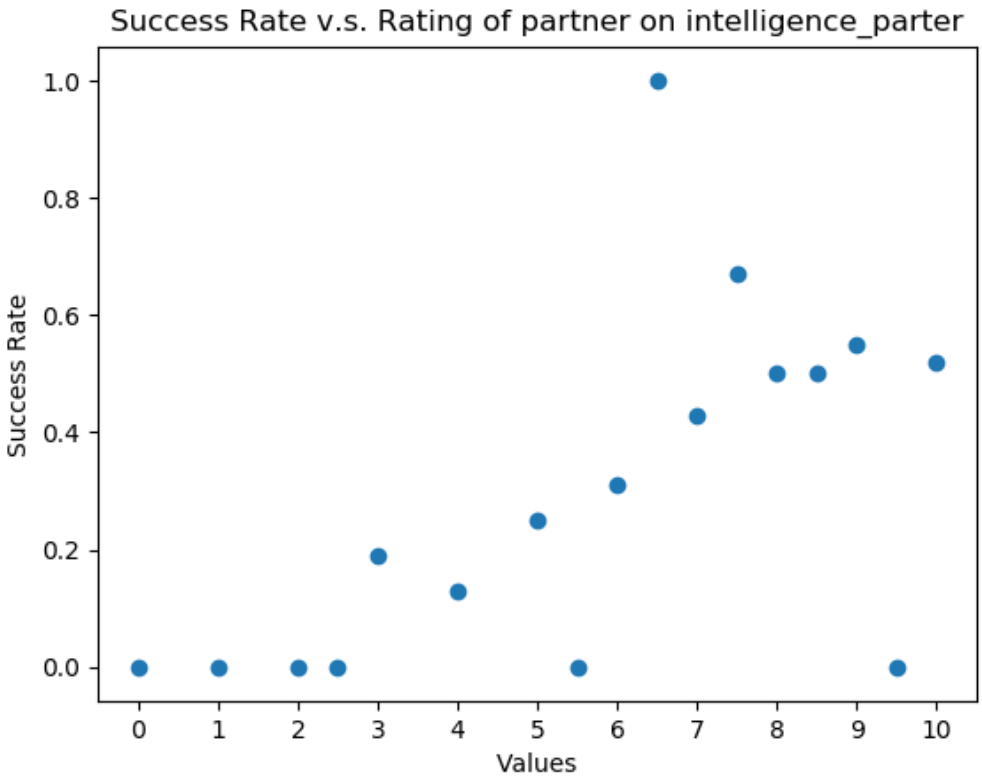

Figure 5: Output of 2_2.py

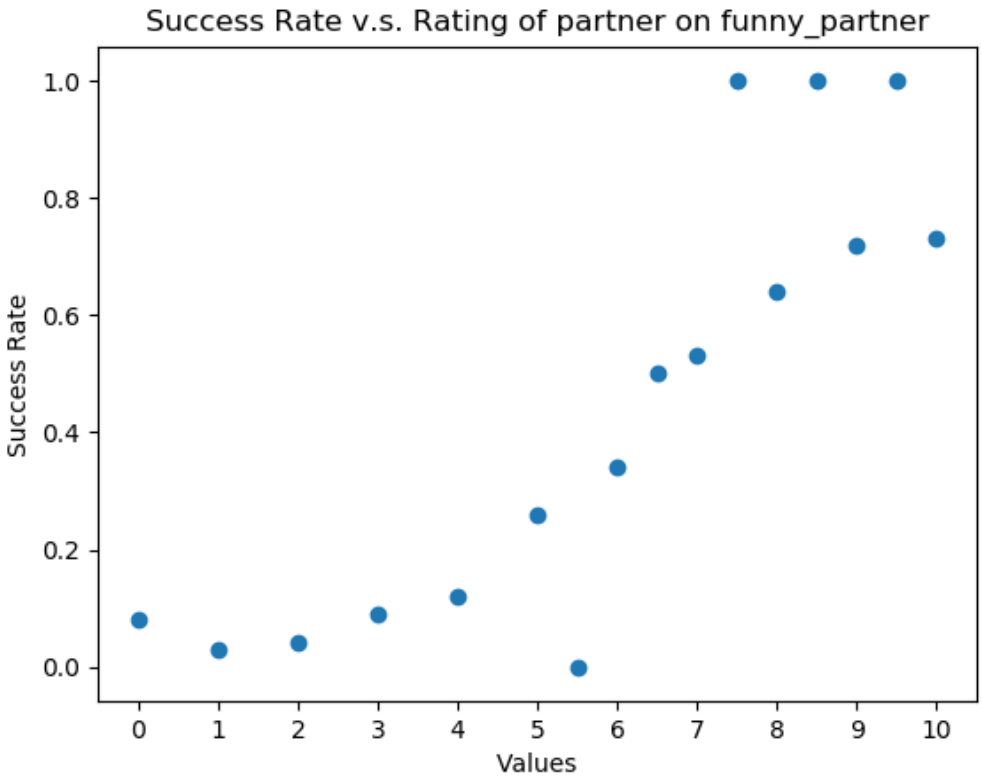

Figure 6: Output of 2_2.py

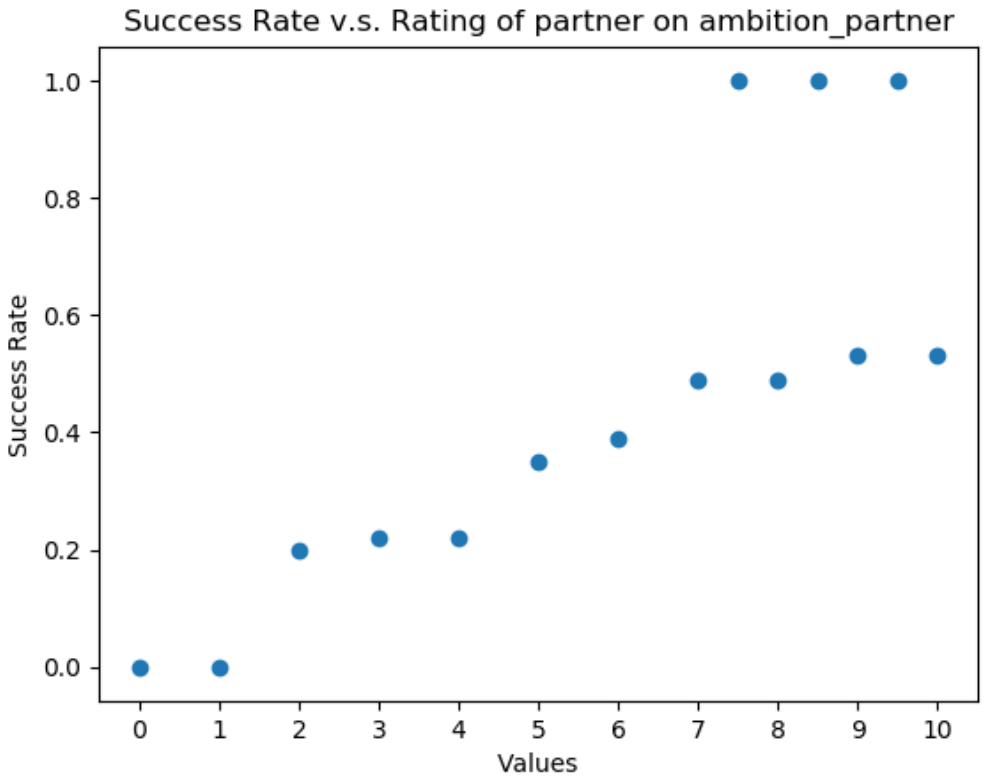

Figure 7: Output of 2_2.py

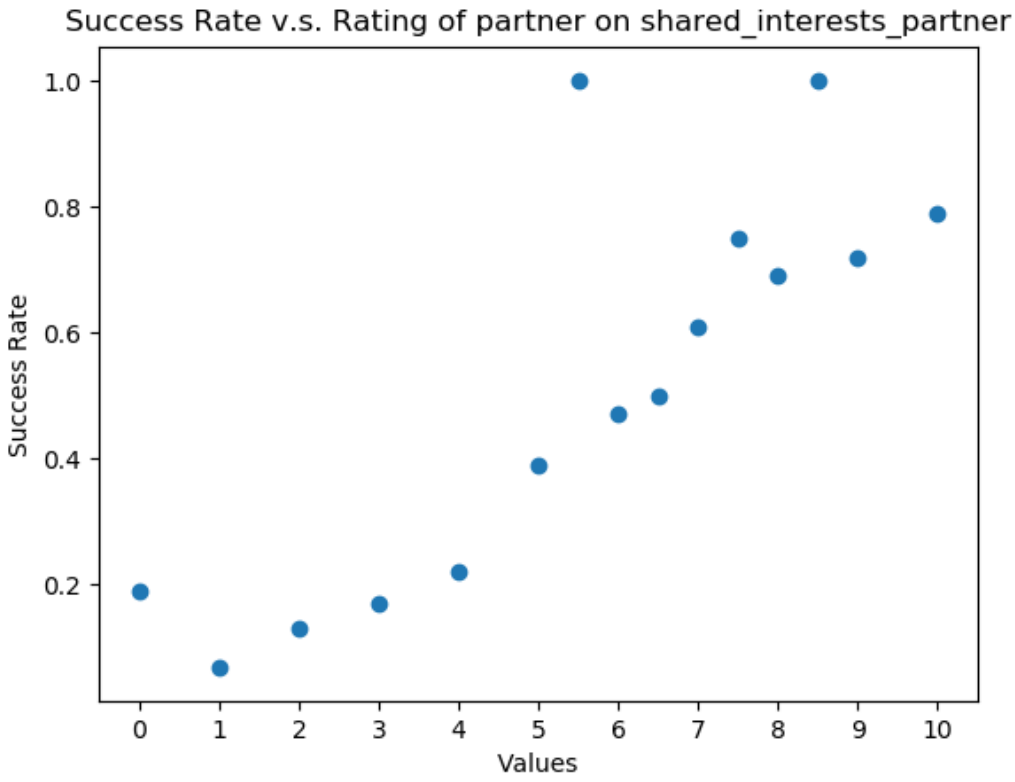

Figure 8: Output of 2_2.py

3 Convert continuous attributes to categorical attributes

The file 'dating-binned.csv' is stored in the same directory from which the script 'discretize.py' was run. The output from the script is shown in Figure 9 and Figure 10.

4 Training-Test Split

The output files 'trainingSet.csv' and 'testSet.csv' are stored in the same directory from which the script 'split.py' was run.

5 Implement a Naive Bayes Classifier

(i) Following is the output after running the script '5_1.py':

Training Accuracy: 77.2937905468026

Testing Accuracy: 75.31504818383988

(ii) The output from the script '5_2.py' is shown in Figure 11. The learning curve plot to show how the value of b affects the learned NBC model's performance on the training dataset and the test dataset is shown in Figure 12.

We can see from the plot that as the number of bins increases, both the training and the testing accuracy also increase. A possible reason for this is that if the number of bins are higher then we would get better probability estimates for the continuous values as each continuous value appearing in our data would likely end up in a different bin. So, when it is time to infer a label for a new row, the bin that a continuous attribute's value that this row has would have a higher probability. Therefore, as the number of bins increases, we approximate (learn) the dataset better.

(iii) The learning curve plot to show how the value of f affects the learned NBC model's performance on the training dataset and the test dataset is shown in Figure 13.

We can see that as the size of training data to learn on increases, the training accuracy decreases and the test accuracy increases. This shows that our model is going from over-fitting to generalizing/learning the dataset better. When our training data is very small, we are learning the dataset probabilities very well, therefore, we get a very good accuracy for training set. However, because we have basically memorized the training datas, we over-fit and get a bad test accuracy. As the training set size increases, we learn more probabilities and therefore, generalize better and increase the test accuracy.

```
data 193 $ python3 discretize.py dating.csv dating-binned.csv
age: [3710, 2932, 97, 0, 5]
age_o: [3704, 2899, 136, 0, 5]
importance_same_race: [2980, 1213, 977, 1013, 561]
importance_same_religion: [3203, 1188, 1110, 742, 501]
pref_o_attractive: [4333, 1987, 344, 51, 29]
pref_o_sincere: [5500, 1225, 19, 0, 0]
pref_o_intelligence: [4601, 2062, 81, 0, 0]
pref_o_funny: [5616, 1103, 25, 0, 0]
pref_o_ambitious: [6656, 88, 0, 0, 0]
pref_o_shared_interests: [6467, 277, 0, 0, 0]
attractive_important: [4323, 2017, 328, 57, 19]
sincere_important: [5495, 1235, 14, 0, 0]
intelligence_important: [4606, 2071, 67, 0, 0]
funny_important: [5588, 1128, 28, 0, 0]
ambition_important: [6644, 100, 0, 0, 0]
shared_interests_important: [6494, 250, 0, 0, 0]
attractive: [18, 276, 1462, 4122, 866]
sincere: [33, 117, 487, 2715, 3392]
intelligence: [34, 185, 1049, 3190, 2286]
funny: [0, 19, 221, 3191, 3313]
ambition: [84, 327, 1070, 2876, 2387]
attractive_partner: [284, 948, 2418, 2390, 704]
sincere_partner: [94, 353, 1627, 3282, 1388]
intelligence_parter: [36, 193, 1509, 3509, 1497]
funny_partner: [279, 733, 2296, 2600, 836]
ambition_partner: [119, 473, 2258, 2804, 1090]
shared_interests_partner: [701, 1269, 2536, 1774, 464]
sports: [650, 961, 1369, 2077, 1687]
tvsports: [2151, 1292, 1233, 1383, 685]
exercise: [619, 952, 1775, 2115, 1283]
dining: [39, 172, 1118, 2797, 2618]
museums: [117, 732, 1417, 2737, 1741]
art: [224, 946, 1557, 2500, 1517]
hiking: [963, 1386, 1575, 1855, 965]
gaming: [2565, 1522, 1435, 979, 243]
clubbing: [912, 1068, 1668, 2193, 903]
reading: [131, 398, 1071, 2317, 2827]
```

```
reading: [131, 398, 1071, 2317, 2827]
tv: [1188, 1216, 1999, 1642, 699]
theater: [288, 811, 1585, 2300, 1760]
movies: [45, 248, 843, 2783, 2825]
concerts: [222, 777, 1752, 2282, 1711]
music: [62, 196, 1106, 2583, 2797]
shopping: [1093, 1098, 1709, 1643, 1201]
yoga: [2285, 1392, 1369, 1056, 642]
interests_correlate: [18, 758, 2520, 2875, 573]
expected_happy_with_sd_people: [321, 1262, 3292, 1596, 273]
like: [273, 865, 2539, 2560, 507]
```

Figure 10: Output of discretize.py

```
data 215 $ python3 5_2.py
Bin size: 2
Training Accuracy: 75.05097312326228
Testing Accuracy: 72.72053372868793
Bin size: 5
Training Accuracy: 77.2937905468026
Testing Accuracy: 75.31504818383988
Bin size: 10
Training Accuracy: 78.68396663577386
Testing Accuracy: 76.5752409191994
Bin size: 50
Training Accuracy: 79.72196478220575
Testing Accuracy: 77.53891771682729
Bin size: 100
Training Accuracy: 79.75903614457832
Testing Accuracy: 78.05782060785768
Bin size: 200
Training Accuracy: 80.48192771084337
Testing Accuracy: 78.42846553002224
```

Figure 11: Output of 5_2.py

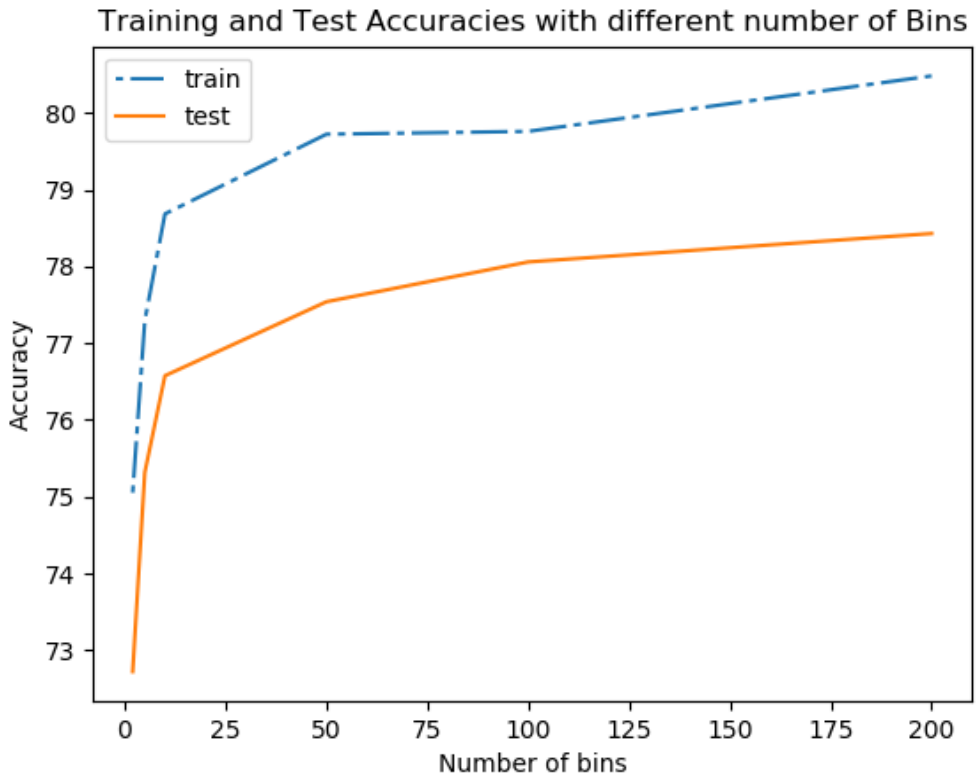

Figure 12: Affect of changing number of bins on the performance of NBC model.

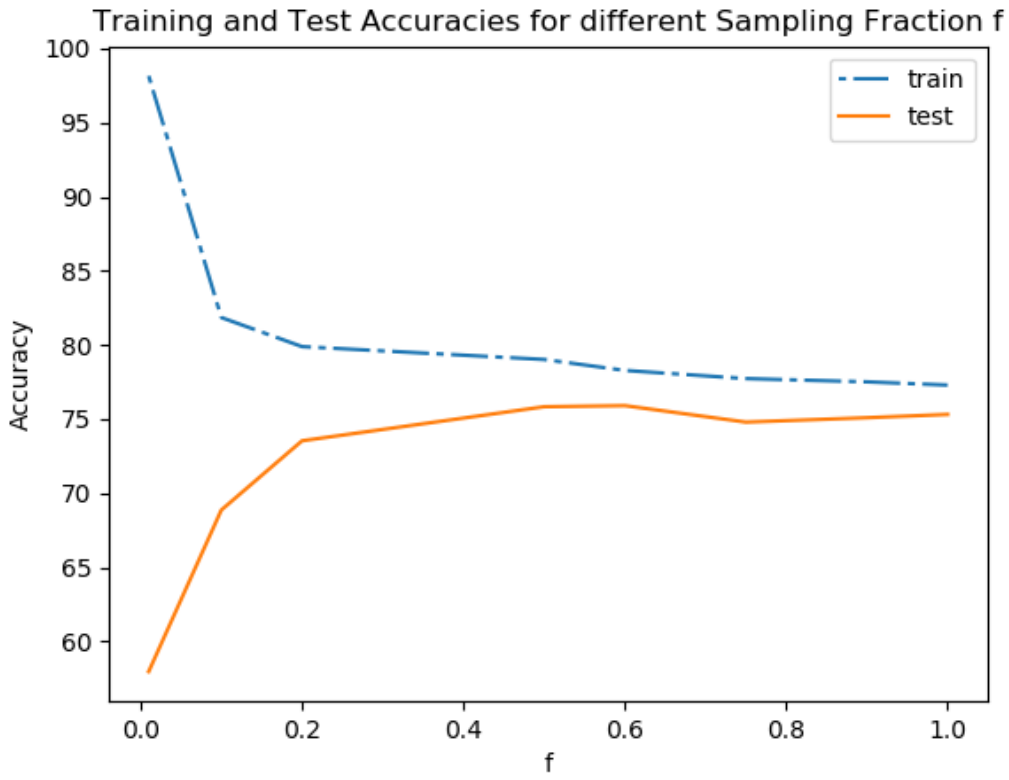

Figure 13: Affect of changing the value of f on performance of NBC model.