تمرین سوم

سوال اول-بخش اول)

```
   support itemsets  length
0      0.6      (A)       1
1      0.8      (B)       1
2      0.4      (C)       1
3      0.4      (D)       1
4      0.9      (E)       1
====================================
   support itemsets  length
5      0.4   (A, B)       2
6      0.4   (A, C)       2
7      0.5   (A, E)       2
8      0.7   (B, E)       2
9      0.4   (C, E)       2
====================================
   support itemsets  length
8      0.7   (B, E)       2
====================================
  antecedents consequents  support  confidence
0         (C)         (E)      0.4         1.0
1      (A, C)         (E)      0.4         1.0
2      (C, E)         (A)      0.4         1.0
3         (C)      (A, E)      0.4         1.0
4         (C)         (A)      0.4         1.0
====================================
confidence of the association rule {B} => {E} is : 0.8749999999999999
```

این خروجیهای به ترتیب سوالهای a تا e است.

در این سوال ورودی به شکل یک فایل csv داده شده است.

```
t001,A,B,D,G,NaN
t002,B,D,E,NaN,NaN
t003,A,B,C,E,F
t004,B,D ,E,G,NaN
t005,A,B,C,E,F
t006,B,E,G,NaN,NaN
t007,A,C,D,E,NaN
t008,B,E,NaN,NaN,NaN
t009,A,B,E,F,NaN
t010,A,C,D,E,NaN
```

در این فایل برای اینکه همه‌ی ردیف ها هم اندازه باشند از NaN کمک گرفته شده.ستون اول ID است و بقیه ستونها itemهاست.

```
store_data = pd.read_csv('venv/apriori.csv', header=None)
records = []
for i in range(0, 10):
    records.append([str(store_data.values[i,j]) for j in range(1, 6)])
```

در کد این خط برای خواندن فایل ورودی و ریختن آن در آرایه records است.

در ادامه برای راحتتر کار کردن با ورودی آن را به این صورت درمیاریم.

	A	B	C	D	D	E	F	G
0	True	True	False	True	False	False	False	True
1	False	True	False	True	False	True	False	False
2	True	True	True	False	False	True	True	False
3	False	True	False	False	True	True	False	True
4	True	True	True	False	False	True	True	False
5	False	True	False	False	False	True	False	True
6	True	False	True	True	False	True	False	False
7	False	True	False	False	False	True	False	False
8	True	True	False	False	False	True	True	False
9	True	False	True	True	False	True	False	False

که برای این تبدیل این کدها نیاز بود

```
te = TransactionEncoder()
te_ary = te.fit(records).transform(records)
df = pd.DataFrame(te_ary, columns=te.columns_)
df = df.drop(columns='nan')
```

و در آخر NaN را نیز حذف کردیم چون نیازی بهش نداریم.

```
frequent_itemsets = apriori(df, min_support=0.4, use_colnames=True)
frequent_itemsets['length'] = frequent_itemsets['itemsets'].apply(lambda x: len(x))
```

در ادامه frequent item set ها رو بدست میاریم. و طبق گفته سوال min support باید 4/10 باشد. در خط بعدی کد برای هر length ای داده هارا جدا میکنیم.

```
res = association_rules(frequent_itemsets, metric="confidence", min_threshold=1)
res = res[['antecedents', 'consequents', 'support', 'confidence']]
```

سپس برای سوال بعدی association rule ها رو برای frequent item set ها با min confidence ۱ بدست میاریم. در ادامه فقط ستون های گفته شده را نگه میداریم چون اطلاعات خیلی زیادی بدست میاورد.

```
res2 = association_rules(frequent_itemsets, metric="confidence", min_threshold=0)
res2 = res2[['antecedents', 'consequents', 'support', 'confidence']]

for i in range(0,15):
    if res2['antecedents'][i]==set('B') and res2['consequents'][i]==set('E'):
        print("confidence of the association rule {B} => {E} is : " + str(res2['confidence'][i]))
```

برای قسمت آخر همه association rule ها رو تولید میکنیم و دنبال rule گفته شده در سوال میگردیم. سپس confidence آن را چاپ میکنیم.

قسمت دوم)

```
   support   itemsets  length
0     0.50         (A)       1
1     0.75         (B)       1
2     0.75         (C)       1
3     0.75         (E)       1
4     0.50      (A, C)       2
5     0.50      (C, B)       2
6     0.75      (B, E)       2
7     0.50      (C, E)       2
8     0.50   (C, B, E)       3
=====================================
   antecedents consequents  support  confidence
0          (C)          (E)     0.50    0.666667
1          (E)          (C)     0.50    0.666667
2       (C, B)          (E)     0.50    1.000000
3       (C, E)          (B)     0.50    1.000000
4       (B, E)          (C)     0.50    0.666667
5          (C)       (B, E)     0.50    0.666667
6          (B)       (C, E)     0.50    0.666667
7          (E)       (C, B)     0.50    0.666667
8          (B)          (E)     0.75    1.000000
9          (E)          (B)     0.75    1.000000
10         (C)          (B)     0.50    0.666667
11         (B)          (C)     0.50    0.666667
12         (A)          (C)     0.50    1.000000
13         (C)          (A)     0.50    0.666667
=====================================
  antecedents consequents  support  confidence
0      (C, B)          (E)     0.50         1.0
1      (C, E)          (B)     0.50         1.0
2         (B)          (E)     0.75         1.0
3         (E)          (B)     0.75         1.0
4         (A)          (C)     0.50         1.0
=====================================
confidence of the association rule {E} => {C} is : 0.6666666666666666
support value of the association rule {B} => {C} is : 0.5
```

این خروجیهای به ترتیب سوالهای a تا e است.

در این سوال ورودی به شکل یک فایل csv داده شده است.

```
t001,A,C,D,NaN
t002,B,C,E,NaN
t003,A,B,C,E
t004,B,E,NaN,NaN
```

مانند قسمت قبل در این فایل برای اینکه همه‌ی ردیف ها هم اندازه باشند از NaN کمک گرفته شده.ستون اول ID است و بقیه ستونها itemهاست.

در کد این خط برای خواندن فایل ورودی و ریختن آن در آرایه records است.

در ادامه برای راحتتر کار کردن با ورودی آن را به این صورت درمیاریم.

```
store_data = pd.read_csv('venv/apriori2.csv', header=None)

records = []
for i in range(0, 4):
    records.append([str(store_data.values[i,j]) for j in range(1, 5)])
te = TransactionEncoder()
te_ary = te.fit(records).transform(records)
df = pd.DataFrame(te_ary, columns=te.columns_)
df = df.drop(columns='nan')
```

	A	B	C	D	E
0	True	False	True	True	False
1	False	True	True	False	True
2	True	True	True	False	True
3	False	True	False	False	True

برای سوال اول باید frequent itemeset ها رو بدست بیاریم.

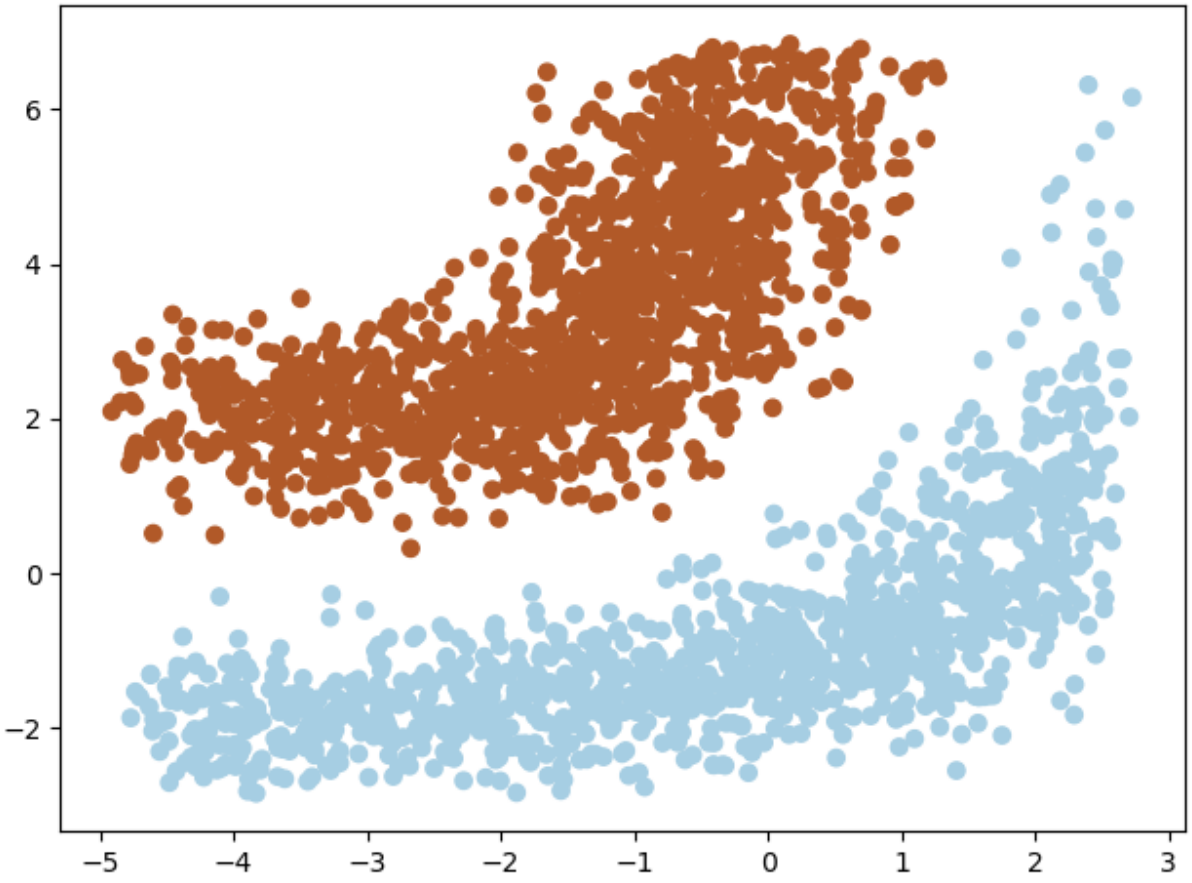
```
frequent_itemsets = apriori(df, min_support=0.5, use_colnames=True)
frequent_itemsets['length'] = frequent_itemsets['itemsets'].apply(lambda x: len(x))
```

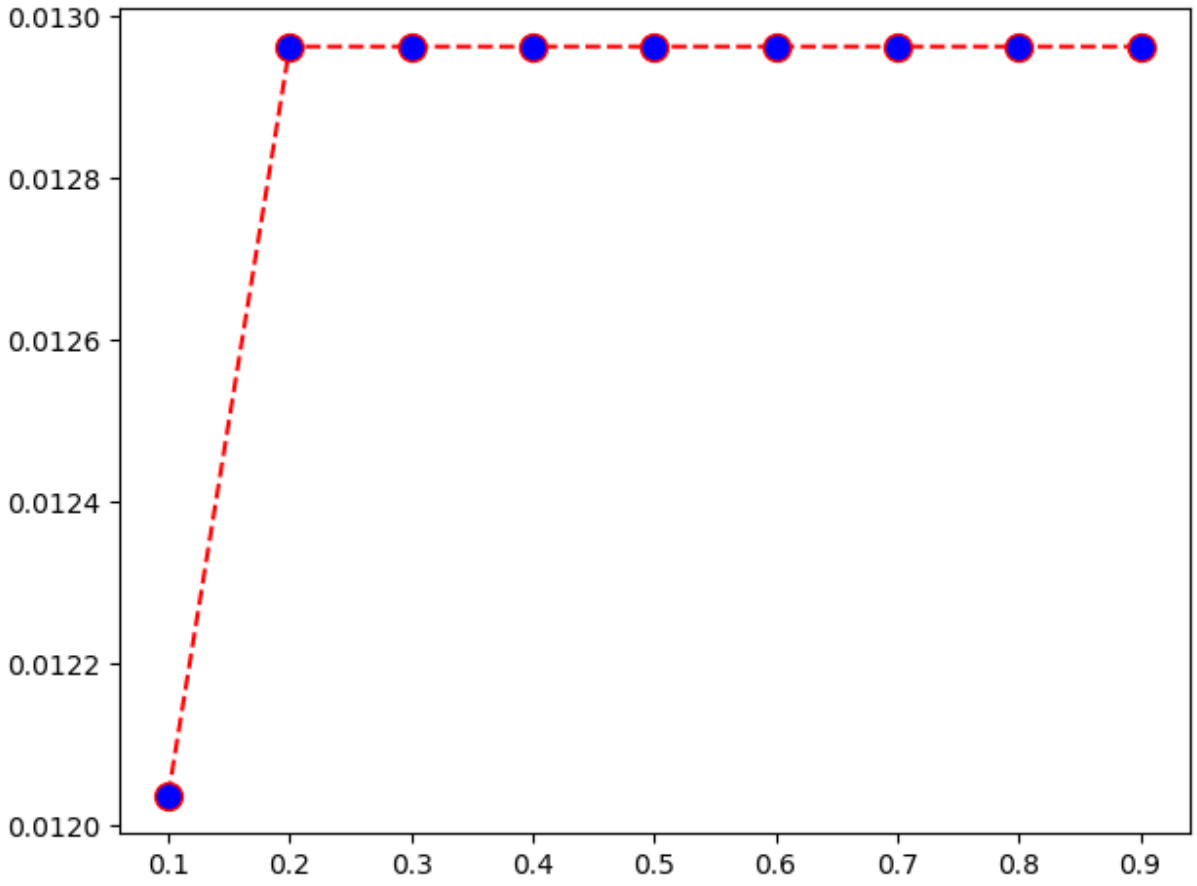
در ادامه هم association rule ها رو تولید میکنیم.

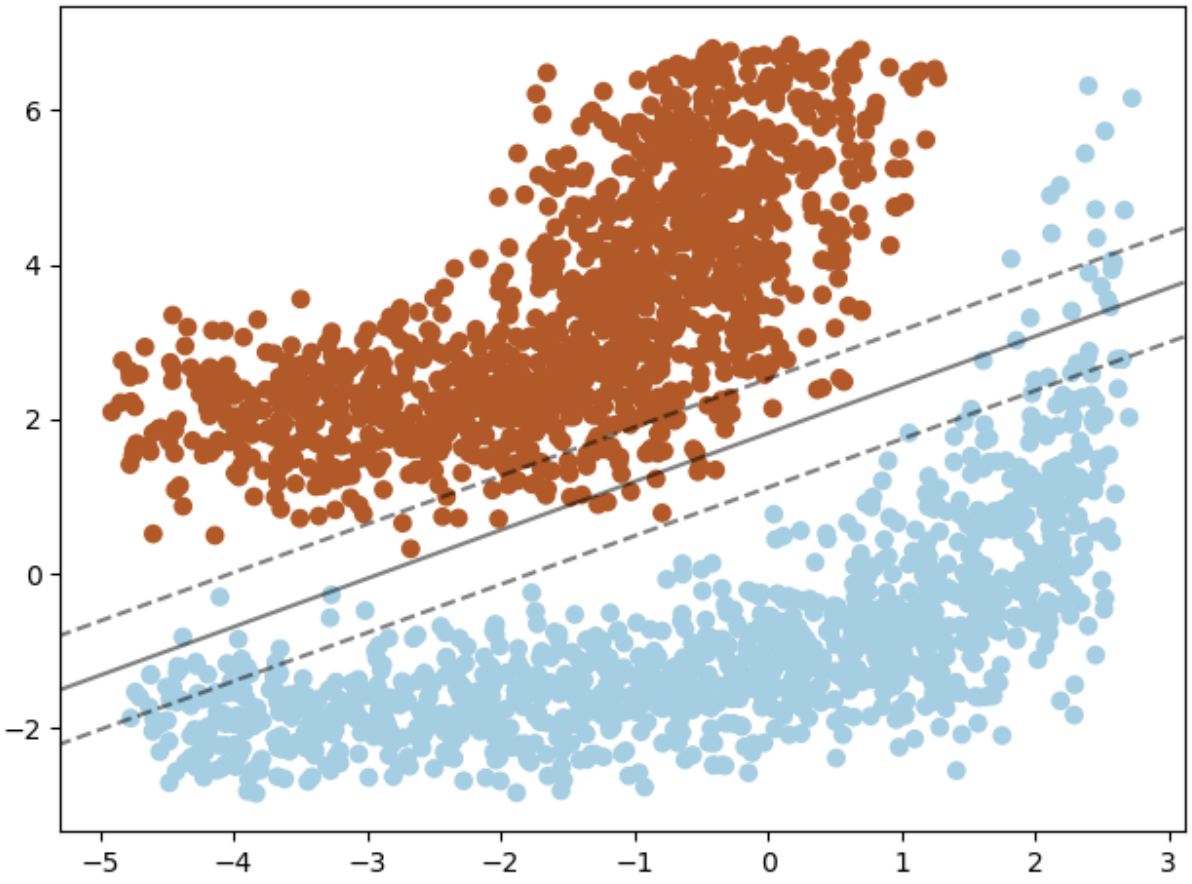
```
res = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.65)
res = res[['antecedents', 'consequents', 'support', 'confidence']]
print(res)
print("====================================")


res2 = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.8)
res2 = res2[['antecedents', 'consequents', 'support', 'confidence']]
print(res2)
print("====================================")
```

سوال ۲) قسمت ۱)

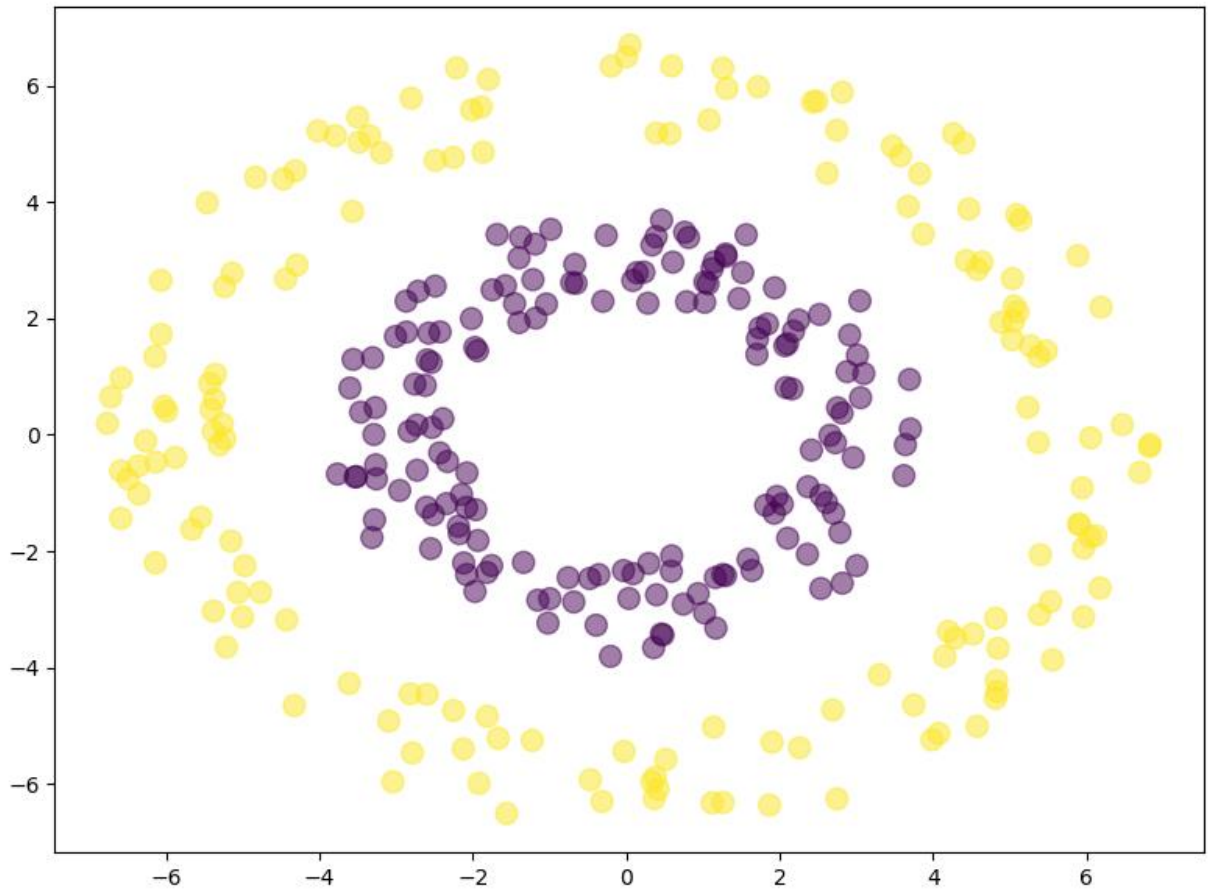


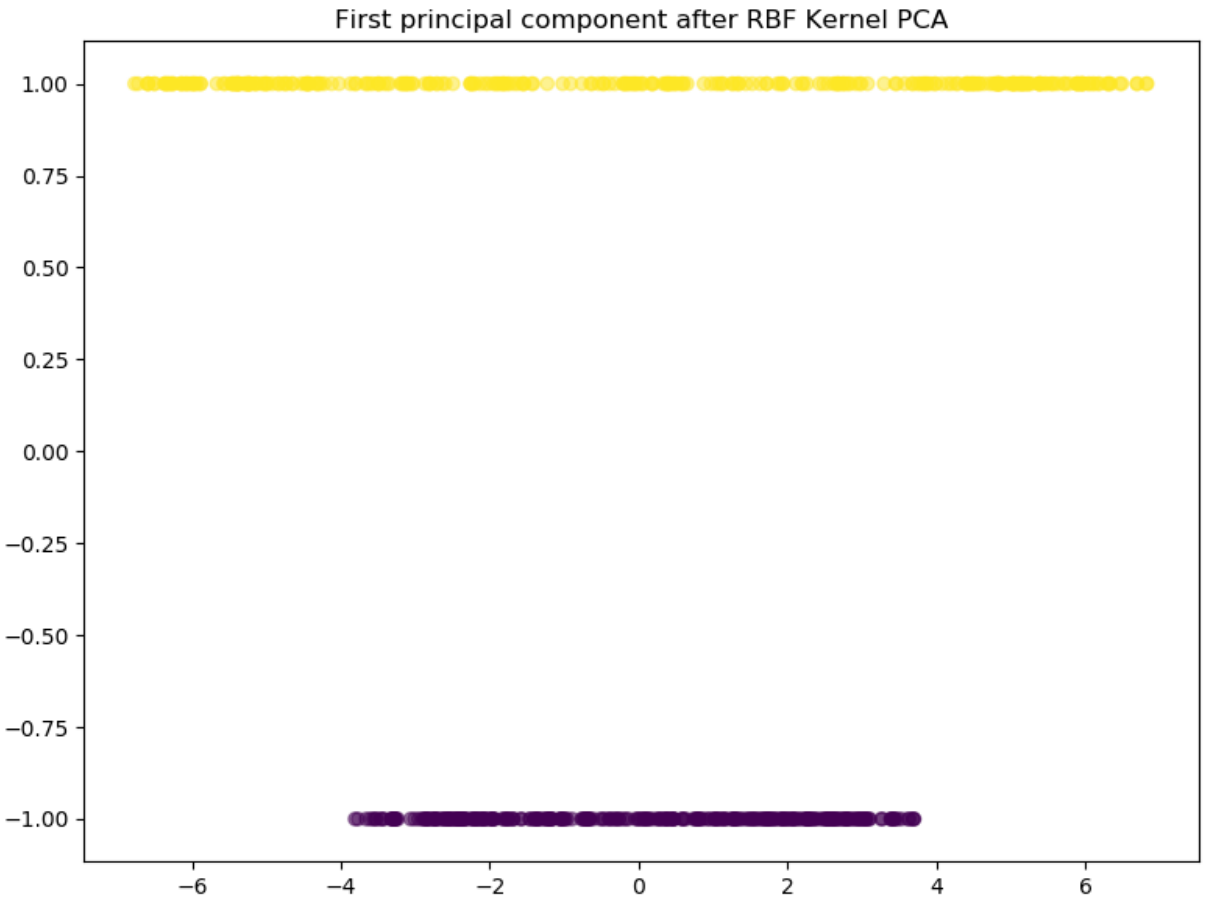

0.0130
0.0128
0.0126
0.0124
0.0122
0.0120
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9




```
the best C is 0.1 and the accuracy of it is 0.9879629629629629
```

قسمت ۲)


6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6




```
('Accuracy_transformed_data:', 0.55)
('Precision_transformed_data:', 0.5909090909090909)
('Recall_transformed_data:', 0.325)
('Accuracy:', 0.55)
('Precision:', 0.5909090909090909)
('Recall:', 0.325)
```