数据挖掘互评作业二: 频繁模式与关联规则挖掘

一、读取数据集，并查看数据集的信息概要

In [1]:

```
import pandas as pds
data = pds.read_csv('./Wine Reviews/winemag-data-130k-v2.csv')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129971 entries, 0 to 129970
Data columns (total 14 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   id                     129971 non-null  int64
 1   country                129908 non-null  object
 2   description            129971 non-null  object
 3   designation            92506 non-null   object
 4   points                 129971 non-null  int64
 5   price                  120975 non-null  float64
 6   province               129908 non-null  object
 7   region_1               108724 non-null  object
 8   region_2               50511 non-null   object
 9   taster_name            103727 non-null  object
 10  taster_twitter_handle  98758 non-null   object
 11  title                  129971 non-null  object
 12  variety                129970 non-null  object
 13  winery                 129971 non-null  object
dtypes: float64(1), int64(2), object(11)
memory usage: 13.9+ MB
```

二、根据数据集的信息筛选出进行频繁模式挖掘的列‘

筛选依据为:

- id 列不具有实际意义
- description、title列完全不具有重复的数值
- price 列为浮点数，region_1 和 winery 列数值重复度低
- desgination、region_2和taster_twitter_handle 列包含太多的的缺失值

所以最终筛选出来的列为: `country, points, province, taster_name, variety`

In [2]:

```
# 仅取部分列进行频繁模式挖掘
to_preserve = ['country', 'points', 'province', 'taster_name', 'variety']
data_reduced = data[to_preserve].copy()
data_reduced.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129971 entries, 0 to 129970
Data columns (total 5 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   country      129908 non-null  object
 1   points       129971 non-null  int64
 2   province     129908 non-null  object
 3   taster_name  103727 non-null  object
 4   variety      129970 non-null  object
dtypes: int64(1), object(4)
memory usage: 5.0+ MB
```

三、对数据集进行处理，以便于进行关联规则挖掘

具体处理如下：

- 将数值属性转化为标称属性
- 将 DataFrame 格式的数据转化为 List 格式，并删除缺失值
- 使用预处理工具将数据编码成挖掘工具规定的形式

In [3]:
```
# 将数值转化为字符串
data_reduced['points'] = data_reduced['points'].map(str)
data_reduced.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129971 entries, 0 to 129970
Data columns (total 5 columns):
 #   Column       Non-Null Count   Dtype 
---  ------       --------------   ----- 
 0   country      129908 non-null  object
 1   points       129971 non-null  object
 2   province     129908 non-null  object
 3   taster_name  103727 non-null  object
 4   variety      129970 non-null  object
dtypes: object(5)
memory usage: 5.0+ MB
```

In [4]:
```
def row_to_list(row):
    return row.dropna().tolist()

data_reduced_list = data_reduced.apply(row_to_list, axis=1).tolist()
```

In [5]:
```
from mlxtend.preprocessing import TransactionEncoder

te = TransactionEncoder()
data_encoded = te.fit_transform(data_reduced_list)
df_encoded = pds.DataFrame(data_encoded, columns=te.columns_)
df_encoded.head(3)
```

Out[5]:

	100	80	81	82	83	84	85	86	87	88	...	Zierfandler-Rotgipfler	Zinfandel	Zlahtina
0	False	False	False	False	False	False	False	False	True	False	...	False	False	False
1	False	False	False	False	False	False	False	False	True	False	...	False	False	False
2	False	False	False	False	False	False	False	False	True	False	...	False	False	False

3 rows × 1184 columns

四、使用 mlxtend 工具包找出数据集中的频繁模式

In [6]:
```
from mlxtend.frequent_patterns import apriori

freq_itemsets = apriori(df_encoded, min_support=0.05, use_colnames=True)
freq_itemsets.sort_values(by='support', ascending=False, inplace=True)
```

In [7]:

```
print(freq_itemsets[freq_itemsets.itemsets.apply(lambda x: len(x)) >= 1])
```

```
     support                           itemsets
21  0.419355                               (US)
10  0.278885                       (California)
29  0.278885                   (California, US)
19  0.196305                       (Roger Voss)
12  0.169984                           (France)
13  0.150341                            (Italy)
32  0.143124               (France, Roger Voss)
3   0.132391                               (88)
2   0.130283                               (87)
5   0.118565                               (90)
15  0.116441                (Michael Schachner)
17  0.102115                       (Pinot Noir)
1   0.096945                               (86)
4   0.094067                               (89)
11  0.090428                       (Chardonnay)
6   0.087396                               (91)
14  0.082911                     (Kerin O’Keefe)
33  0.082911              (Italy, Kerin O’Keefe)
36  0.076055                   (US, Pinot Noir)
7   0.073963                               (92)
22  0.073378                   (Virginie Boone)
37  0.073378               (Virginie Boone, US)
40  0.073339   (Virginie Boone, California, US)
16  0.073339                     (Paul Gregutt)
30  0.073339       (Virginie Boone, California)
0   0.073324                               (85)
9   0.072878               (Cabernet Sauvignon)
35  0.071578                 (Paul Gregutt, US)
18  0.068831                        (Red Blend)
23  0.066469                       (Washington)
38  0.066469                   (US, Washington)
27  0.056282           (Cabernet Sauvignon, US)
25  0.054158                           (88, US)
8   0.053204         (Bordeaux-style Red Blend)
28  0.053058           (California, Pinot Noir)
39  0.053058       (California, Pinot Noir, US)
31  0.052327                   (Chardonnay, US)
26  0.051612                           (US, 90)
20  0.051127                            (Spain)
24  0.051027                           (US, 87)
34  0.050588         (Michael Schachner, Spain)
```

使用 mlxtend 工具包对发现的频繁模式进行关联规则挖掘

In [8]:
```
from mlxtend.frequent_patterns import association_rules

asso_rules = association_rules(freq_itemsets, metric='confidence', min_threshold=0.8
asso_rules.sort_values(by='lift', ascending=False, inplace=True)
```

关联规则挖掘的结果如下，其中第5、6列分别为支持度和置信度，第7、8、9列分别为Lift、Leverage和置信度，都是关联规则的评价指标。

In [9]:
```
print(asso_rules)
```

```
                antecedents          consequents  antecedent support  \
11                  (Spain)  (Michael Schachner)            0.051127
2           (Kerin O’Keefe)              (Italy)            0.082911
1                  (France)         (Roger Voss)            0.169984
5      (Virginie Boone, US)         (California)            0.073378
6          (Virginie Boone)     (California, US)            0.073378
```

```
7             (Virginie Boone)        (California)            0.073378
0                 (California)                (US)            0.278885
3             (Virginie Boone)                (US)            0.073378
4   (Virginie Boone, California)              (US)            0.073339
9                 (Washington)                (US)            0.066469
10     (California, Pinot Noir)               (US)            0.053058
8               (Paul Gregutt)                (US)            0.073339

    consequent support   support  confidence      lift  leverage   conviction
11            0.116441  0.050588    0.989466  8.497546  0.044635    83.874959
2             0.150341  0.082911    1.000000  6.651535  0.070446          inf
1             0.196305  0.143124    0.841986  4.289166  0.109755     5.086229
5             0.278885  0.073339    0.999476  3.583824  0.052875  1375.454198
6             0.278885  0.073339    0.999476  3.583824  0.052875  1375.454198
7             0.278885  0.073339    0.999476  3.583824  0.052875  1375.454198
0             0.419355  0.278885    1.000000  2.384614  0.161933          inf
3             0.419355  0.073378    1.000000  2.384614  0.042607          inf
4             0.419355  0.073339    1.000000  2.384614  0.042584          inf
9             0.419355  0.066469    1.000000  2.384614  0.038595          inf
10            0.419355  0.053058    1.000000  2.384614  0.030808          inf
8             0.419355  0.071578    0.975976  2.327325  0.040822    24.169028
```

上面展示的是置信度大于等于 0.8 的关联规则，且按照 Lift 值从大到小排序，其中全部都是葡萄酒出产的国家和地区的关联性，这也在意料之内，毕竟地区包含在国家之内，并且一个地区一般只在一个省份内，一个省份也只在一个国家内。