

# CIND820\_Capstone\_Project\_Association\_Rule\_edited

July 21, 2023

## 0.1 CIND820 - Capstone Project

### 1 Investigate Airline passenger satisfaction using Machine Learning Techniques

### 2 Preparation:

```
[ ]: # check python version  
[ ]: !python -V
```

Python 3.10.12

```
[ ]: !pip install pandas-profiling
```

```
[ ]: !pip install mlxtend
```

Requirement already satisfied: mlxtend in /usr/local/lib/python3.10/dist-packages (0.22.0)

Requirement already satisfied: scipy>=1.2.1 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (1.10.1)

Requirement already satisfied: numpy>=1.16.2 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (1.22.4)

Requirement already satisfied: pandas>=0.24.2 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (1.5.3)

Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (1.2.2)

Requirement already satisfied: matplotlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (3.7.1)

Requirement already satisfied: joblib>=0.13.2 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (1.3.1)

Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from mlxtend) (67.7.2)

Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlxtend) (1.1.0)

Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlxtend) (0.11.0)

Requirement already satisfied: fonttools>=4.22.0 in

```

/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlxtend)
(4.41.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlxtend)
(1.4.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlxtend) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-
packages (from matplotlib>=3.0.0->mlxtend) (8.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlxtend)
(3.1.0)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlxtend)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas>=0.24.2->mlxtend) (2022.7.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.2->mlxtend)
(3.1.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend) (1.16.0)

```

```
[ ]: !pip install --upgrade ipykernel
```

```

Requirement already satisfied: ipykernel in /usr/local/lib/python3.10/dist-
packages (6.24.0)
Requirement already satisfied: comm>=0.1.1 in /usr/local/lib/python3.10/dist-
packages (from ipykernel) (0.1.3)
Requirement already satisfied: debugpy>=1.6.5 in /usr/local/lib/python3.10/dist-
packages (from ipykernel) (1.6.6)
Requirement already satisfied: ipython>=7.23.1 in
/usr/local/lib/python3.10/dist-packages (from ipykernel) (7.34.0)
Requirement already satisfied: jupyter-client>=6.1.12 in
/usr/local/lib/python3.10/dist-packages (from ipykernel) (6.1.12)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in
/usr/local/lib/python3.10/dist-packages (from ipykernel) (5.3.1)
Requirement already satisfied: matplotlib-inline>=0.1 in
/usr/local/lib/python3.10/dist-packages (from ipykernel) (0.1.6)
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.10/dist-
packages (from ipykernel) (1.5.6)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from ipykernel) (23.1)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages
(from ipykernel) (5.9.5)
Requirement already satisfied: pyzmq>=20 in /usr/local/lib/python3.10/dist-
packages (from ipykernel) (23.2.1)
Requirement already satisfied: tornado>=6.1 in /usr/local/lib/python3.10/dist-

```

```

packages (from ipykernel) (6.3.1)
Requirement already satisfied: traitlets>=5.4.0 in
/usr/local/lib/python3.10/dist-packages (from ipykernel) (5.7.1)
Requirement already satisfied: setuptools>=18.5 in
/usr/local/lib/python3.10/dist-packages (from ipython>=7.23.1->ipykernel)
(67.7.2)
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.10/dist-
packages (from ipython>=7.23.1->ipykernel) (0.18.2)
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-
packages (from ipython>=7.23.1->ipykernel) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-
packages (from ipython>=7.23.1->ipykernel) (0.7.5)
Requirement already satisfied: prompt-toolkit!=3.0.0,!<3.0.1,<3.1.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from ipython>=7.23.1->ipykernel)
(3.0.39)
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-
packages (from ipython>=7.23.1->ipykernel) (2.14.0)
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-
packages (from ipython>=7.23.1->ipykernel) (0.2.0)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-
packages (from ipython>=7.23.1->ipykernel) (4.8.0)
Requirement already satisfied: python-dateutil>=2.1 in
/usr/local/lib/python3.10/dist-packages (from jupyter-client>=6.1.12->ipykernel)
(2.8.2)
Requirement already satisfied: platformdirs>=2.5 in
/usr/local/lib/python3.10/dist-packages (from jupyter-
core!=5.0.*,>=4.12->ipykernel) (3.8.1)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in
/usr/local/lib/python3.10/dist-packages (from
jedi>=0.16->ipython>=7.23.1->ipykernel) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in
/usr/local/lib/python3.10/dist-packages (from
pexpect>4.3->ipython>=7.23.1->ipykernel) (0.7.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-
packages (from prompt-
toolkit!=3.0.0,!<3.0.1,<3.1.0,>=2.0.0->ipython>=7.23.1->ipykernel) (0.2.6)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.1->jupyter-client>=6.1.12->ipykernel) (1.16.0)

```

```
[ ]: pip install tabulate
```

```

Requirement already satisfied: tabulate in /usr/local/lib/python3.10/dist-
packages (0.8.10)

```

```
[1]: # Importing required libraries
```

```
import pandas as pd
```

```

import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder
from sklearn import preprocessing

from sklearn.preprocessing import MinMaxScaler
import sklearn.feature_selection as fs
import sklearn.datasets as datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import sklearn.metrics as metrics
import matplotlib.pyplot as plt

from sklearn.feature_selection import SelectKBest, f_classif
import sklearn.feature_selection as fs

import sklearn.datasets as datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import sklearn.metrics as metrics
import matplotlib.pyplot as plt

from tabulate import tabulate

from imblearn.over_sampling import SMOTE
from collections import Counter
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import RandomOverSampler
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import classification_report
# from xgboost import XGBClassifier

import sklearn
import time
from resource import getrusage, RUSAGE_SELF
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import f1_score

```

```

from sklearn.metrics import recall_score
from sklearn.metrics import roc_auc_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve
import xgboost
from xgboost import XGBClassifier
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

from sklearn.model_selection import GridSearchCV
from sklearn.metrics import make_scorer, accuracy_score

import pandas as pd
import numpy as np
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

from mlxtend.preprocessing import TransactionEncoder

import warnings
warnings.filterwarnings("ignore")

```

Import csv file (the dataset and the data dictionary)

```

[2]: # Import the dataset
      # Use first column "ID" as Index by using index_col=0

      url = 'https://raw.githubusercontent.com/HitomiMo/CIND820_Capstone-Project/main/
      ↪airline_passenger_satisfaction.csv'
      df1 = pd.read_csv(url, index_col=0)
      df1.head()

```

```

[2]:
      Gender  Age Customer Type Type of Travel      Class  Flight Distance \
ID
1      Male   48   First-time      Business Business              821
2      Female 35   Returning      Business Business              821
3      Male   41   Returning      Business Business              853
4      Male   50   Returning      Business Business             1905
5      Female 49   Returning      Business Business             3470

      Departure Delay  Arrival Delay  Departure and Arrival Time Convenience \
ID
1                   2              5.0                                   3
2                  26             39.0                                   2
3                   0              0.0                                   4
4                   0              0.0                                   2
5                   0              1.0                                   3

```

	Ease of Online Booking	...	On-board Service	Seat Comfort	\
ID		...			
1	3	...	3	5	
2	2	...	5	4	
3	4	...	3	5	
4	2	...	5	5	
5	3	...	3	4	

	Leg Room Service	Cleanliness	Food and Drink	In-flight Service	\
ID					
1	2	5	5	5	
2	5	5	3	5	
3	3	5	5	3	
4	5	4	4	5	
5	4	5	4	3	

	In-flight Wifi Service	In-flight Entertainment	Baggage Handling	\
ID				
1	3	5	5	
2	2	5	5	
3	4	3	3	
4	2	5	5	
5	3	3	3	

	Satisfaction
ID	
1	Neutral or Dissatisfied
2	Satisfied
3	Satisfied
4	Satisfied
5	Satisfied

[5 rows x 23 columns]

```
[ ]: # Import the dictionary

url2 = 'https://raw.githubusercontent.com/HitomiMo/CIND820_Capstone-Project/
      ↪main/data_dictionary.csv'
data_dictionary = pd.read_csv(url2, index_col=0)
data_dictionary
```

```
[ ]: Description
Field
ID Unique passenger
identifier
Gender Gender of the passenger
```

(Female/Male)	
Age	Age of the
passenger	
Customer Type	Type of airline customer (First-
time/Returning)	
Type of Travel	Purpose of the flight
(Business/Personal)	
Class	Travel class in the airplane for the
passenger...	
Flight Distance	Flight distance
in miles	
Departure Delay	Flight departure delay
in minutes	
Arrival Delay	Flight arrival delay
in minutes	
Departure and Arrival Time Convenience	Satisfaction level with the convenience
of the...	
Ease of Online Booking	Satisfaction level with the online
booking exp...	
Check-in Service	Satisfaction level with the check-in
service f...	
Online Boarding	Satisfaction level with the online
boarding ex...	
Gate Location	Satisfaction level with the gate
location in t...	
On-board Service	Satisfaction level with the on-boarding
servic...	
Seat Comfort	Satisfaction level with the comfort of
the air...	
Leg Room Service	Satisfaction level with the leg room of
the ai...	
Cleanliness	Satisfaction level with the cleanliness
of the...	
Food and Drink	Satisfaction level with the food and
drinks on...	
In-flight Service	Satisfaction level with the in-flight
service ...	
In-flight Wifi Service	Satisfaction level with the in-flight
Wifi ser...	
In-flight Entertainment	Satisfaction level with the in-flight
entertai...	
Baggage Handling	Satisfaction level with the baggage
handling f...	
Satisfaction	Overall satisfaction level with the
airline (S...	

## 2.1 Data preparation for Association Rules

```
[3]: df1.shape
```

```
[3]: (129880, 23)
```

```
[4]: # Checking % of missing values
percent_missing = df1['Arrival Delay'].isnull().sum() * 100 / len(df1['Arrival_
↳ Delay'])
percent_missing
```

```
[4]: 0.3025870033877425
```

```
[5]: # Missing value ratio is only 0.3%. Therefore, I will remove missing values_
↳ before splitting dataset
AR_df = df1.copy()
AR_df = AR_df.dropna(how='any',axis=0)
AR_df.isnull().sum()
```

```
[5]: Gender                                0
Age                                         0
Customer Type                             0
Type of Travel                             0
Class                                       0
Flight Distance                           0
Departure Delay                           0
Arrival Delay                             0
Departure and Arrival Time Convenience    0
Ease of Online Booking                    0
Check-in Service                          0
Online Boarding                           0
Gate Location                             0
On-board Service                          0
Seat Comfort                              0
Leg Room Service                          0
Cleanliness                               0
Food and Drink                            0
In-flight Service                         0
In-flight Wifi Service                    0
In-flight Entertainment                   0
Baggage Handling                          0
Satisfaction                              0
dtype: int64
```

```
[6]: # check the # of entries after removing missing values
row_count = len(AR_df.index)
row_count
```



[6]: 129487

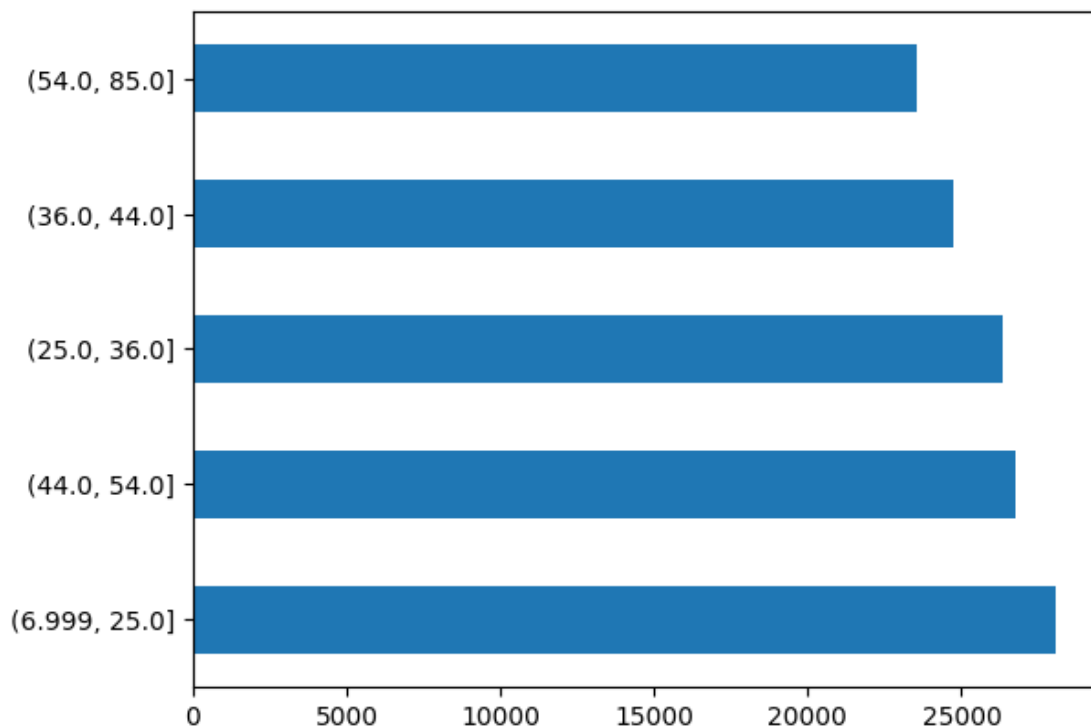
Apply Quantile-cut (qcut) for Age, Flight Distance, Departure Delay and Arrival Delay.

```
[7]: AR_df['Age_qcut'] = pd.qcut(AR_df.Age, q=5)
AR_df['Age_qcut'].head()
```

```
[7]: ID
1    (44.0, 54.0]
2    (25.0, 36.0]
3    (36.0, 44.0]
4    (44.0, 54.0]
5    (44.0, 54.0]
Name: Age_qcut, dtype: category
Categories (5, interval[float64, right]): [(6.999, 25.0] < (25.0, 36.0] < (36.0, 44.0] <
                                         (44.0, 54.0] < (54.0, 85.0]]
```

```
[8]: AR_df['Age_qcut'].value_counts().plot(kind='barh')
plt.show()

qcut_series, qcut_intervals = pd.qcut(AR_df.Age, q=5, retbins=True)
qcut_series.value_counts()
```



```
[8]: (6.999, 25.0]      28089
      (44.0, 54.0]      26757
      (25.0, 36.0]      26346
      (36.0, 44.0]      24757
      (54.0, 85.0]      23538
      Name: Age, dtype: int64
```

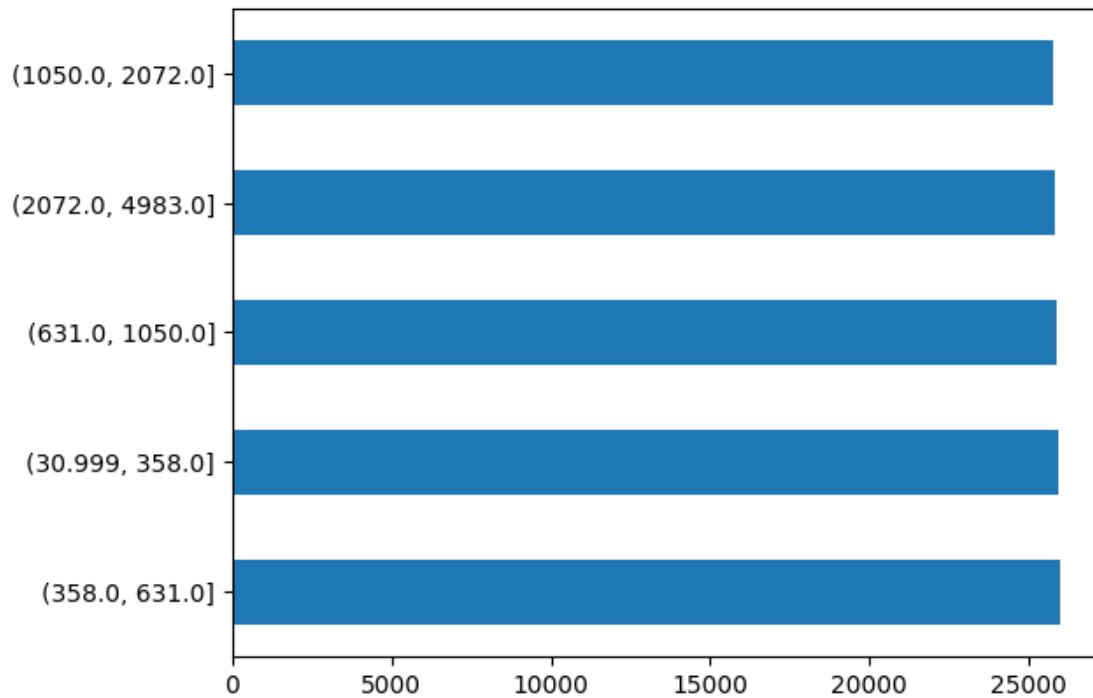
```
[9]: AR_df['Flight Distance_qcut'] = pd.qcut(AR_df["Flight Distance"], q=5)
      AR_df['Flight Distance_qcut'].head()
```

```
[9]: ID
      1      (631.0, 1050.0]
      2      (631.0, 1050.0]
      3      (631.0, 1050.0]
      4      (1050.0, 2072.0]
      5      (2072.0, 4983.0]
      Name: Flight Distance_qcut, dtype: category
      Categories (5, interval[float64, right]): [(30.999, 358.0] < (358.0, 631.0] <
      (631.0, 1050.0] <
      (1050.0, 2072.0] < (2072.0, 4983.0]]
```

```
[10]: AR_df['Flight Distance_qcut'].value_counts().plot(kind='barh')
      plt.show()

      qcut_series, qcut_intervals = pd.qcut(AR_df["Flight Distance"], q=5,
      ↪retbins=True)

      qcut_series.value_counts()
```



```
[10]: (358.0, 631.0]      26021
      (30.999, 358.0]    25955
      (631.0, 1050.0]    25899
      (2072.0, 4983.0]   25850
      (1050.0, 2072.0]   25762
      Name: Flight Distance, dtype: int64
```

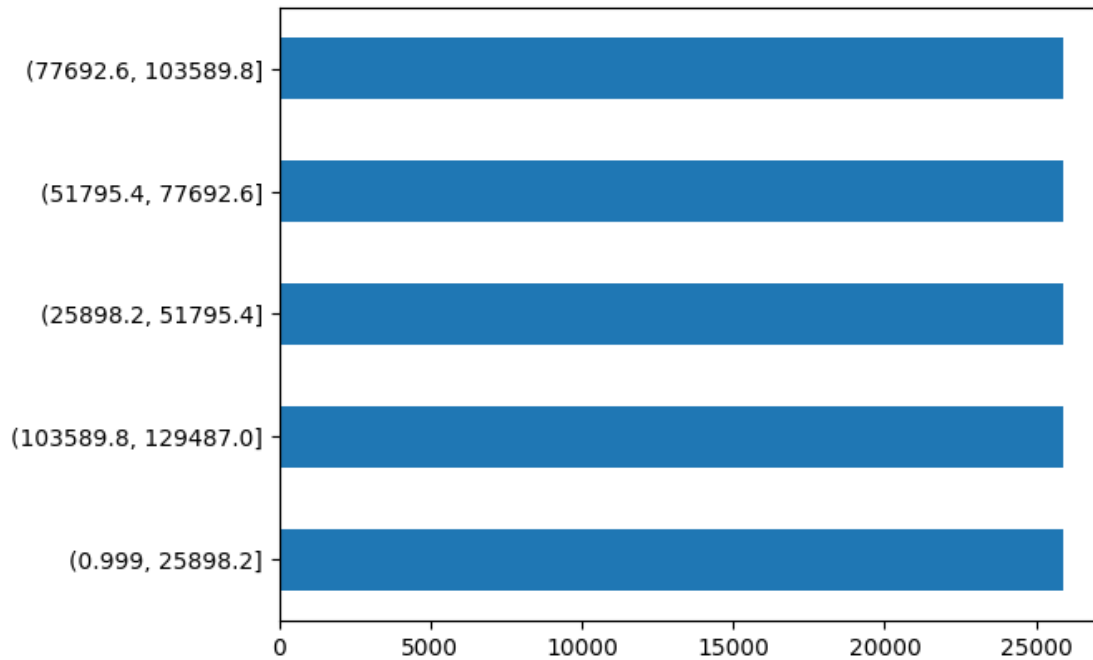
```
[11]: AR_df['Departure Delay_qcut'] = pd.qcut(AR_df["Departure Delay"],
      ↪rank(method='first'), q=5)
      AR_df['Departure Delay_qcut'].head()
```

```
[11]: ID
      1      (51795.4, 77692.6]
      2      (103589.8, 129487.0]
      3      (0.999, 25898.2]
      4      (0.999, 25898.2]
      5      (0.999, 25898.2]
      Name: Departure Delay_qcut, dtype: category
      Categories (5, interval[float64, right]): [(0.999, 25898.2] < (25898.2, 51795.4]
      <
      (51795.4, 77692.6] < (77692.6,
      103589.8] <
      (103589.8, 129487.0]]
```

```
[12]: AR_df['Departure Delay_qcut'].value_counts().plot(kind='barh')
plt.show()

qcut_series, qcut_intervals = pd.qcut(AR_df["Departure Delay"].
    ↪rank(method='first'), q=5, retbins=True)

qcut_series.value_counts()
```



```
[12]: (0.999, 25898.2]      25898
      (103589.8, 129487.0]  25898
      (25898.2, 51795.4]   25897
      (51795.4, 77692.6]   25897
      (77692.6, 103589.8]  25897
      Name: Departure Delay, dtype: int64
```

```
[13]: AR_df['Arrival Delay_qcut'] = pd.qcut(AR_df["Arrival Delay"].
    ↪rank(method='first'), q=5)
AR_df['Arrival Delay_qcut'].head()
```

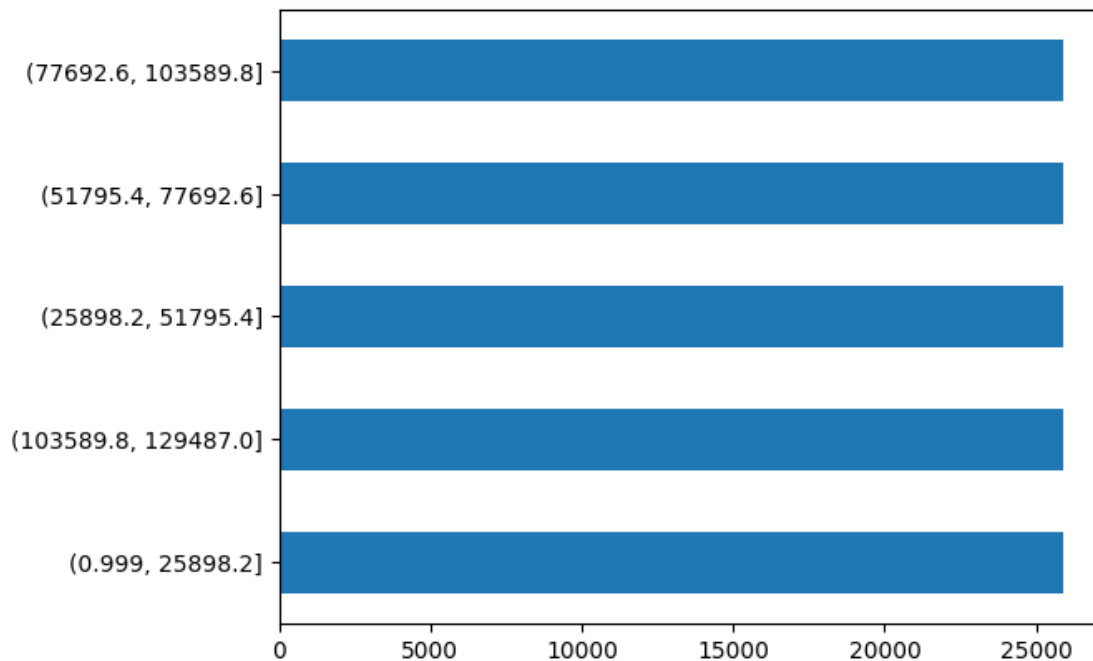
```
[13]: ID
1      (77692.6, 103589.8]
2      (103589.8, 129487.0]
3      (0.999, 25898.2]
4      (0.999, 25898.2]
5      (51795.4, 77692.6]
```

```
Name: Arrival Delay_qcut, dtype: category
Categories (5, interval[float64, right]): [(0.999, 25898.2] < (25898.2, 51795.4]
<
(51795.4, 77692.6] < (77692.6,
103589.8] <
(103589.8, 129487.0]]
```

```
[14]: AR_df['Arrival Delay_qcut'].value_counts().plot(kind='barh')
plt.show()

qcut_series, qcut_intervals = pd.qcut(AR_df["Arrival Delay"].
    ↪rank(method='first'), q=5, retbins=True)

qcut_series.value_counts()
```



```
[14]: (0.999, 25898.2]      25898
      (103589.8, 129487.0] 25898
      (25898.2, 51795.4]   25897
      (51795.4, 77692.6]   25897
      (77692.6, 103589.8]  25897
      Name: Arrival Delay, dtype: int64
```

```
[15]: columns = ['Satisfaction',
                  'Gender',
                  'Age_qcut',
```

```

'Customer Type',
'Type of Travel',
'Class',
'Flight Distance_qcut',
'Departure Delay_qcut',
'Arrival Delay_qcut',
'Departure and Arrival Time Convenience',
'Ease of Online Booking',
'Check-in Service',
'Online Boarding',
'Gate Location',
'On-board Service',
'Seat Comfort',
'Leg Room Service',
'Cleanliness',
'Food and Drink',
'In-flight Service',
'In-flight Wifi Service',
'In-flight Entertainment',
'Baggage Handling']

```

```
not_used_columns = list(set(AR_df.columns.to_list()) - set(columns))
```

```
[16]: AR_df = pd.get_dummies(AR_df, columns=columns)
```

```
[17]: AR_df.drop(labels=not_used_columns, axis=1, inplace=True)
```

```
[18]: AR_df.head()
```

```
[18]:
```

	Satisfaction_Neutral or Dissatisfied	Satisfaction_Satisfied	\
ID			
1	1	0	
2	0	1	
3	0	1	
4	0	1	
5	0	1	

	Gender_Female	Gender_Male	Age_qcut_(6.999, 25.0]	Age_qcut_(25.0, 36.0]	\
ID					
1	0	1	0	0	
2	1	0	0	1	
3	0	1	0	0	
4	0	1	0	0	
5	1	0	0	0	

	Age_qcut_(36.0, 44.0]	Age_qcut_(44.0, 54.0]	Age_qcut_(54.0, 85.0]	\
ID				

1	0	1	0
2	0	0	0
3	1	0	0
4	0	1	0
5	0	1	0

	Customer Type_First-time	...	In-flight Entertainment_1	\
ID		...		
1	1	...	0	
2	0	...	0	
3	0	...	0	
4	0	...	0	
5	0	...	0	

	In-flight Entertainment_2	In-flight Entertainment_3	\
ID			
1	0	0	
2	0	0	
3	0	1	
4	0	0	
5	0	1	

	In-flight Entertainment_4	In-flight Entertainment_5	Baggage Handling_1	\
ID				
1	0	1	0	
2	0	1	0	
3	0	0	0	
4	0	1	0	
5	0	0	0	

	Baggage Handling_2	Baggage Handling_3	Baggage Handling_4	\
ID				
1	0	0	0	
2	0	0	0	
3	0	1	0	
4	0	0	0	
5	0	1	0	

	Baggage Handling_5
ID	
1	1
2	1
3	0
4	1
5	0

[5 rows x 114 columns]

```
[19]: AR_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 129487 entries, 1 to 129880
Columns: 114 entries, Satisfaction_Neutral or Dissatisfied to Baggage Handling_5
dtypes: uint8(114)
memory usage: 15.1 MB
```

```
[20]: # iterating the columns
      for col in AR_df.columns:
          print(col)
```

```
Satisfaction_Neutral or Dissatisfied
Satisfaction_Satisfied
Gender_Female
Gender_Male
Age_qcut_(6.999, 25.0]
Age_qcut_(25.0, 36.0]
Age_qcut_(36.0, 44.0]
Age_qcut_(44.0, 54.0]
Age_qcut_(54.0, 85.0]
Customer_Type_First-time
Customer_Type_Returning
Type of Travel_Business
Type of Travel_Personal
Class_Business
Class_Economy
Class_Economy Plus
Flight_Distance_qcut_(30.999, 358.0]
Flight_Distance_qcut_(358.0, 631.0]
Flight_Distance_qcut_(631.0, 1050.0]
Flight_Distance_qcut_(1050.0, 2072.0]
Flight_Distance_qcut_(2072.0, 4983.0]
Departure_Delay_qcut_(0.999, 25898.2]
Departure_Delay_qcut_(25898.2, 51795.4]
Departure_Delay_qcut_(51795.4, 77692.6]
Departure_Delay_qcut_(77692.6, 103589.8]
Departure_Delay_qcut_(103589.8, 129487.0]
Arrival_Delay_qcut_(0.999, 25898.2]
Arrival_Delay_qcut_(25898.2, 51795.4]
Arrival_Delay_qcut_(51795.4, 77692.6]
Arrival_Delay_qcut_(77692.6, 103589.8]
Arrival_Delay_qcut_(103589.8, 129487.0]
Departure_and_Arrival_Time_Convenience_0
Departure_and_Arrival_Time_Convenience_1
Departure_and_Arrival_Time_Convenience_2
Departure_and_Arrival_Time_Convenience_3
Departure_and_Arrival_Time_Convenience_4
```



Departure and Arrival Time Convenience\_5  
Ease of Online Booking\_0  
Ease of Online Booking\_1  
Ease of Online Booking\_2  
Ease of Online Booking\_3  
Ease of Online Booking\_4  
Ease of Online Booking\_5  
Check-in Service\_0  
Check-in Service\_1  
Check-in Service\_2  
Check-in Service\_3  
Check-in Service\_4  
Check-in Service\_5  
Online Boarding\_0  
Online Boarding\_1  
Online Boarding\_2  
Online Boarding\_3  
Online Boarding\_4  
Online Boarding\_5  
Gate Location\_0  
Gate Location\_1  
Gate Location\_2  
Gate Location\_3  
Gate Location\_4  
Gate Location\_5  
On-board Service\_0  
On-board Service\_1  
On-board Service\_2  
On-board Service\_3  
On-board Service\_4  
On-board Service\_5  
Seat Comfort\_0  
Seat Comfort\_1  
Seat Comfort\_2  
Seat Comfort\_3  
Seat Comfort\_4  
Seat Comfort\_5  
Leg Room Service\_0  
Leg Room Service\_1  
Leg Room Service\_2  
Leg Room Service\_3  
Leg Room Service\_4  
Leg Room Service\_5  
Cleanliness\_0  
Cleanliness\_1  
Cleanliness\_2  
Cleanliness\_3  
Cleanliness\_4

```

Cleanliness_5
Food and Drink_0
Food and Drink_1
Food and Drink_2
Food and Drink_3
Food and Drink_4
Food and Drink_5
In-flight Service_0
In-flight Service_1
In-flight Service_2
In-flight Service_3
In-flight Service_4
In-flight Service_5
In-flight Wifi Service_0
In-flight Wifi Service_1
In-flight Wifi Service_2
In-flight Wifi Service_3
In-flight Wifi Service_4
In-flight Wifi Service_5
In-flight Entertainment_0
In-flight Entertainment_1
In-flight Entertainment_2
In-flight Entertainment_3
In-flight Entertainment_4
In-flight Entertainment_5
Baggage Handling_1
Baggage Handling_2
Baggage Handling_3
Baggage Handling_4
Baggage Handling_5

```

```
[21]: len(AR_df.columns)
```

```
[21]: 114
```

**Exploratory Data Analysis (EDA) using pandas-profiling after catagorizing dataset**

```
[ ]: pip install pandas-profiling
```

```
[ ]: from pandas_profiling import ProfileReport
prof = ProfileReport(AR_df)
prof.to_file(output_file='output.html')
```

```
Summarize dataset: 0%|          | 0/5 [00:00<?, ?it/s]
```

```
Generate report structure: 0%|          | 0/1 [00:00<?, ?it/s]
```

```
Render HTML: 0%|          | 0/1 [00:00<?, ?it/s]
```

```
Export report to file: 0%|          | 0/1 [00:00<?, ?it/s]
```

## Apply Apriori algorithm

```
[22]: AR_AP = AR_df.copy()
```

```
[23]: AR_AP['Satisfaction_Satisfied'].value_counts()
```

```
[23]: 0    73225
      1    56262
      Name: Satisfaction_Satisfied, dtype: int64
```

## Set Satisfaction\_Satisfied as consequents

```
[37]: '''
      Filtering only consequents with Satisfaction_Satisfied
      '''

      #Apriori min support
      min_support = 0.1

      #Max lenght of apriori n-grams
      max_len = 3

      frequent_items = apriori(AR_AP, use_colnames=True, min_support=min_support,
      ↪max_len=max_len + 1)
      rules = association_rules(frequent_items, metric='lift', min_threshold=1)

      target = '{\'Satisfaction_Satisfied\'}'

      # results_Satisfaction_Satisfied = rules[rules['consequents'].astype(str).str.
      ↪contains(target, na=False)].sort_values(by='confidence', ascending=False)
      results_Satisfaction_Satisfied = rules[rules['consequents'].astype(str).str.
      ↪contains(target, na=False)]

      results_Satisfaction_Satisfied.nlargest(n = 10, columns = 'support')
```

```
[37]: antecedents \
      92          (Type of Travel_Business)
      91          (Customer Type_Returning)
      1005 (Type of Travel_Business, Customer Type_Return...
      94          (Class_Business)
      1149 (Type of Travel_Business, Class_Business)
      1011 (Class_Business, Customer Type_Returning)
      3320 (Type of Travel_Business, Class_Business, Cust...
      84          (Gender_Male)
      966 (Type of Travel_Business, Gender_Female)
      980 (Type of Travel_Business, Gender_Male)
```

	consequents	antecedent support	consequent support \
92	(Satisfaction_Satisfied)	0.690764	0.434499
91	(Satisfaction_Satisfied)	0.816862	0.434499
1005	(Satisfaction_Satisfied)	0.509163	0.434499
94	(Satisfaction_Satisfied)	0.478735	0.434499
1149	(Satisfaction_Satisfied)	0.458154	0.434499
1011	(Satisfaction_Satisfied)	0.407608	0.434499
3320	(Satisfaction_Satisfied)	0.387128	0.434499
84	(Satisfaction_Satisfied)	0.492590	0.434499
966	(Satisfaction_Satisfied)	0.352630	0.434499
980	(Satisfaction_Satisfied)	0.338134	0.434499

	support	confidence	lift	leverage	conviction	zhangs_metric
92	0.403183	0.583677	1.343333	0.103047	1.358323	0.826499
91	0.390572	0.478137	1.100433	0.035646	1.083620	0.498348
1005	0.359488	0.706037	1.624944	0.138257	1.923712	0.783547
94	0.332466	0.694467	1.598316	0.124456	1.850866	0.718141
1149	0.330049	0.720388	1.657972	0.130981	2.022446	0.732411
1011	0.304239	0.746400	1.717840	0.127133	2.229894	0.705401
3320	0.301822	0.779644	1.794351	0.133615	2.566307	0.722329
84	0.216879	0.440283	1.013311	0.002849	1.010333	0.025889
966	0.201889	0.572524	1.317663	0.048672	1.322882	0.372400
980	0.201294	0.595309	1.370103	0.054375	1.397364	0.408131

sort by highest support

```
[52]: results_Satisfaction_Satisfied_support = \
      ↪results_Satisfaction_Satisfied[["antecedents", "consequents", "support"]]
      results_Satisfaction_Satisfied_support = results_Satisfaction_Satisfied_support.
      ↪sort_values(by='support', ascending=False)
      results_Satisfaction_Satisfied_support.head(10)
```

```
[52]: antecedents \
92 (Type of Travel_Business)
91 (Customer Type_Returning)
1005 (Type of Travel_Business, Customer Type_Return...
94 (Class_Business)
1149 (Type of Travel_Business, Class_Business)
1011 (Class_Business, Customer Type_Returning)
3320 (Type of Travel_Business, Class_Business, Cust...
84 (Gender_Male)
966 (Type of Travel_Business, Gender_Female)
980 (Type of Travel_Business, Gender_Male)

consequents support
92 (Satisfaction_Satisfied) 0.403183
91 (Satisfaction_Satisfied) 0.390572
```

1005	(Satisfaction_Satisfied)	0.359488
94	(Satisfaction_Satisfied)	0.332466
1149	(Satisfaction_Satisfied)	0.330049
1011	(Satisfaction_Satisfied)	0.304239
3320	(Satisfaction_Satisfied)	0.301822
84	(Satisfaction_Satisfied)	0.216879
966	(Satisfaction_Satisfied)	0.201889
980	(Satisfaction_Satisfied)	0.201294

```
[39]: results_Satisfaction_Satisfied.nlargest(n = 10, columns = 'confidence')
```

```
[39]: antecedents \
```

3404	(Type of Travel_Business, Customer Type_Return...
3376	(Type of Travel_Business, Check-in Service_5, ...
4174	(Type of Travel_Business, Class_Business, Onli...
3628	(Type of Travel_Business, Baggage Handling_5, ...
3459	(Type of Travel_Business, Seat Comfort_5, Cust...
3572	(Type of Travel_Business, In-flight Service_5,...
3516	(Type of Travel_Business, Cleanliness_5, Custo...
137	(In-flight Wifi Service_5)
3600	(Type of Travel_Business, In-flight Entertainm...
1185	(Type of Travel_Business, Online Boarding_5)

	consequents	antecedent support	consequent support	\
3404	(Satisfaction_Satisfied)	0.147343	0.434499	
3376	(Satisfaction_Satisfied)	0.102651	0.434499	
4174	(Satisfaction_Satisfied)	0.138076	0.434499	
3628	(Satisfaction_Satisfied)	0.135473	0.434499	
3459	(Satisfaction_Satisfied)	0.151691	0.434499	
3572	(Satisfaction_Satisfied)	0.134863	0.434499	
3516	(Satisfaction_Satisfied)	0.119680	0.434499	
137	(Satisfaction_Satisfied)	0.110467	0.434499	
3600	(Satisfaction_Satisfied)	0.144524	0.434499	
1185	(Satisfaction_Satisfied)	0.165839	0.434499	

	support	confidence	lift	leverage	conviction	zhangs_metric
3404	0.147289	0.999633	2.300656	0.083269	1541.312779	0.663035
3376	0.102582	0.999323	2.299942	0.057980	835.181824	0.629863
4174	0.137442	0.995414	2.290945	0.077449	123.299860	0.653768
3628	0.134670	0.994071	2.287855	0.075807	95.384757	0.651118
3459	0.150695	0.993432	2.286385	0.084785	86.105166	0.663235
3572	0.133967	0.993357	2.286212	0.075369	85.132243	0.650296
3516	0.118599	0.990966	2.280708	0.066598	62.596897	0.637881
137	0.109378	0.990143	2.278813	0.061380	57.368250	0.630865
3600	0.142632	0.986908	2.271369	0.079836	43.195027	0.654299
1185	0.163584	0.986402	2.270205	0.091527	41.587547	0.670747

sort by highest confidence

```
[53]: results_Satisfaction_Satisfied_confidence =  
      ↪ results_Satisfaction_Satisfied[["antecedents", "consequents", "confidence"]]  
      results_Satisfaction_Satisfied_confidence =  
      ↪ results_Satisfaction_Satisfied_confidence.sort_values(by='confidence',  
      ↪ ascending=False)  
      results_Satisfaction_Satisfied_confidence.head(10)
```

```
[53]:  
      antecedents \  
3404 (Type of Travel_Business, Customer Type_Return...  
3376 (Type of Travel_Business, Check-in Service_5, ...  
4174 (Type of Travel_Business, Class_Business, Onli...  
3628 (Type of Travel_Business, Baggage Handling_5, ...  
3459 (Type of Travel_Business, Seat Comfort_5, Cust...  
3572 (Type of Travel_Business, In-flight Service_5,...  
3516 (Type of Travel_Business, Cleanliness_5, Custo...  
137      (In-flight Wifi Service_5)  
3600 (Type of Travel_Business, In-flight Entertainm...  
1185      (Type of Travel_Business, Online Boarding_5)  
  
      consequents confidence  
3404 (Satisfaction_Satisfied) 0.999633  
3376 (Satisfaction_Satisfied) 0.999323  
4174 (Satisfaction_Satisfied) 0.995414  
3628 (Satisfaction_Satisfied) 0.994071  
3459 (Satisfaction_Satisfied) 0.993432  
3572 (Satisfaction_Satisfied) 0.993357  
3516 (Satisfaction_Satisfied) 0.990966  
137  (Satisfaction_Satisfied) 0.990143  
3600 (Satisfaction_Satisfied) 0.986908  
1185 (Satisfaction_Satisfied) 0.986402
```

Set Satisfaction\_Neutral or Dissatisfied as consequents

```
[41]: '''  
      Filtering only consequents with Satisfaction_Neutral or Dissatisfied  
      '''  
  
      #Apriori min support  
      min_support = 0.1  
  
      #Max lenght of apriori n-grams  
      max_len = 3  
  
      frequent_items = apriori(AR_AP, use_colnames=True, min_support=min_support,  
      ↪ max_len=max_len + 1)  
      rules = association_rules(frequent_items, metric='lift', min_threshold=1)
```

```

target = '{\'Satisfaction_Neutral or Dissatisfied\'}'

# results_Satisfaction_Neutral_or_Dissatisfied = rules[rules['consequents'].
↳ astype(str).str.contains(target, na=False)].sort_values(by='confidence',
↳ ascending=False)
results_Satisfaction_Neutral_or_Dissatisfied = rules[rules['consequents'].
↳ astype(str).str.contains(target, na=False)]

results_Satisfaction_Neutral_or_Dissatisfied.nlargest(n = 10, columns =
↳ 'support')

```

```

[41]:
          antecedents \
12          (Class_Economy)
0          (Gender_Female)
10         (Type of Travel_Personal)
740  (Type of Travel_Personal, Customer Type_Return...
746          (Class_Economy, Customer Type_Returning)
840         (Type of Travel_Personal, Class_Economy)
3184  (Type of Travel_Personal, Class_Economy, Custo...
72          (In-flight Wifi Service_2)
700         (Class_Economy, Gender_Female)
74          (In-flight Wifi Service_3)

          consequents antecedent support \
12  (Satisfaction_Neutral or Dissatisfied)      0.448825
0  (Satisfaction_Neutral or Dissatisfied)      0.507410
10  (Satisfaction_Neutral or Dissatisfied)      0.309236
740  (Satisfaction_Neutral or Dissatisfied)      0.307699
746  (Satisfaction_Neutral or Dissatisfied)      0.343857
840  (Satisfaction_Neutral or Dissatisfied)      0.253971
3184  (Satisfaction_Neutral or Dissatisfied)      0.252566
72  (Satisfaction_Neutral or Dissatisfied)      0.248952
700  (Satisfaction_Neutral or Dissatisfied)      0.228370
74  (Satisfaction_Neutral or Dissatisfied)      0.247801

consequent support  support  confidence  lift  leverage \
12          0.565501  0.364631   0.812413  1.436626  0.110820
0          0.565501  0.289790   0.571115  1.009929  0.002849
10          0.565501  0.277920   0.898731  1.589266  0.103047
740          0.565501  0.276615   0.898978  1.589703  0.102611
746          0.565501  0.274800   0.799169  1.413206  0.080348
840          0.565501  0.228092   0.898103  1.588154  0.084471
3184          0.565501  0.226919   0.898453  1.588774  0.084092
72          0.565501  0.187362   0.752606  1.330866  0.046580
700          0.565501  0.186111   0.814954  1.441119  0.056968
74          0.565501  0.185401   0.748185  1.323048  0.045269

```

	conviction	zhangs_metric
12	2.316253	0.551412
0	1.013091	0.019958
10	4.290559	0.536766
740	4.301056	0.535824
746	2.163507	0.445618
840	4.264083	0.496413
3184	4.278790	0.495808
72	1.756303	0.331017
700	2.348059	0.396686
74	1.725467	0.324608

sort by highest support

```
[54]: results_Satisfaction_Neutral_or_Dissatisfied_support =
      ↪ results_Satisfaction_Neutral_or_Dissatisfied[["antecedents",
      ↪ "consequents", "support"]]
results_Satisfaction_Neutral_or_Dissatisfied_support =
      ↪ results_Satisfaction_Neutral_or_Dissatisfied_support.
      ↪ sort_values(by='support', ascending=False)
results_Satisfaction_Neutral_or_Dissatisfied_support.head(10)
```

```
[54]:
```

	antecedents \	consequents	support
12	(Class_Economy)	(Satisfaction_Neutral or Dissatisfied)	0.364631
0	(Gender_Female)	(Satisfaction_Neutral or Dissatisfied)	0.289790
10	(Type of Travel_Personal)	(Satisfaction_Neutral or Dissatisfied)	0.277920
740	(Type of Travel_Personal, Customer Type_Return...	(Satisfaction_Neutral or Dissatisfied)	0.276615
746	(Class_Economy, Customer Type_Returning)	(Satisfaction_Neutral or Dissatisfied)	0.274800
840	(Type of Travel_Personal, Class_Economy)	(Satisfaction_Neutral or Dissatisfied)	0.228092
3184	(Type of Travel_Personal, Class_Economy, Custoo...	(Satisfaction_Neutral or Dissatisfied)	0.226919
72	(In-flight Wifi Service_2)	(Satisfaction_Neutral or Dissatisfied)	0.187362
700	(Class_Economy, Gender_Female)	(Satisfaction_Neutral or Dissatisfied)	0.186111
74	(In-flight Wifi Service_3)	(Satisfaction_Neutral or Dissatisfied)	0.185401



```
[43]: results_Satisfaction_Neutral_or_Dissatisfied.nlargest(n = 10, columns =_
↳ 'confidence')
```

```
[43]:
```

	antecedents \	consequents	antecedent support	support	confidence	lift	leverage	conviction	zhangs_metric
880	(In-flight Wifi Service_2, Class_Economy)	(Satisfaction_Neutral or Dissatisfied)	0.123758	0.565501	0.979345	1.731819	0.051216	21.035800	0.482255
886	(In-flight Wifi Service_3, Class_Economy)	(Satisfaction_Neutral or Dissatisfied)	0.124190	0.565501	0.974939	1.724028	0.050848	17.337920	0.479514
3198	(Online Boarding_2, In-flight Wifi Service_2, ...)	(Satisfaction_Neutral or Dissatisfied)	0.105246	0.565501	0.972630	1.719944	0.042849	15.874947	0.467822
902	(Online Boarding_2, Ease of Online Booking_2)	(Satisfaction_Neutral or Dissatisfied)	0.118174	0.565501	0.966606	1.709291	0.047400	13.011168	0.470572
926	(Online Boarding_2, In-flight Wifi Service_2)	(Satisfaction_Neutral or Dissatisfied)	0.129588	0.565501	0.962336	1.701741	0.051425	11.536229	0.473760
3212	(Ease of Online Booking_3, In-flight Wifi Serv...)	(Satisfaction_Neutral or Dissatisfied)	0.110768	0.565501	0.961654	1.700535	0.043881	11.330950	0.463265
932	(Online Boarding_3, In-flight Wifi Service_3)	(Satisfaction_Neutral or Dissatisfied)	0.140385	0.565501	0.955221	1.689159	0.054711	9.703104	0.474619
864	(Online Boarding_3, Class_Economy)	(Satisfaction_Neutral or Dissatisfied)	0.116220	0.565501	0.946442	1.673635	0.044273		
914	(Ease of Online Booking_3, Online Boarding_3)	(Satisfaction_Neutral or Dissatisfied)	0.127997	0.565501	0.943767	1.668905	0.048417		
852	(Class_Economy, Ease of Online Booking_2)	(Satisfaction_Neutral or Dissatisfied)	0.118244	0.565501	0.924237	1.634370	0.042418		

864	8.112629	0.455428
914	7.726813	0.459637
852	5.735015	0.440193

sort by highest confidence

```
[55]: results_Satisfaction_Neutral_or_Dissatisfied_confidence =
      ↪results_Satisfaction_Neutral_or_Dissatisfied[["antecedents",
      ↪"consequents", "confidence"]]
results_Satisfaction_Neutral_or_Dissatisfied_confidence =
      ↪results_Satisfaction_Neutral_or_Dissatisfied_confidence.
      ↪sort_values(by='confidence', ascending=False)
results_Satisfaction_Neutral_or_Dissatisfied_confidence.head(10)
```

```
[55]:
      antecedents \
880      (In-flight Wifi Service_2, Class_Economy)
886      (In-flight Wifi Service_3, Class_Economy)
3198 (Online Boarding_2, In-flight Wifi Service_2, ...
902      (Online Boarding_2, Ease of Online Booking_2)
926      (Online Boarding_2, In-flight Wifi Service_2)
3212 (Ease of Online Booking_3, In-flight Wifi Serv...
932      (Online Boarding_3, In-flight Wifi Service_3)
864      (Online Boarding_3, Class_Economy)
914      (Ease of Online Booking_3, Online Boarding_3)
852      (Class_Economy, Ease of Online Booking_2)

      consequents confidence
880  (Satisfaction_Neutral or Dissatisfied)  0.979345
886  (Satisfaction_Neutral or Dissatisfied)  0.974939
3198 (Satisfaction_Neutral or Dissatisfied)  0.972630
902  (Satisfaction_Neutral or Dissatisfied)  0.966606
926  (Satisfaction_Neutral or Dissatisfied)  0.962336
3212 (Satisfaction_Neutral or Dissatisfied)  0.961654
932  (Satisfaction_Neutral or Dissatisfied)  0.955221
864  (Satisfaction_Neutral or Dissatisfied)  0.946442
914  (Satisfaction_Neutral or Dissatisfied)  0.943767
852  (Satisfaction_Neutral or Dissatisfied)  0.924237
```