Association Rules

Association rule mining is a technique to identify underlying relations between different items. In this part, two different Python packages were selected to use the Apriori algorithm.

1. Basic idea

As the previous work shows, the flights delay could be regarded as “arrival delay” and “departure delay”. In order to find representative rules, the algorithm would be implemented to diverse subsets aimed at different kinds of delay. In this case, two subsets were set up, and their attributes were not all exactly the same.

1. Features and thresholds selection

There are two fundamental criteria for selecting the features of the subsets. First, the features should be essential and related with the machine learning part to keep the coherence. Second, the data type should be factor. The binning methods were used to process the numeric data. Finally, for arrival delay subset, the attributes are “aircrafttype”, “filed\_airspeed\_kts\_bin”, “destination”, ”actualarrivaltime\_week”, “airline” and “arr\_delay\_sig”. For departure delay subset, the attributes are “aircrafttype”, “filed\_airspeed\_kts\_bin”, “origin”, ”actualdeparturetime\_week”, “airline” and “dep\_delay\_sig”.

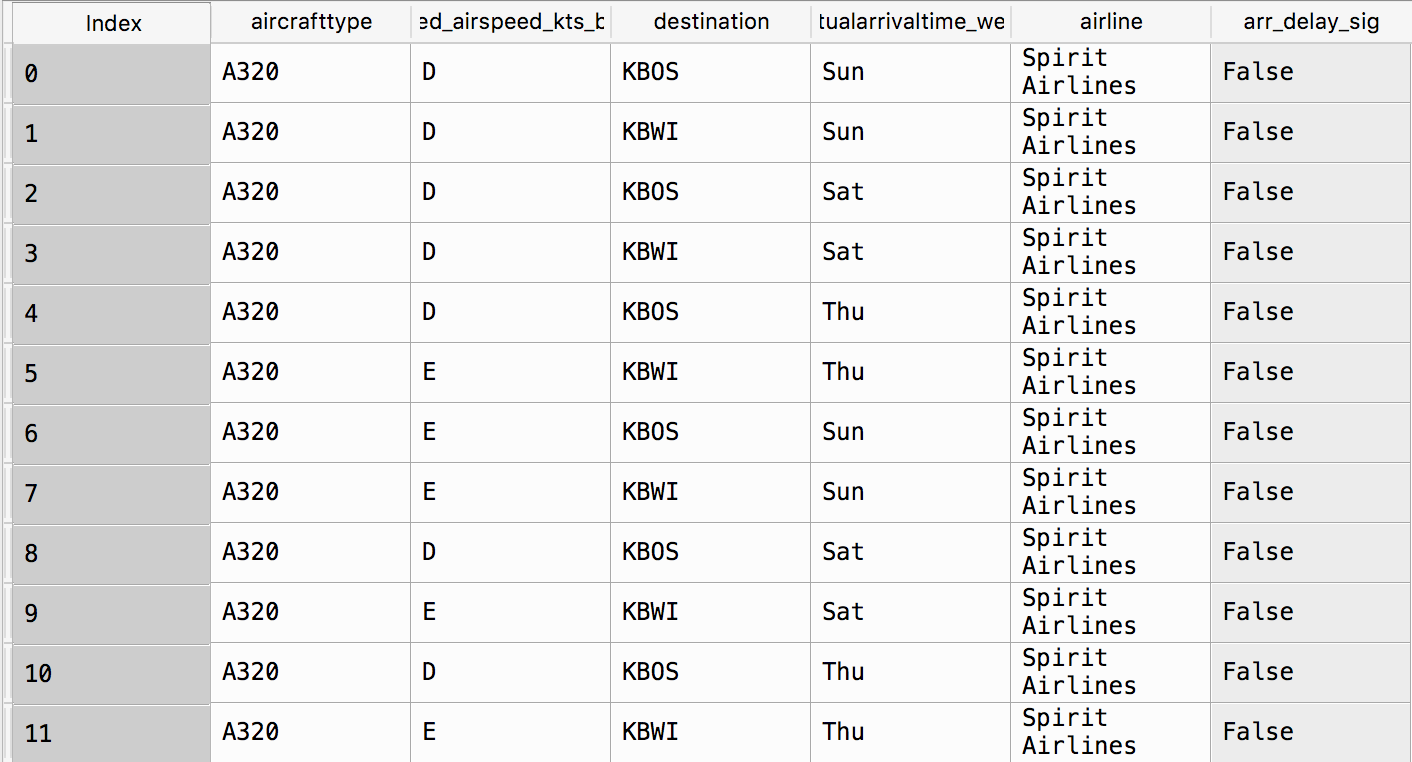
For thresholds, as required, at least three different support levels were chosen. They are:

|  |  |  |  |
| --- | --- | --- | --- |
| min\_support | min\_confidence | min\_lift | min\_length |
| 0.05 | 0.2 | 3 | 5 |
| 0.01 | 0.2 | 3 | 5 |
| 0.005 | 0.2 | 3 | 5 |

Also, when using mlxtend package to find the association rules, min\_support=0.002 is the new threshold.

1. Apyori Package

To do the association rules, the dataset needed to be changed to an appropriate format. To use the Apyori 1.1.1 package, the data was transformed from Figure 3.1 to Figure 3.2.

Figure 3.1

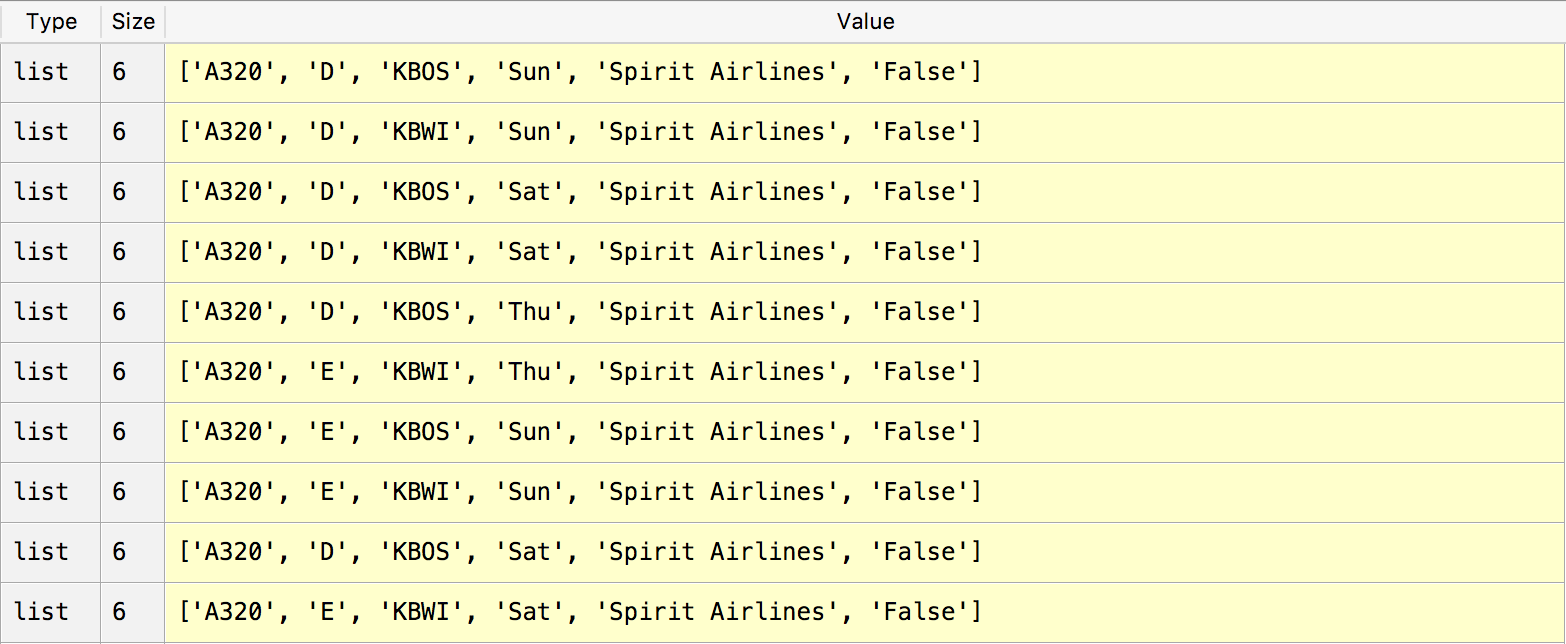


Figure 3.2

After that, apply the command “apriori(data, min\_support, min\_confidence, min\_lift, min\_len)”. The outcome is with the class generator, so use “list( )” to change the data type. The part of list with min\_support=0.01 in arrival delay subset shown below:

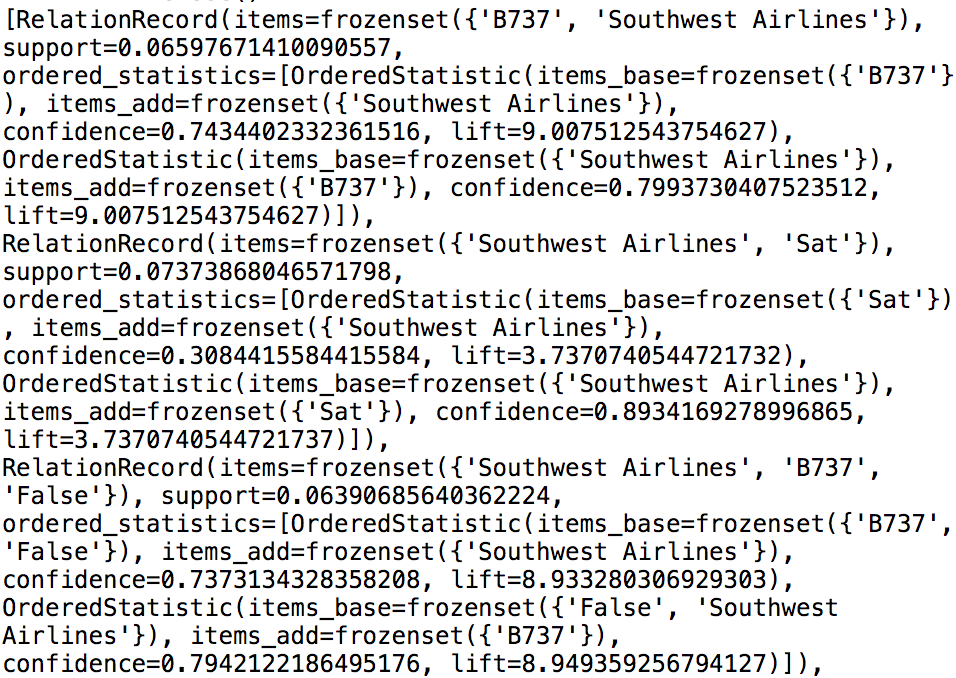


Figure 3.3

However, we can find that the “min\_length” attribute in the command is actually invalid, and the sizes of the rules were from 2 to 5. Also, the list is still bewildering, so the further step is necessary. After the several steps’ processing, six “.txt” files were wrote. The files names were like “A\_B.txt”, where “A” represent the subset’s type and “B” is the minimum support number. The basic format of these files is like:

Rule: ['False', 'Sat', 'KBWI', 'D'] -> Southwest Airlines

Support: 0.016597510373443983

Confidence: 0.6666666666666667

Lift: 3.3472222222222228

=====================================

Rule: ['KFLL', 'Spirit Airlines', 'False', 'Mon'] -> D

Support: 0.016597510373443983

Confidence: 1.0

Lift: 3.3943661971830985

=====================================

And for all files, the sizes of the rules follow the ascending order. Under this circumstance, to find the meaningful rules, it is efficient to search from the bottom of the text.

1. Mlxtend package

However, during the coding, there were several disadvantages of the “Apyori 1.1.1” package. For example, when setting a small min\_support, the speed of the code is quite slow. Also, the output is difficult to process, and it is hard to constrain the rules’ sizes and select the frequent patterns. As a further attempt, the “mlxtend” package is selected and the command “from mlxtend.frequent\_patterns import apriori” can import the Apriori algorithm function.

Although the algorithm is the same, this new package requires Dataframe as input data format. Also, it needs all elements of the subset as the new columns of the new Dataframe and use “True” or “False” as the class (see the graph below).

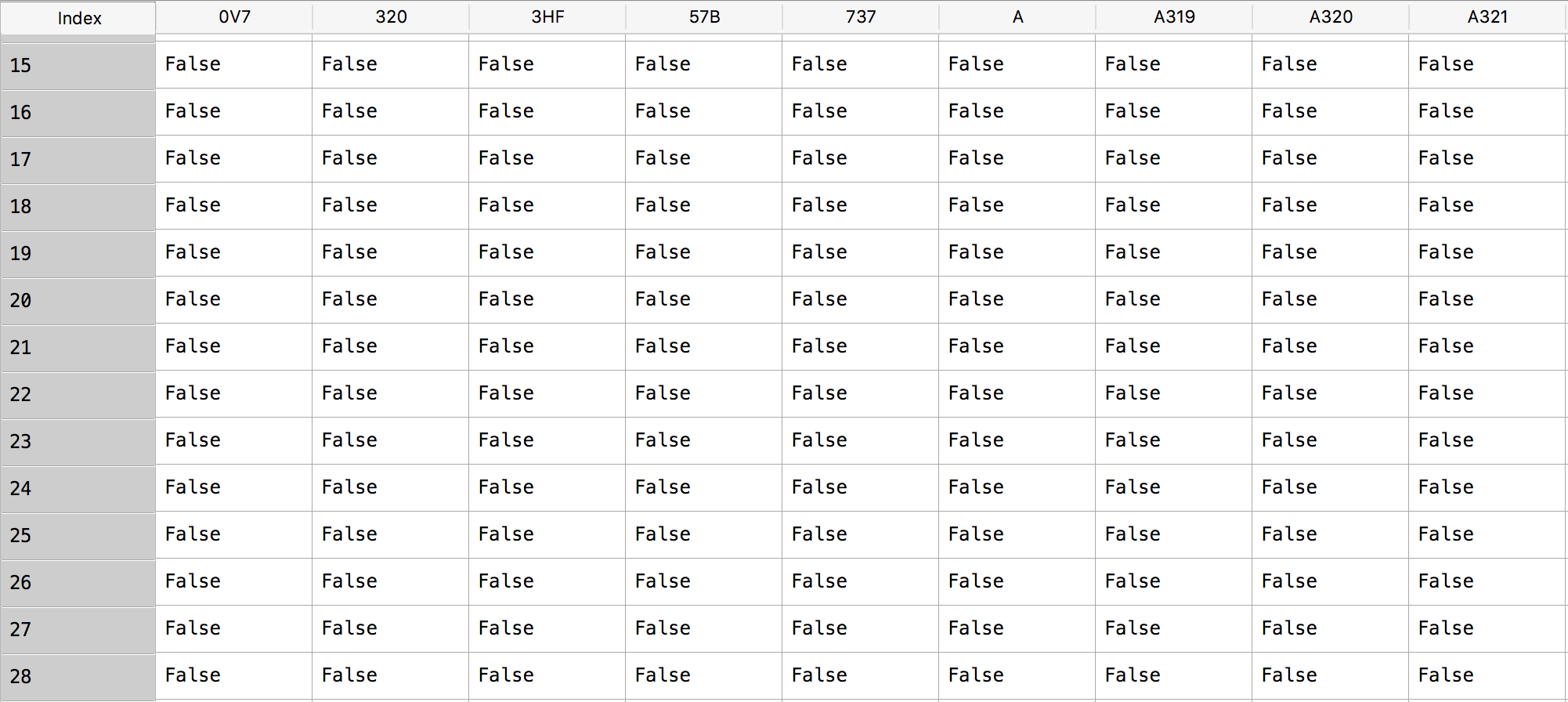


Figure 4.1

Then, use following codes, the outcome with required minimum support and rule size can be obtained. The threshold in this test is min\_support=0.002, since it is much more easier to find the rules and frequent sets with long size.

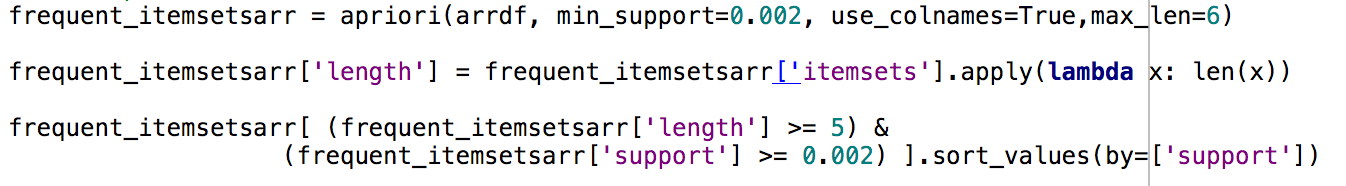


Figure 4.2

Four “.csv” files were created to save the output, with type “arrival” and “departure” and frequent itemset size 5 and 6.

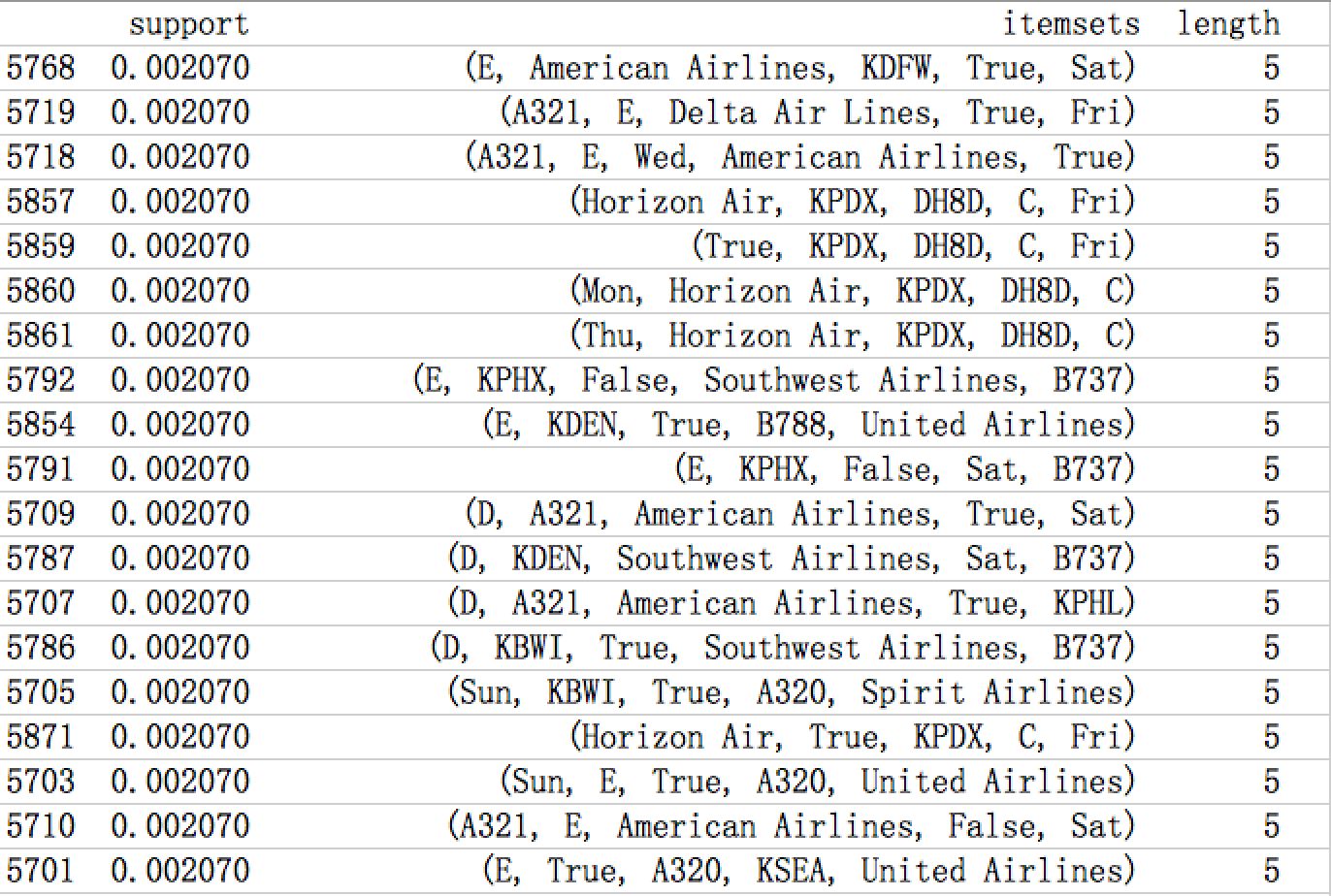


Figure 4.3

The itemsets in the files were sorted by support values.

1. Findings and results

By analyzing the experimental data, the higher the minimum support is, the smaller frequent itemset sizes of the outcome. And the number of rules would also decline rapidly. Focus on the frequent itemsets which have no less than 5 elements inside, there are some results:

In arrival subset, the most frequent patterns with length 5 is (E, False, Southwest Airlines, Sat, B737), with support 0.036999. The second one is (E, False, Southwest Airlines, Sat, B738), with support 0.013972. That means if a person taking Southwest Airlines’ boeing 737 or 738 planes in Saturday with airspeed catalog “E”, there is large probability that this flight will not has arrival delay.

When the patterns’ length is 6, there are only 5 itemsets, they are

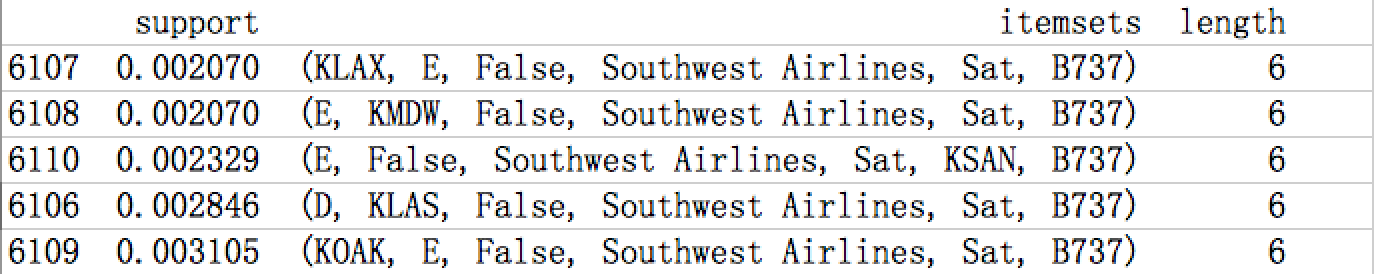


Figure 5.1

Still the Saturday, Southwest Airlines, Boeing 737, E type airspeed and no delay. The supports became much smaller, but the patterns are consistent with former.

In departure subset, when length equals to 5, the (E, True, Southwest Airlines, Sat, B737) is still the most frequent with support 0.025097. However, the second one is now (D, True, E75L, Mesa Airlines, KIAH) with support 0.017076. Notice that in this pattern, the delay option is “True”, which means the high frequency of delay for all Mesa Airlines’ flights with E75L plane arriving at KIAH with airspeed type D. When the length is 6, there are in total 11 patterns, they are

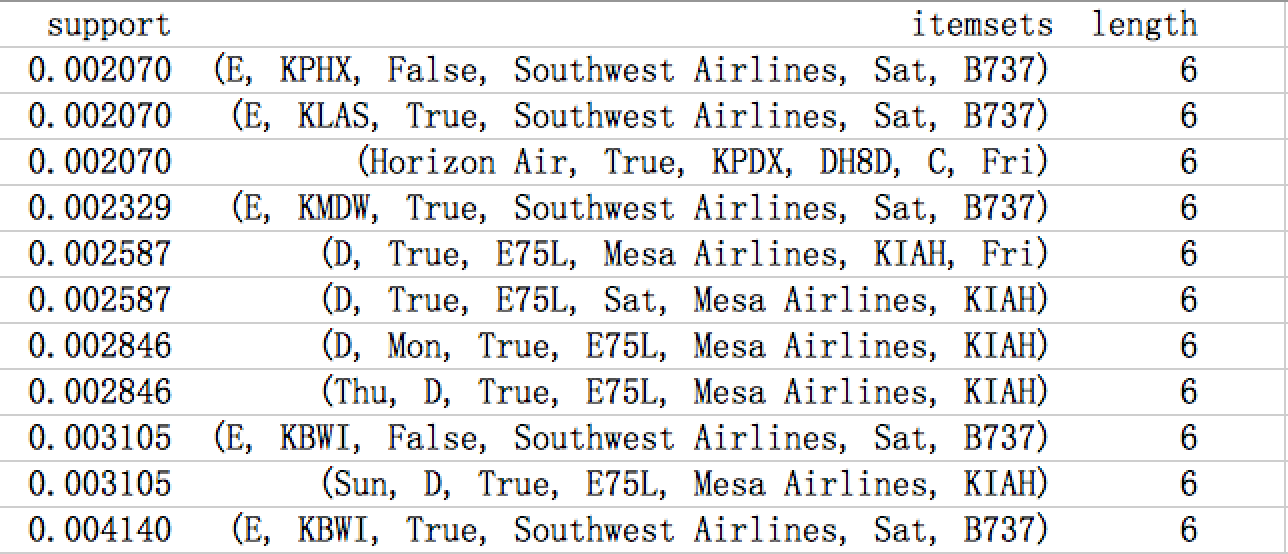


Figure 5.2

The most frequent one still contains the Southwest Airlines, B737 and Saturday. The interesting thing is, in this chart, most patterns are with “delay=true” while in arrival part, all patterns are with “delay= False” if the length is 6. So it might be reasonable to say that the when considering all features, departure delay is more likely to occurs than arrival delay.

Then thinking about the rules. The real key rules should be the rules that demonstrate if the flights will delay or not. Same as the frequent patterns, only the rules whose sizes are as long as possible will be taken into account.

For arrival delay subset, there are 5 key rules:

Rule: ['Southwest Airlines', 'Sat', 'B737'] -> False

Support: 0.05717981888745149

Confidence: 0.9608695652173913

Lift: 11.64188360365272

=====================================

Rule: ['Southwest Airlines', 'B737'] -> False

Support: 0.06390685640362224

Confidence: 0.7373134328358208

Lift: 8.933280306929303

=====================================

Rule: ['KFLL', 'A320', 'Fri', 'Spirit Airlines'] -> False

Support: 0.016597510373443983

Confidence: 1.0

Lift: 3.3943661971830985

=====================================

Rule: ['KBOS', 'Mon', 'JetBlue Airways'] -> False

Support: 0.012448132780082987

Confidence: 0.6

Lift: 4.131428571428571

=====================================

Rule: ['KBOS', 'Fri', 'JetBlue Airways'] -> False

Support: 0.016597510373443983

Confidence: 0.6666666666666667

Lift: 4.590476190476191

=====================================

For departure delay subset, the 5 key rules are:

Rule: ['Southwest Airlines', 'Sat', 'B737'] -> False

Support: 0.024062095730918498

Confidence: 0.9789473684210526

Lift: 11.860914040587362

=====================================

Rule: ['Southwest Airlines', 'E', 'B737'] -> False

Support: 0.01500646830530401

Confidence: 0.8656716417910447

Lift: 10.488466757123474

=====================================

Rule: ['E', 'Sat', 'B737'] -> False

Support: 0.01371280724450194

Confidence: 0.7910447761194029

Lift: 3.2490840166859645

=====================================

Rule: ['Southwest Airlines', 'D', 'B737'] -> False

Support: 0.010090556274256144

Confidence: 0.8125

Lift: 9.844239811912226

=====================================

Rule: ['D', 'Sat', 'B737'] -> False

Support: 0.01034928848641656

Confidence: 0.8333333333333335

Lift: 3.4227771873893027

=====================================

The rules are generally come from the most frequent patterns that showed before. Basically, the results are not surprising since they agree with the results in machine learning part.