红酒数据集频繁模式与关联规则挖掘

姓名：范啸猛

学号：3220200870

1. 数据集预处理

红酒数据集中有11种属性，其中description，Unnamed: 0，designation与winery为unique的属性无法进行频繁模式挖掘，因此需要将上述属性删除。而国家，省份，区域存在较大的冗余与常识关系，因此在这里我们选取US国家作为研究对象，只保留省份属性，将非US国家的信息删除，并将属性'region_2', 'winery'删除。具体步骤如下：
首先，我们将无法进行频繁模式挖掘的列删除，即将'Unnamed: 0', 'description', 'designation', 'region_1', 'region_2', 'winery'列删除。处理数据的代码如下：

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import mode
import sklearn.ensemble

data_name = r'C:\Users\Xiaomeng Fan\Desktop\研究生的各种作业\数据挖掘\作业四\wine_1\winemag-data_first150k.csv'
file = pd.read_csv(data_name)
data = pd.DataFrame(file)

data = data.dropna()
data = data.drop(['Unnamed: 0', 'description', 'designation', 'region_1', 'region_2', 'winery'], axis=1)
data_US = data.drop(index=(data.loc[(data['country']!='US')].index))
```

接下来，我们统计省份属性的词频，代码及结果如下：

```
counts = data_US['province'].value_counts()
print(counts)
```

```
California    28557
Washington     6122
Oregon         3037
New York       1525
Name: province, dtype: int64
```

可以看出，California，Washington，Oregon，New York出现次数远远大于其他省份的出现次数，因此我们将除却这四个城市的其他城市替换为others。代码如下，

```
data_US.loc[(data_US['province']!= 'California') & (data_US['province']!= 'Washington') &
(data_US['province']!= 'Oregon') & (data_US['province']!= 'New York'), 'province'] = 'Other province'
```

```
counts = data_US['province'].value_counts()
print(counts)
```

```
California    28557
Washington     6122
Oregon         3037
New York       1525
Name: province, dtype: int64
```

按照上述依次对variety属性进行处理，代码如下：

```
counts = data_US['variety'].value_counts()
print(counts)
```

```
Pinot Noir             7442
Chardonnay             4892
Cabernet Sauvignon     4889
Red Blend              2934
Zinfandel              2814
                       ...
Viognier-Chardonnay       1
Picpoul                   1
Grenache-Mourvèdre        1
Seyval Blanc              1
Pinot Auxerrois           1
Name: variety, Length: 180, dtype: int64
```

```
data_US.loc[(data_US['variety']!= 'Pinot Noir') & (data_US['variety']!= 'Cabernet Sauvignon') &
(data_US['variety']!= 'Chardonnay') & (data_US['variety']!= 'Syrah') & (data_US['variety']!= 'Zinfandel'),
'variety'] = 'Other variety'
```

```
counts = data_US['variety'].value_counts()
print(counts)
```

```
Other variety         16413
Pinot Noir             7442
Chardonnay             4892
Cabernet Sauvignon     4889
Zinfandel              2814
Syrah                  2791
Name: variety, dtype: int64
```

下面，我们对数值属性进行处理。在这里，我们使用四分数，中位数，以及75%分位数将数据划分为1，2，3，4类。代码如下，

由于points与price是数值属性，我们首先计算US的points与price的五数概括，代码及结果如下：

```
data_US.describe()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	points	price
count	39241.000000	39241.000000
mean	88.359981	37.546316
std	3.438542	26.716547
min	80.000000	4.000000
25%	86.000000	22.000000
50%	88.000000	32.000000
75%	91.000000	45.000000
max	100.000000	2013.000000

注意：在这里我们将小于price的四分数的price值分箱为1，将介于四分位数与中位数之间的price值分箱为2，将介于中位数与四分之三位数之间的price值分箱为3，将大于四分之三分数的值分箱为4.代码如下：

```
data_US.loc[data_US['price']<19,'price']=1
data_US.loc[(data_US['price']>=19) & (data_US['price']<28), 'price' ]=2
data_US.loc[(data_US['price']>=28) & (data_US['price']<41), 'price' ]=3
data_US.loc[(data_US['price']>=41), 'price'] = 4
```

```
counts = data_US['price'].value_counts()
print(counts)
```

```
4.0    12594
3.0    11593
2.0     8209
1.0     6845
Name: price, dtype: int64
```

注意：在这里我们将小于points的四分数的price值分箱为10，将介于四分位数与中位数之间的points值分箱为20，将介于中位数与四分之三位数之间的points值分箱为30，将大于四分之三分数的points值分箱为40.代码如下：

```
data_US.loc[data_US['points']<85,'points']=10
data_US.loc[(data_US['points']>=85) & (data_US['points']<88), 'points' ]=20
data_US.loc[(data_US['points']>=88) & (data_US['points']<90), 'points' ]=30
data_US.loc[(data_US['points']>=90), 'points'] = 40
```

```
counts = data_US['points'].value_counts()
print(counts)
```

```
40    15465
20    10903
30     7110
10     5763
Name: points, dtype: int64
```

```
data_US_copy = data_US.copy(deep = False)
```

```
data_US = data_US.drop(['country'], axis=1)
```

```
data_US_copy = data_US_copy.drop(['country'], axis=1)
```

```
data_US_copy['price'] = data_US_copy['price'].apply(str)
data_US_copy['points'] = data_US_copy['points'].apply(str)
```

我们将处理之后的表格数据转换为字符串类型，代码如下：

```
all_transactions = np.array(data_US_copy)
```

```
all_transactions = all_transactions.tolist()
```

我们将转换为字符串类型的数据转换为可以进行频繁模式挖掘的数据类型，代码如下：

```
from prettytable import PrettyTable
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules, fpgrowth

trans_encoder = TransactionEncoder() # Instanciate the encoder
trans_encoder_matrix = trans_encoder.fit(all_transactions).transform(all_transactions)
trans_encoder_matrix = pd.DataFrame(trans_encoder_matrix, columns=trans_encoder.columns_)
```

```
trans_encoder_matrix.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	1.0	10	2.0	20	3.0	30	4.0	40	Cabernet Sauvignon	California	Chardonnay	New York	Oregon	Other variety	Pinot Noir
0	False	False	False	False	False	False	True	True	True	True	False	False	False	False	False
1	False	False	False	False	False	False	True	True	False	True	False	False	False	True	False
2	False	False	False	False	False	False	True	True	False	False	False	False	True	False	True
3	False	False	False	False	False	False	True	True	False	False	False	False	True	False	True
4	False	False	False	False	False	False	True	True	False	True	False	False	False	False	True

2.使用FP g.rowth方法对数据集进行挖掘

```
import time
def perform_rule_calculation(transact_items_matrix, rule_type="fpgrowth", min_support=0.001):
    """
    desc: this function performs the association rule calculation
    @params:
        - transact_items_matrix: the transaction X Items matrix
```

```
        - rule_type:
                    - apriori or Growth algorithms (default="fpgrowth")

        - min_support: minimum support threshold value (default = 0.001)

    @returns:
        - the matrix containing 3 columns:
            - support: support values for each combination of items
            - itemsets: the combination of items
            - number_of_items: the number of items in each combination of items

        - the excution time for the corresponding algorithm

    """
    start_time = 0
    total_execution = 0

    if(not rule_type=="fpgrowth"):
        start_time = time.time()
        rule_items = apriori(transact_items_matrix,
                       min_support=min_support,
                       use_colnames=True)
        total_execution = time.time() - start_time
        print("Computed Apriori!")

    else:
        start_time = time.time()
        rule_items = fpgrowth(transact_items_matrix,
                       min_support=min_support,
                       use_colnames=True)
        total_execution = time.time() - start_time
        print("Computed Fp Growth!")

    rule_items['number_of_items'] = rule_items['itemsets'].apply(lambda x: len(x))

    return rule_items, total_execution


def compute_association_rule(rule_matrix, metric="lift", min_thresh=1):
    """
    @desc: Compute the final association rule
    @params:
        - rule_matrix: the corresponding algorithms matrix
        - metric: the metric to be used (default is lift)
        - min_thresh: the minimum threshold (default is 1)

    @returns:
        - rules: all the information for each transaction satisfying the given metric & threshold
    """
    rules = association_rules(rule_matrix,
                              metric=metric,
                              min_threshold=min_thresh)

    return rules
```

```
def plot_metrics_relationship(rule_matrix, col1, col2):
    """
    desc: shows the relationship between the two input columns
    @params:
        - rule_matrix: the matrix containing the result of a rule (apriori or Fp Growth)
        - col1: first column
        - col2: second column
    """
    fit = np.polyfit(rule_matrix[col1], rule_matrix[col2], 1)
    fit_funt = np.poly1d(fit)
    plt.plot(rule_matrix[col1], rule_matrix[col2], 'yo', rule_matrix[col1],
    fit_funt(rule_matrix[col1]))
    plt.xlabel(col1)
    plt.ylabel(col2)
    plt.title('{} vs {}'.format(col1, col2))
```

```
def compare_time_exec(algo1=list, alg2=list):
    """
    @desc: shows the execution time between two algorithms
    @params:
        - algo1: list containing the description of first algorithm, where

        - algo2: list containing the description of second algorithm, where
    """

    execution_times = [algo1[1], algo2[1]]
    algo_names = (algo1[0], algo2[0])
    y=np.arange(len(algo_names))

    plt.bar(y,execution_times,color=['orange', 'blue'])
```

```
    plt.xticks(y,algo_names)
    plt.xlabel('Algorithms')
    plt.ylabel('Time')
    plt.title("Execution Time (seconds) Comparison")
    plt.show()
```

计算数据集的频繁模式

```
fpgrowth_matrix, fp_growth_exec_time = perform_rule_calculation(trans_encoder_matrix) # Run the algorithm
print("Fp Growth execution took: {} seconds".format(fp_growth_exec_time))
```

```
Computed Fp Growth!
Fp Growth execution took: 0.19874858856201172 seconds
```

```
fpgrowth_matrix.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	support	itemsets	number_of_items
0	0.727734	(California)	1
1	0.394103	(40)	1
2	0.320940	(4.0)	1
3	0.124589	(Cabernet Sauvignon)	1
4	0.418262	(Other variety)	1

```
fpgrowth_matrix_choose = fpgrowth_matrix[(fpgrowth_matrix['number_of_items']>=2) &(fpgrowth_matrix['support']>=0.1) ]
print(fpgrowth_matrix_choose)
```

```
      support                       itemsets  number_of_items
18   0.284167                (40, California)                2
19   0.126806              (40, Other variety)               2
21   0.253791                (4.0, California)               2
22   0.213195                        (40, 4.0)               2
24   0.164343            (40, 4.0, California)               3
28   0.102954  (Cabernet Sauvignon, California)               2
89   0.267705       (Other variety, California)              2
90   0.117097                (4.0, Pinot Noir)               2
91   0.100278                 (40, Pinot Noir)               2
92   0.138223          (California, Pinot Noir)              2
183  0.101730          (Chardonnay, California)              2
239  0.142428                (2.0, California)               2
240  0.116995             (2.0, Other variety)               2
252  0.103005       (Washington, Other variety)              2
348  0.216126                (3.0, California)               2
349  0.113810                        (40, 3.0)               2
350  0.112255             (3.0, Other variety)               2
400  0.207156                 (California, 20)               2
402  0.131877              (Other variety, 20)               2
411  0.115390                (1.0, California)               2
413  0.111491             (1.0, Other variety)               2
478  0.111032                 (30, California)               2
501  0.125379                 (California, 10)               2
```

```
fpgrowth_matrix.tail()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	support	itemsets	number_of_items
533	0.004281	(3.0, 10, Pinot Noir)	3
534	0.002268	(2.0, California, 10, Pinot Noir)	4
535	0.003364	(4.0, California, 10, Pinot Noir)	4
536	0.001605	(1.0, California, 10, Pinot Noir)	4
537	0.003491	(3.0, California, 10, Pinot Noir)	4

计算关联规则及评价指标

```
fp_growth_rule_lift = compute_association_rule(fpgrowth_matrix)
```

lift指标

下面，我们将对指标lift进行分析，首先将得到的规则按照conviction进行排序，代码如下，

```
fp_growth_rule_lift_sort = fp_growth_rule_lift.sort_values('lift',axis = 0,ascending = False)
```

```
fp_growth_rule_lift_sort.head(20)
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
589	(2.0, Pinot Noir)	(Oregon, 20)	0.015545	0.017329	0.002599	0.167213	9.649426	0.002330	1.179979
590	(Oregon, 20)	(2.0, Pinot Noir)	0.017329	0.015545	0.002599	0.150000	9.649426	0.002330	1.158182
1571	(Chardonnay, 1.0)	(New York, 10)	0.025229	0.007849	0.001656	0.065657	8.365030	0.001458	1.061870
1570	(New York, 10)	(Chardonnay, 1.0)	0.007849	0.025229	0.001656	0.211039	8.365030	0.001458	1.235513
449	(Oregon, 3.0)	(30, Pinot Noir)	0.022400	0.033944	0.005428	0.242321	7.138822	0.004668	1.275020
448	(30, Pinot Noir)	(Oregon, 3.0)	0.033944	0.022400	0.005428	0.159910	7.138822	0.004668	1.163685
444	(30, 3.0, Pinot Noir)	(Oregon)	0.011875	0.077394	0.005428	0.457082	5.905939	0.004509	1.699346
453	(Oregon)	(30, 3.0, Pinot Noir)	0.077394	0.011875	0.005428	0.070135	5.905939	0.004509	1.062654
379	(Oregon, 4.0)	(40, Pinot Noir)	0.031651	0.100278	0.018705	0.590982	5.893453	0.015531	2.199715
382	(40, Pinot Noir)	(Oregon, 4.0)	0.100278	0.031651	0.018705	0.186531	5.893453	0.015531	1.190395
566	(2.0, 30, Pinot Noir)	(Oregon)	0.003109	0.077394	0.001402	0.450820	5.825030	0.001161	1.679970
571	(Oregon)	(2.0, 30, Pinot Noir)	0.077394	0.003109	0.001402	0.018110	5.825030	0.001161	1.015278
463	(Oregon, 4.0)	(30, Pinot Noir)	0.031651	0.033944	0.006116	0.193237	5.692794	0.005042	1.197447
462	(30, Pinot Noir)	(Oregon, 4.0)	0.033944	0.031651	0.006116	0.180180	5.692794	0.005042	1.181173
1572	(Chardonnay, 10)	(New York, 1.0)	0.015570	0.018807	0.001656	0.106383	5.656605	0.001364	1.098002
1569	(New York, 1.0)	(Chardonnay, 10)	0.018807	0.015570	0.001656	0.088076	5.656605	0.001364	1.079508
446	(30, Oregon)	(3.0, Pinot Noir)	0.019546	0.050024	0.005428	0.277705	5.551419	0.004450	1.315219
451	(3.0, Pinot Noir)	(30, Oregon)	0.050024	0.019546	0.005428	0.108507	5.551419	0.004450	1.099789
1567	(10, Chardonnay, 1.0)	(New York)	0.008002	0.038862	0.001656	0.207006	5.326647	0.001345	1.212037
1574	(New York)	(10, Chardonnay, 1.0)	0.038862	0.008002	0.001656	0.042623	5.326647	0.001345	1.036162

通过将lift分数从小到大进行排序，我们可以得到看到上面显示的20条关联规则的lift值很高，代表我们可能找到了有用的关联规则。下面对这个20条关联规则进行分析。通过上述表格，我们可以清楚的看到，葡萄 Pinot Noir与省份Oregon存在非常强的关联关系。而 Pinot Noir（黑皮诺）恰恰只生长在美国的Oregon（俄亥冈州）。除此之外，我们还可以看到对于由Pinot Noir生产出来的葡萄酒价格与分数有着较强的关联，即价格越高分数越高，反之亦然。

conviction指标

下面，我们将对指标conviction进行分析，首先将得到的规则按照conviction进行排序，代码如下，

```
fp_growth_rule_conviction_sort = fp_growth_rule_lift.sort_values('conviction',axis = 0,ascending = False)
fp_growth_rule_conviction_sort.head(20)
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
1904	(30, 1.0, Zinfandel)	(California)	0.001249	0.727734	0.001249	1.000000	1.374129	0.000340	inf
2043	(1.0, Zinfandel, 10)	(California)	0.005785	0.727734	0.005785	1.000000	1.374129	0.001575	inf
1406	(Syrah, 4.0, 10)	(California)	0.001402	0.727734	0.001402	1.000000	1.374129	0.000382	inf
1840	(Zinfandel, 40, 4.0)	(California)	0.003670	0.727734	0.003670	1.000000	1.374129	0.000999	inf
1846	(30, 4.0, Zinfandel)	(California)	0.001325	0.727734	0.001325	1.000000	1.374129	0.000361	inf
1983	(3.0, Zinfandel, 10)	(California)	0.006065	0.727734	0.006040	0.995798	1.368355	0.001626	64.799368
1964	(Zinfandel, 10)	(California)	0.019444	0.727734	0.019342	0.994758	1.366925	0.005192	51.934788
1832	(4.0, Zinfandel)	(California)	0.008129	0.727734	0.008078	0.993730	1.365514	0.002162	43.426467
1996	(2.0, Zinfandel, 10)	(California)	0.006371	0.727734	0.006320	0.992000	1.363136	0.001684	34.033282
259	(4.0, Cabernet Sauvignon, 10)	(California)	0.003134	0.727734	0.003109	0.991870	1.362957	0.000828	33.488749
2008	(1.0, Zinfandel)	(California)	0.013685	0.727734	0.013557	0.990689	1.361334	0.003598	29.241395
1882	(30, 3.0, Zinfandel)	(California)	0.004613	0.727734	0.004562	0.988950	1.358945	0.001205	24.640096
2029	(1.0, Zinfandel, 20)	(California)	0.005861	0.727734	0.005785	0.986957	1.356206	0.001519	20.873746
1850	(4.0, Zinfandel, 20)	(California)	0.001911	0.727734	0.001886	0.986667	1.355807	0.000495	20.419969
456	(30, Oregon, 4.0)	(Pinot Noir)	0.006422	0.189649	0.006116	0.952381	5.021820	0.004898	17.017380
1864	(3.0, Zinfandel)	(California)	0.029535	0.727734	0.028924	0.979292	1.345674	0.007430	13.148191
1854	(4.0, Zinfandel, 10)	(California)	0.001223	0.727734	0.001198	0.979167	1.345501	0.000308	13.068780
1925	(3.0, Zinfandel, 20)	(California)	0.008613	0.727734	0.008410	0.976331	1.341605	0.002141	11.503249
1788	(Zinfandel)	(California)	0.071711	0.727734	0.070003	0.976190	1.341412	0.017817	11.435183
1912	(Zinfandel, 20)	(California)	0.022655	0.727734	0.022069	0.974128	1.338578	0.005582	10.523683

通过上述表格，我们可以看到，Zinfandel与California存在着较强的相关关系，并且，可以注意到Zinfandel是自变量，California是因变量。其中原因为Zinfandel在美国地区均是由California地区所生产的。

leverage指标

下面，我们将对指标leverage进行分析，首先将得到的规则按照leverage进行排序，代码如下，

```
fp_growth_rule_conviction_sort = fp_growth_rule_lift.sort_values('leverage',axis = 0,ascending = False)
fp_growth_rule_conviction_sort.head(20)
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
2	(40)	(4.0)	0.394103	0.320940	0.213195	0.540963	1.685560	0.086712	1.479316
3	(4.0)	(40)	0.320940	0.394103	0.213195	0.664285	1.685560	0.086712	1.804793
5	(40, California)	(4.0)	0.284167	0.320940	0.164343	0.578334	1.802001	0.073143	1.610421
8	(4.0)	(40, California)	0.320940	0.284167	0.164343	0.512069	1.802001	0.073143	1.467079
6	(4.0, California)	(40)	0.253791	0.394103	0.164343	0.647555	1.643111	0.064324	1.719125
7	(40)	(4.0, California)	0.394103	0.253791	0.164343	0.417006	1.643111	0.064324	1.279961
272	(4.0)	(Pinot Noir)	0.320940	0.189649	0.117097	0.364856	1.923855	0.056231	1.275855
273	(Pinot Noir)	(4.0)	0.189649	0.320940	0.117097	0.617442	1.923855	0.056231	1.775049
288	(4.0)	(40, Pinot Noir)	0.320940	0.100278	0.078617	0.244958	2.442794	0.046434	1.191619
285	(40, Pinot Noir)	(4.0)	0.100278	0.320940	0.078617	0.783990	2.442794	0.046434	3.143649
280	(California, Pinot Noir)	(4.0)	0.138223	0.320940	0.088963	0.643621	2.005426	0.044602	1.905444
281	(4.0)	(California, Pinot Noir)	0.320940	0.138223	0.088963	0.277195	2.005426	0.044602	1.192269
283	(Pinot Noir)	(4.0, California)	0.189649	0.253791	0.088963	0.469094	1.848351	0.040832	1.405540
278	(4.0, California)	(Pinot Noir)	0.253791	0.189649	0.088963	0.350537	1.848351	0.040832	1.247726
1733	(Other variety)	(1.0)	0.418262	0.174435	0.111491	0.266557	1.528117	0.038531	1.125602
1732	(1.0)	(Other variety)	0.174435	0.418262	0.111491	0.639153	1.528117	0.038531	1.612146
284	(40, 4.0)	(Pinot Noir)	0.213195	0.189649	0.078617	0.368754	1.944409	0.038185	1.283734
289	(Pinot Noir)	(40, 4.0)	0.189649	0.213195	0.078617	0.414539	1.944409	0.038185	1.343906
956	(Washington)	(Other variety)	0.156010	0.418262	0.103005	0.660242	1.578538	0.037751	1.712213
957	(Other variety)	(Washington)	0.418262	0.156010	0.103005	0.246268	1.578538	0.037751	1.119748

我们可以看到，与上述两个指标不同。通过对leverage指标进行分析，可以得到以下的结论：

- 价格与分数存在着较强的关联关系，并且，当价格很高时，得分通常也比较高
- 由Pinot Noir制成的葡萄酒价格与得分往往很高，其主要原因为Pinot Noir同样也被看作是天性娇弱的贵族葡萄品种。它早熟、皮薄、色素低、产量少, 适合较寒冷地区, 然而发芽早的特点又使其对霜冻极为敏感, 皮薄使它对强光的耐受性差, 也使它对病菌及潮湿的抵抗力差, 易发生腐烂, 造成大量减产

下面，我们将对置信度进行分析，代码如下：

```
fp_growth_rule = compute_association_rule(fpgrowth_matrix, metric="confidence", min_thresh=0.2)
fp_growth_rule.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(40)	(California)	0.394103	0.727734	0.284167	0.721048	0.990812	-0.002635	0.976031
1	(California)	(40)	0.727734	0.394103	0.284167	0.390482	0.990812	-0.002635	0.994059
2	(40)	(Other variety)	0.394103	0.418262	0.126806	0.321759	0.769277	-0.038032	0.857716
3	(Other variety)	(40)	0.418262	0.394103	0.126806	0.303174	0.769277	-0.038032	0.869510
4	(40, Other variety)	(California)	0.126806	0.727734	0.074463	0.587219	0.806914	-0.017818	0.659590

通过上述表格，我们可以看到California与最高的酒的得分数存在关联规则。

分析

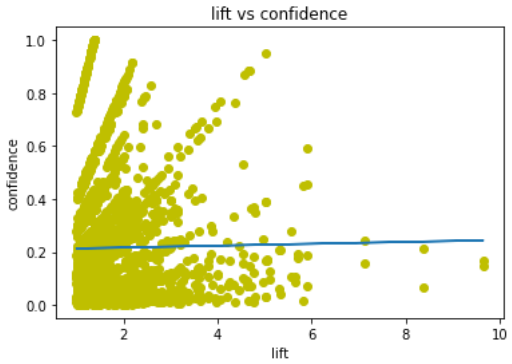
通过使用FP growth方法对关联规则进行挖掘，我们可以得到下面的结论：

- 葡萄 Pinot Noir与省份Oregon存在非常强的关联关系。而 Pinot Noir（黑皮诺）恰恰只生长在美国的Oregon（俄亥冈州）。
- 由Pinot Noir生产出来的葡萄酒价格与分数有着较强的关联，即价格越高分数越高，反之亦然。
- Zinfandel与California存在着较强的相关关系，并且，可以注意到Zinfandel是自变量，California是因变量。其中原因为Zinfandel在美国地区均是由California地区所生产的。
- 价格与分数存在着较强的关联关系，并且，当价格很高时，得分通常也比较高
- 由Pinot Noir制成的葡萄酒价格与得分往往很高，其主要原因为Pinot Noir同样也被看作是天性娇弱的贵族葡萄品种。它早熟、皮薄、色素低、产量少, 适合较寒冷地区, 然而发芽早的特点又使其对霜冻极为敏感, 皮薄使它对强光的耐受性差, 也使它对病菌及潮湿的抵抗力差, 易发生腐烂, 造成大量减产。
- California与最高的酒的得分数存在关联规则。

可视化

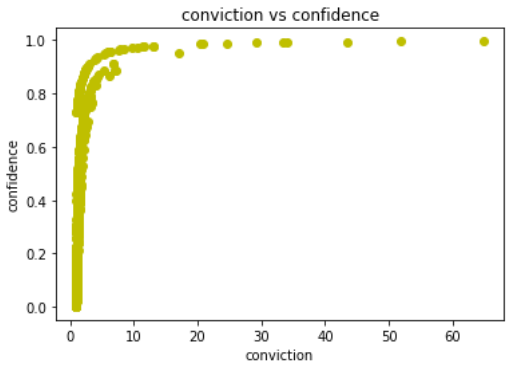
下面，我们将绘制三个指标与置信度之间关系的图像，

```
plot_metrics_relationship(fp_growth_rule_lift, col1='lift', col2='confidence')
```



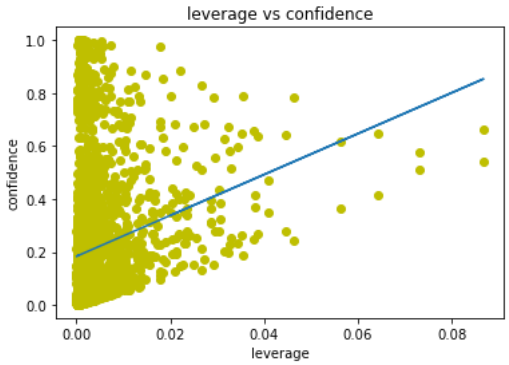

```
def plot_metrics_relationship_1(rule_matrix, col1, col2):
    """
    desc: shows the relationship between the two input columns
    @params:
        - rule_matrix: the matrix containing the result of a rule (apriori or Fp Growth)
        - col1: first column
        - col2: second column
    """
#     fit = np.polyfit(rule_matrix[col1], rule_matrix[col2], 1)
#     fit_funt = np.poly1d(fit)
    plt.plot(rule_matrix[col1], rule_matrix[col2], 'yo' )
#     fit_funt(rule_matrix[col1]))
    plt.xlabel(col1)
    plt.ylabel(col2)
    plt.title('{} vs {}'.format(col1, col2))

plot_metrics_relationship_1(fp_growth_rule_lift, col1='conviction', col2='confidence')
```



我们可以看到，指标conviction与confidence存在较强的相关关系，即conviction越大，confidence越大。

```
plot_metrics_relationship(fp_growth_rule_lift, col1='leverage', col2='confidence')
```



3.使用Apriori 方法对数据集进行挖掘

```
apriori_matrix, apriori_exec_time = perform_rule_calculation(trans_encoder_matrix, rule_type="apriori")
```

```
Computed Apriori!
```

计算数据集的频繁模式

```
apriori_matrix.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	support	itemsets	number_of_items
0	0.174435	(1.0)	1
1	0.146862	(10)	1
2	0.209194	(2.0)	1
3	0.277847	(20)	1
4	0.295431	(3.0)	1

```
apriori_matrix.tail()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	support	itemsets	number_of_items
533	0.003670	(Zinfandel, 40, 4.0, California)	4
534	0.001019	(Oregon, 40, Chardonnay, 4.0)	4
535	0.018705	(Oregon, 40, 4.0, Pinot Noir)	4
536	0.014831	(Washington, 40, Other variety, 4.0)	4
537	0.005632	(Syrah, Washington, 40, 4.0)	4

计算关联规则及评价指标

```
apriori_rule_lift = compute_association_rule(apriori_matrix)
```

lift指标

下面，我们将对指标lift进行分析，首先将得到的规则按照lift进行排序，代码如下，

```
apriori_rule_lift_sort = apriori_rule_lift.sort_values('lift',axis = 0,ascending = False)
```

```
apriori_rule_lift_sort.head(20)
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
1538	(Oregon, 20)	(2.0, Pinot Noir)	0.017329	0.015545	0.002599	0.150000	9.649426	0.002330	1.158182
1537	(2.0, Pinot Noir)	(Oregon, 20)	0.015545	0.017329	0.002599	0.167213	9.649426	0.002330	1.179979
1000	(New York, 10)	(Chardonnay, 1.0)	0.007849	0.025229	0.001656	0.211039	8.365030	0.001458	1.235513
1001	(Chardonnay, 1.0)	(New York, 10)	0.025229	0.007849	0.001656	0.065657	8.365030	0.001458	1.061870
1955	(Oregon, 3.0)	(30, Pinot Noir)	0.022400	0.033944	0.005428	0.242321	7.138822	0.004668	1.275020
1954	(30, Pinot Noir)	(Oregon, 3.0)	0.033944	0.022400	0.005428	0.159910	7.138822	0.004668	1.163685
1959	(Oregon)	(30, 3.0, Pinot Noir)	0.077394	0.011875	0.005428	0.070135	5.905939	0.004509	1.062654
1950	(30, 3.0, Pinot Noir)	(Oregon)	0.011875	0.077394	0.005428	0.457082	5.905939	0.004509	1.699346
2239	(Oregon, 4.0)	(40, Pinot Noir)	0.031651	0.100278	0.018705	0.590982	5.893453	0.015531	2.199715
2242	(40, Pinot Noir)	(Oregon, 4.0)	0.100278	0.031651	0.018705	0.186531	5.893453	0.015531	1.190395
1624	(2.0, 30, Pinot Noir)	(Oregon)	0.003109	0.077394	0.001402	0.450820	5.825030	0.001161	1.679970
1629	(Oregon)	(2.0, 30, Pinot Noir)	0.077394	0.003109	0.001402	0.018110	5.825030	0.001161	1.015278
2146	(30, Pinot Noir)	(Oregon, 4.0)	0.033944	0.031651	0.006116	0.180180	5.692794	0.005042	1.181173
2147	(Oregon, 4.0)	(30, Pinot Noir)	0.031651	0.033944	0.006116	0.193237	5.692794	0.005042	1.197447
1002	(Chardonnay, 10)	(New York, 1.0)	0.015570	0.018807	0.001656	0.106383	5.656605	0.001364	1.098002
999	(New York, 1.0)	(Chardonnay, 10)	0.018807	0.015570	0.001656	0.088076	5.656605	0.001364	1.079508
1957	(3.0, Pinot Noir)	(30, Oregon)	0.050024	0.019546	0.005428	0.108507	5.551419	0.004450	1.099789
1952	(30, Oregon)	(3.0, Pinot Noir)	0.019546	0.050024	0.005428	0.277705	5.551419	0.004450	1.315219
997	(10, Chardonnay, 1.0)	(New York)	0.008002	0.038862	0.001656	0.207006	5.326647	0.001345	1.212037
1004	(New York)	(10, Chardonnay, 1.0)	0.038862	0.008002	0.001656	0.042623	5.326647	0.001345	1.036162

在这里，我们可以得到与FP growth相似的结论。

conviction指标

下面，我们将对指标lift进行分析，首先将得到的规则按照conviction进行排序，代码如下，

```
apriori_rule_lift_sort = apriori_rule_lift.sort_values('conviction',axis = 0,ascending = False)
```

```
apriori_rule_lift_sort.head(20)
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
1412	(Syrah, 4.0, 10)	(California)	0.001402	0.727734	0.001402	1.000000	1.374129	0.000382	inf
981	(1.0, Zinfandel, 10)	(California)	0.005785	0.727734	0.005785	1.000000	1.374129	0.001575	inf
2136	(30, 4.0, Zinfandel)	(California)	0.001325	0.727734	0.001325	1.000000	1.374129	0.000361	inf
1180	(30, 1.0, Zinfandel)	(California)	0.001249	0.727734	0.001249	1.000000	1.374129	0.000340	inf
2224	(4.0, 40, Zinfandel)	(California)	0.003670	0.727734	0.003670	1.000000	1.374129	0.000999	inf
1377	(3.0, Zinfandel, 10)	(California)	0.006065	0.727734	0.006040	0.995798	1.368355	0.001626	64.799368
369	(Zinfandel, 10)	(California)	0.019444	0.727734	0.019342	0.994758	1.366925	0.005192	51.934788
862	(4.0, Zinfandel)	(California)	0.008129	0.727734	0.008078	0.993730	1.365514	0.002162	43.426467
1313	(2.0, Zinfandel, 10)	(California)	0.006371	0.727734	0.006320	0.992000	1.363136	0.001684	34.033282
1395	(4.0, Cabernet Sauvignon, 10)	(California)	0.003134	0.727734	0.003109	0.991870	1.362957	0.000828	33.488749
244	(1.0, Zinfandel)	(California)	0.013685	0.727734	0.013557	0.990689	1.361334	0.003598	29.241395
1926	(30, 3.0, Zinfandel)	(California)	0.004613	0.727734	0.004562	0.988950	1.358945	0.001205	24.640096
1093	(1.0, Zinfandel, 20)	(California)	0.005861	0.727734	0.005785	0.986957	1.356206	0.001519	20.873746
1860	(4.0, Zinfandel, 20)	(California)	0.001911	0.727734	0.001886	0.986667	1.355807	0.000495	20.419969
2140	(30, Oregon, 4.0)	(Pinot Noir)	0.006422	0.189649	0.006116	0.952381	5.021820	0.004898	17.017380
710	(3.0, Zinfandel)	(California)	0.029535	0.727734	0.028924	0.979292	1.345674	0.007430	13.148191
1420	(4.0, Zinfandel, 10)	(California)	0.001223	0.727734	0.001198	0.979167	1.345501	0.000308	13.068780
1789	(3.0, Zinfandel, 20)	(California)	0.008613	0.727734	0.008410	0.976331	1.341605	0.002141	11.503249
100	(Zinfandel)	(California)	0.071711	0.727734	0.070003	0.976190	1.341412	0.017817	11.435183
597	(Zinfandel, 20)	(California)	0.022655	0.727734	0.022069	0.974128	1.338578	0.005582	10.523683

在这里，我们可以得到与FP growth相似的结论。

leverage指标

下面，我们将对指标lift进行分析，首先将得到的规则按照leverage进行排序，代码如下，

```
apriori_rule_lift_sort = apriori_rule_lift.sort_values('leverage',axis = 0,ascending = False)
```

```
apriori_rule_lift_sort.head(20)
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
71	(4.0)	(40)	0.320940	0.394103	0.213195	0.664285	1.685560	0.086712	1.804793
70	(40)	(4.0)	0.394103	0.320940	0.213195	0.540963	1.685560	0.086712	1.479316
812	(4.0)	(40, California)	0.320940	0.284167	0.164343	0.512069	1.802001	0.073143	1.467079
809	(40, California)	(4.0)	0.284167	0.320940	0.164343	0.578334	1.802001	0.073143	1.610421
811	(40)	(4.0, California)	0.394103	0.253791	0.164343	0.417006	1.643111	0.064324	1.279961
810	(4.0, California)	(40)	0.253791	0.394103	0.164343	0.647555	1.643111	0.064324	1.719125
79	(Pinot Noir)	(4.0)	0.189649	0.320940	0.117097	0.617442	1.923855	0.056231	1.775049
78	(4.0)	(Pinot Noir)	0.320940	0.189649	0.117097	0.364856	1.923855	0.056231	1.275855
832	(4.0)	(40, Pinot Noir)	0.320940	0.100278	0.078617	0.244958	2.442794	0.046434	1.191619
829	(40, Pinot Noir)	(4.0)	0.100278	0.320940	0.078617	0.783990	2.442794	0.046434	3.143649
858	(California, Pinot Noir)	(4.0)	0.138223	0.320940	0.088963	0.643621	2.005426	0.044602	1.905444
859	(4.0)	(California, Pinot Noir)	0.320940	0.138223	0.088963	0.277195	2.005426	0.044602	1.192269
856	(4.0, California)	(Pinot Noir)	0.253791	0.189649	0.088963	0.350537	1.848351	0.040832	1.247726
861	(Pinot Noir)	(4.0, California)	0.189649	0.253791	0.088963	0.469094	1.848351	0.040832	1.405540
9	(Other variety)	(1.0)	0.418262	0.174435	0.111491	0.266557	1.528117	0.038531	1.125602
8	(1.0)	(Other variety)	0.174435	0.418262	0.111491	0.639153	1.528117	0.038531	1.612146
828	(40, 4.0)	(Pinot Noir)	0.213195	0.189649	0.078617	0.368754	1.944409	0.038185	1.283734
833	(Pinot Noir)	(40, 4.0)	0.189649	0.213195	0.078617	0.414539	1.944409	0.038185	1.343906
108	(Washington)	(Other variety)	0.156010	0.418262	0.103005	0.660242	1.578538	0.037751	1.712213
109	(Other variety)	(Washington)	0.418262	0.156010	0.103005	0.246268	1.578538	0.037751	1.119748

在这里，我们可以得到与FP growth相似的结论。下面，我们将绘制leverage与置信度之间关系的图像，代码如下：

分析

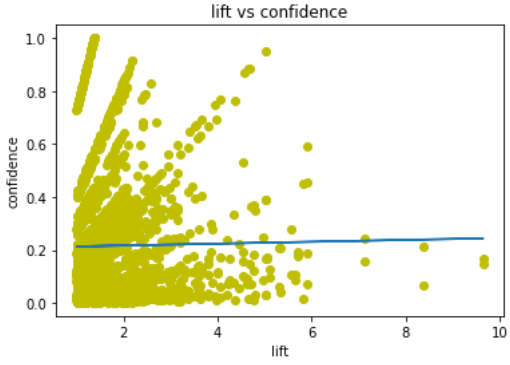
通过使用Apriori 方法对关联规则进行挖掘，我们可以得到下面的结论：

- 葡萄 Pinot Noir与省份Oregon存在非常强的关联关系。而 Pinot Noir（黑皮诺）恰恰只生长在美国的Oregon（俄亥冈州）。
- 由Pinot Noir生产出来的葡萄酒价格与分数有着较强的关联，即价格越高分数越高，反之亦然。
- Zinfandel与California存在着较强的相关关系，并且，可以注意到Zinfandel是自变量，California是因变量。其中原因为Zinfandel在美国地区均是由California地区所生产的。
- 价格与分数存在着较强的关联关系，并且，当价格很高时，得分通常也比较高
- 由Pinot Noir制成的葡萄酒价格与得分往往很高，其主要原因为Pinot Noir同样也被看作是天性娇弱的贵族葡萄品种。它早熟、皮薄、色素低、产量少, 适合较寒冷地区, 然而发芽早的特点又使其对霜冻极为敏感, 皮薄使它对强光的耐受性差, 也使它对病菌及潮湿的抵抗力差, 易发生腐烂, 造成大量减产。
- California与最高的酒的得分数存在关联规则。

可视化

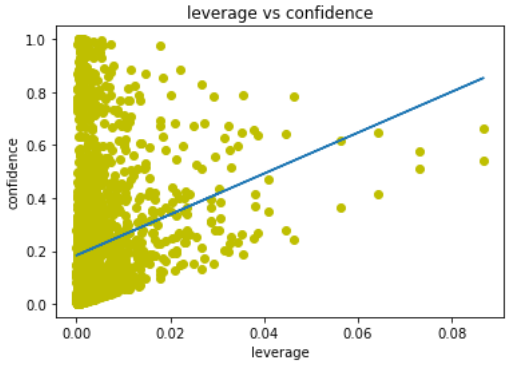
下面我们将绘制三个指标与置信度关系的图像，代码如下：

```
plot_metrics_relationship(apriori_rule_lift, col1='lift', col2='confidence')
```



```
plot_metrics_relationship_1(apriori_rule_lift, col1='conviction', col2='confidence')
```


```
plot_metrics_relationship(apriori_rule_lift, col1='leverage', col2='confidence')
```



代码地址

https://github.com/XiaomengFanmcislab/DataMining_2