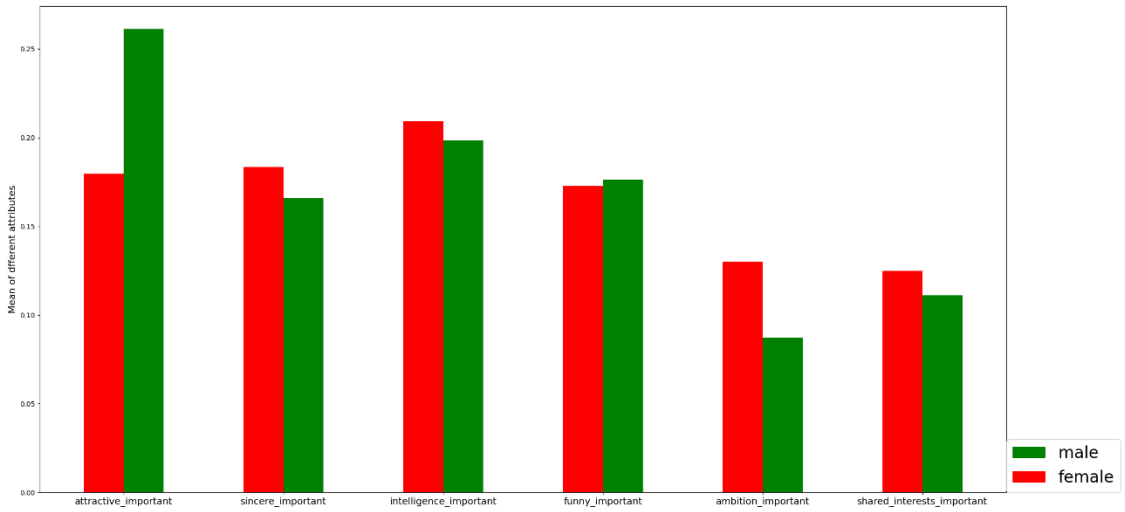
1

```
In [301]: runfile('C:/Users/Ben/Desktop/cs573/hw2/2020F-CS573-HW2/py
folder/preprocess.py', wdir='C:/Users/Ben/Desktop/cs573/hw2/2020F-
CS573-HW2/py folder')
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
```

2_1



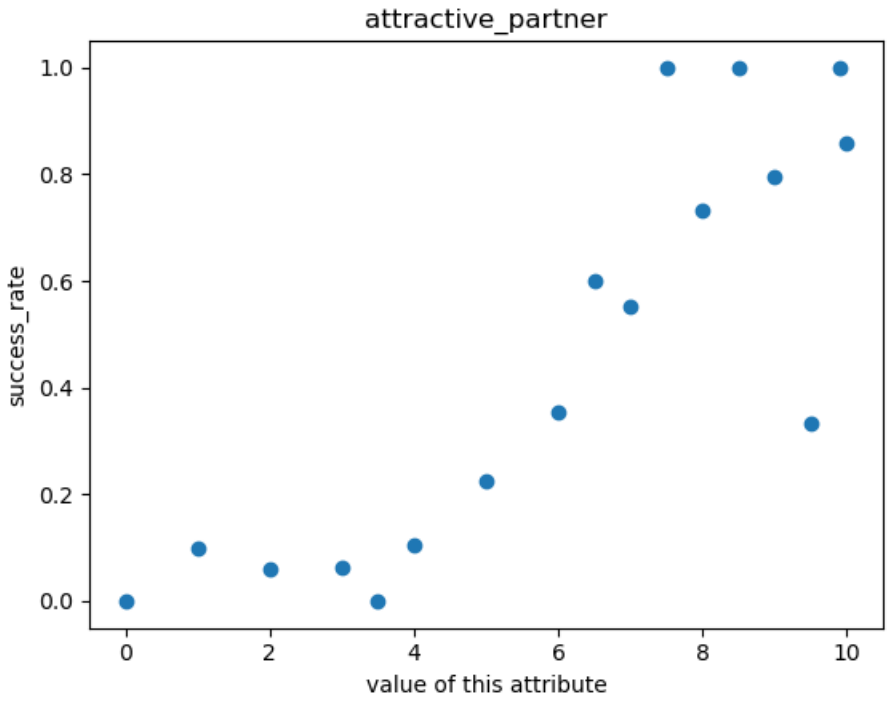
From the plot we can see that males in our data believe that the attractiveness of a partner is the most important among all attributes. The importance of other attributes decreases in the order of intelligence, funny,

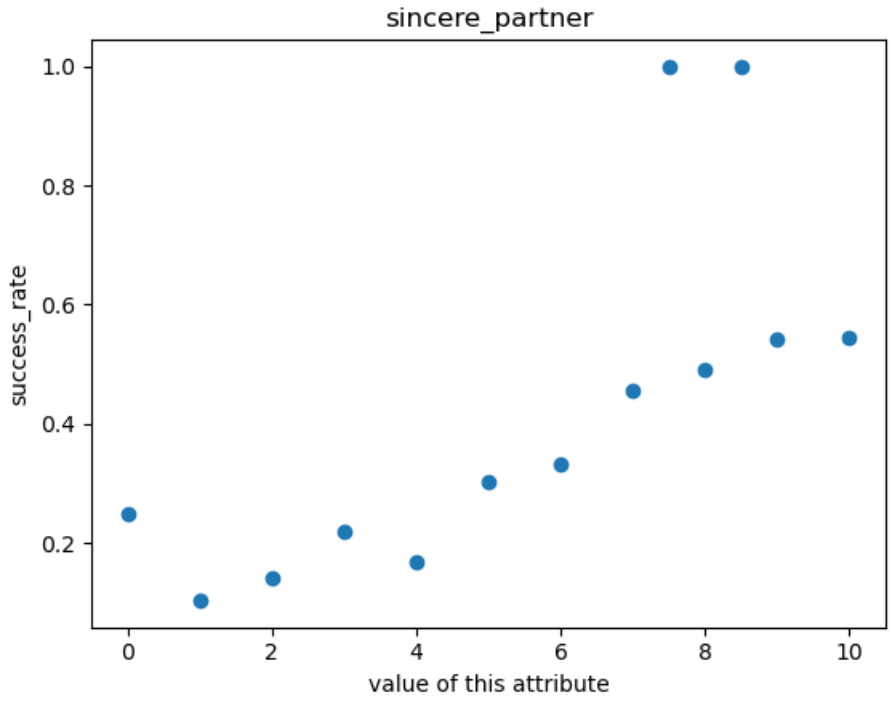
sincere, shared interest, and ambition is the least important attribute.

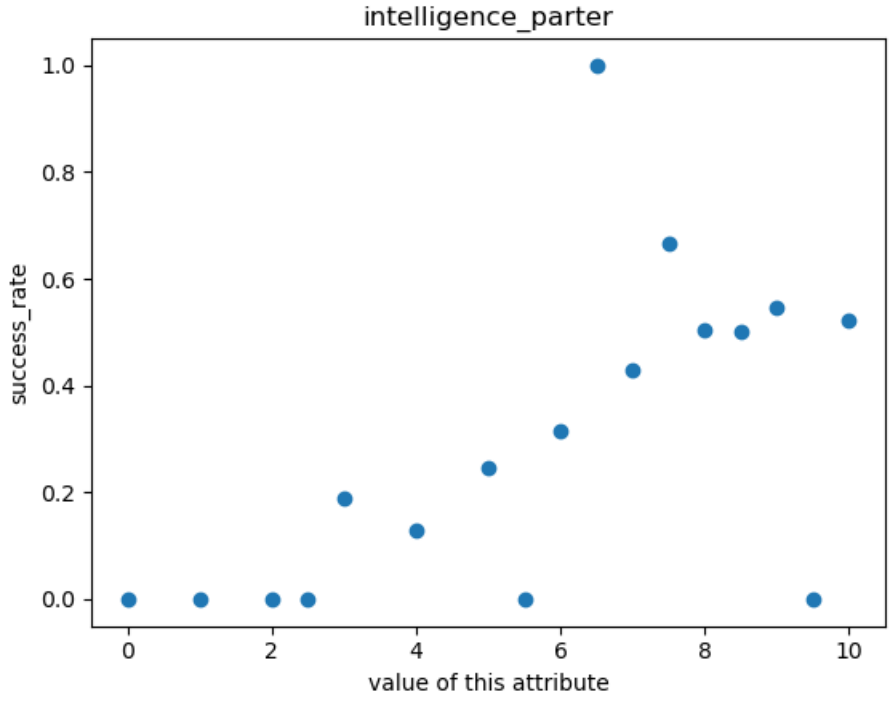
However, females in our data prefer intelligence among all other attributes. And it looks like that the score difference between the most and least valued attributes of females is smaller than that of males.

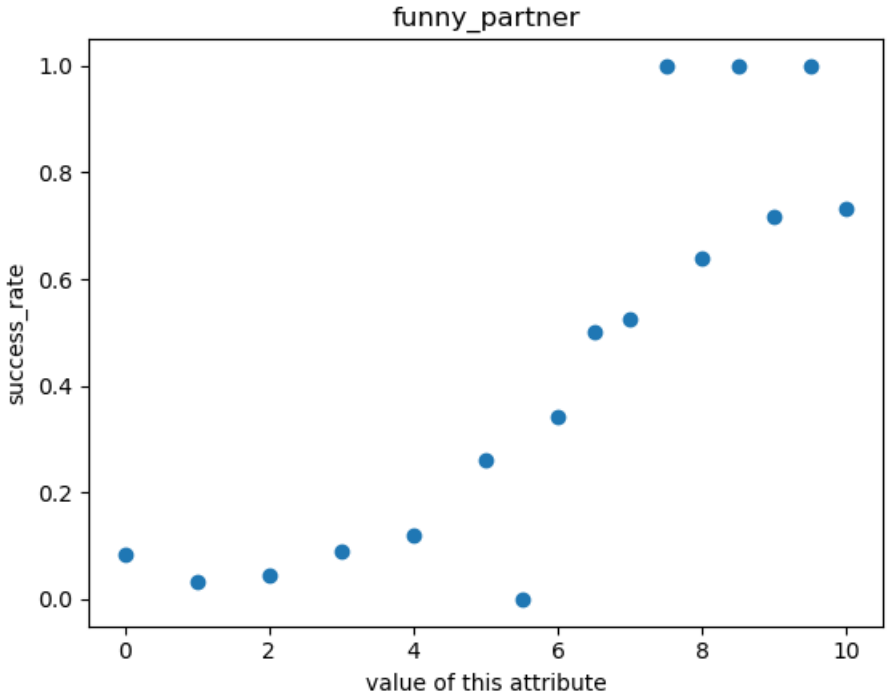
2_2

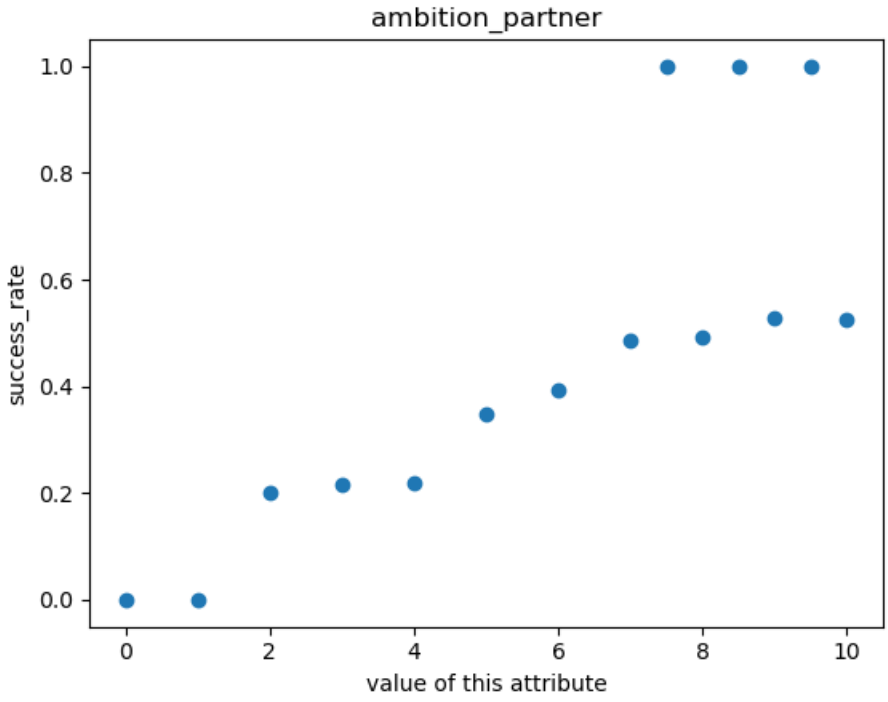
```
In [91]: runfile('C:/Users/Ben/Desktop/cs573/hw2/2020F-CS573-HW2/py folder/2_2.py', wdir='C:/
Users/Ben/Desktop/cs573/hw2/2020F-CS573-HW2/py folder')
number of distinct values for attribute attractive_partner : 17
number of distinct values for attribute sincere_partner : 13
number of distinct values for attribute intelligence_parter : 17
number of distinct values for attribute funny_partner : 16
number of distinct values for attribute ambition_partner : 14
number of distinct values for attribute shared_interests_partner : 15
```

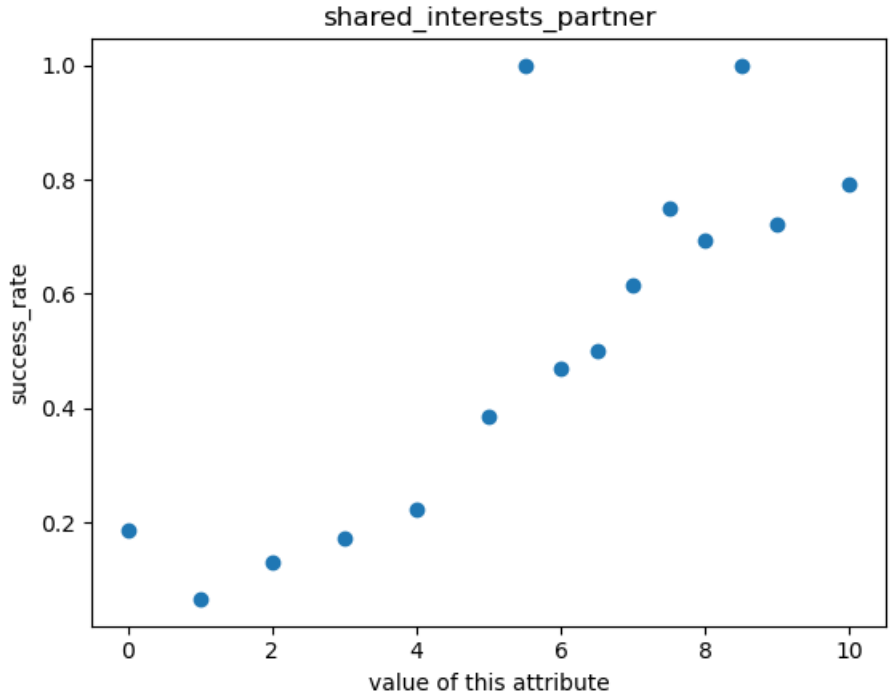



sincere_partner
success_rate
value of this attribute
1.0
0.8
0.6
0.4
0.2
0
2
4
6
8
10


intelligence_parter
success_rate
value of this attribute
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10


funny_partner
success_rate
value of this attribute
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10


ambition_partner
success_rate
value of this attribute
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10



From the above plots we can estimate that the success rate of each attribute is almost linearly dependent on the value of the attribute, where higher value of an attribute leads to higher success rate. Some data points may be considered outliers due to large deviation from the other points.

3.

runfile('C:/Users/Ben/Desktop/cs573/hw2/2020F-CS573-HW2/py folder/discretize.py', wdir='C:/Users/Ben/Desktop/cs573/hw2/2020F-CS573-HW2/py folder')

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]

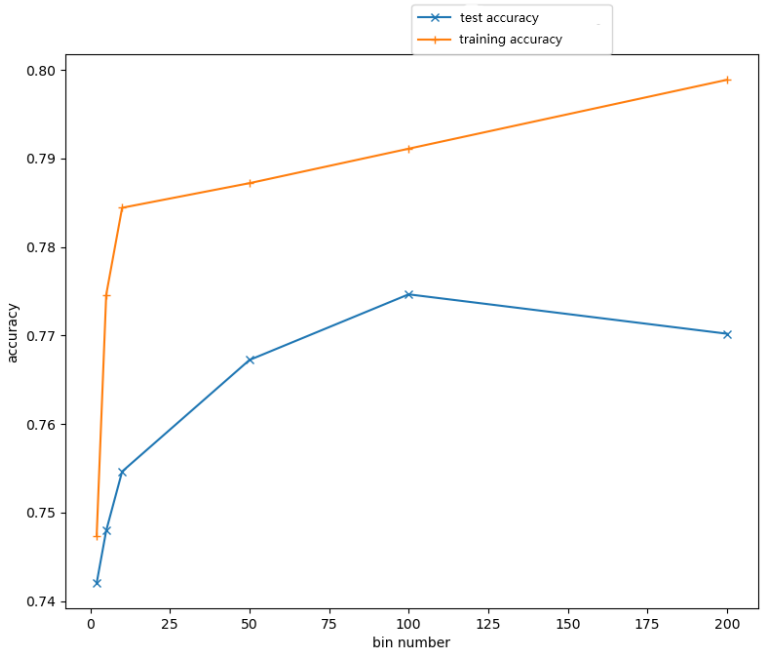
sincere_important : [5495 1235 14 0 0]
intelligence_important : [4606 2071 67 0 0]
funny_important : [5588 1128 28 0 0]
ambition_important : [6644 100 0 0 0]
shared_interests_important : [6494 250 0 0 0]
attractive : [18 276 1462 4122 866]
sincere : [33 117 487 2715 3392]
intelligence : [34 185 1049 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]
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]

5_1

Training Accuracy: 0.77

Test Accuracy: 0.75

5_2

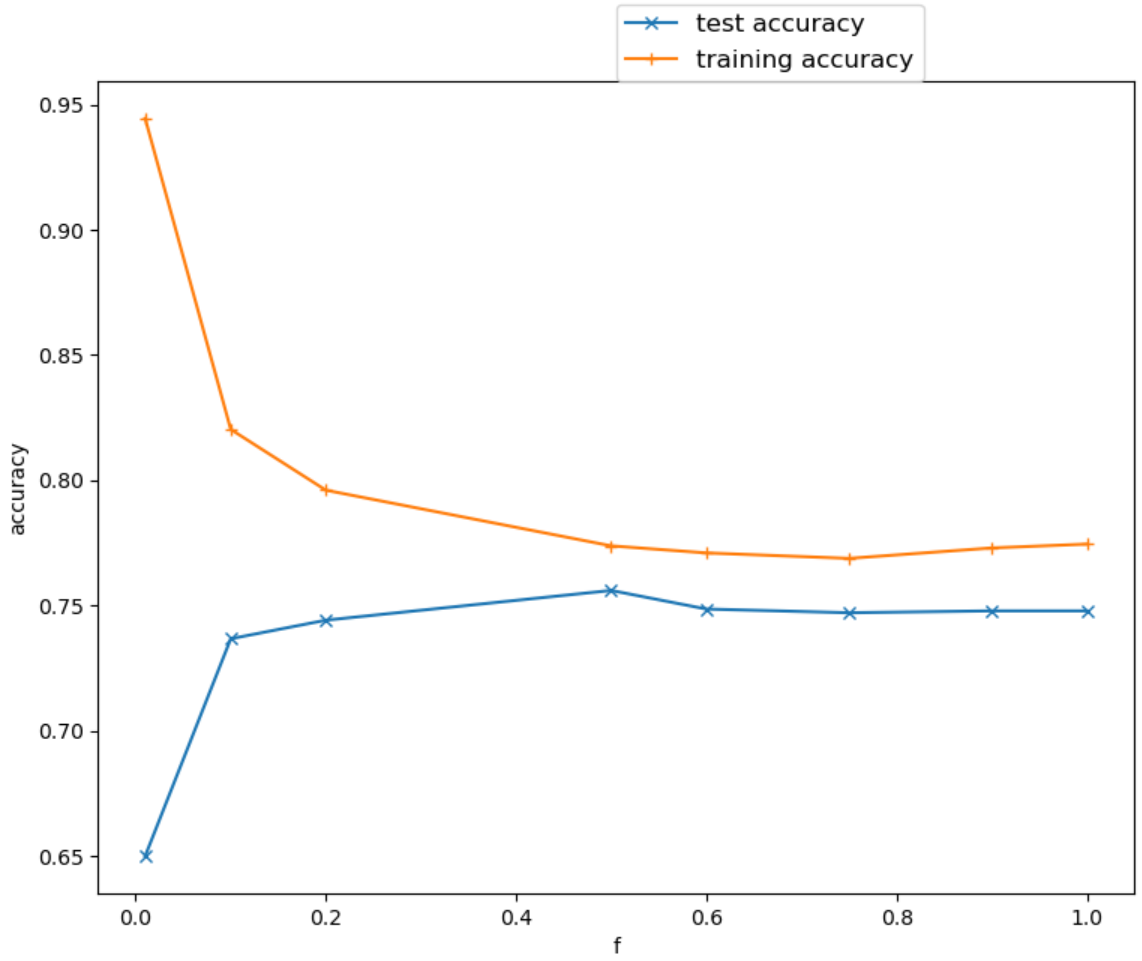



```
Bin size: 2
Training Accuracy: 0.75
Testing Accuracy: 0.74
Bin size: 5
Training Accuracy: 0.77
Testing Accuracy: 0.75
Bin size: 10
Training Accuracy: 0.78
Testing Accuracy: 0.75
Bin size: 50
Training Accuracy: 0.79
Testing Accuracy: 0.77
Bin size: 100
Training Accuracy: 0.79
Testing Accuracy: 0.77
Bin size: 200
Training Accuracy: 0.80
Testing Accuracy: 0.77
```

We can see that the accuracy of both training and test are increasing with the number of bins increasing. The accuracy of training is always higher than test. The trend of accuracy increase of both curves shows sign of exponential or logarithmic function. The upper bond of training curve is higher than test curve.

5_3

```
f : 0.01
Training Accuracy: 0.94
Test Accuracy: 0.65
f : 0.1
Training Accuracy: 0.82
Test Accuracy: 0.74
f : 0.2
Training Accuracy: 0.80
Test Accuracy: 0.74
f : 0.5
Training Accuracy: 0.77
Test Accuracy: 0.76
f : 0.6
Training Accuracy: 0.77
Test Accuracy: 0.75
f : 0.75
Training Accuracy: 0.77
Test Accuracy: 0.75
f : 0.9
Training Accuracy: 0.77
Test Accuracy: 0.75
f : 1
Training Accuracy: 0.77
Test Accuracy: 0.75
```



From the output data and plot we can see that the training accuracy monotonically decreases, and the test accuracy monotonically increases as the value of f increases. This is reasonable since when training on small set of data, accuracy of predicting the same small set of data is always high. However, the bias of small dataset is obvious, it led to a poor test accuracy. When we increase the amount of training data, we can see that the accuracy of both training and test becomes stable and very close to each other in this case. That means the more data we use to train our model, the better test accuracy it obtains whenever a new data comes in.