Laporan Analisis Asosiasi
Data Mining

Kelompok : 6 (Enam)
Anggota : - Muhammad Ihza Mahendra (1301174682)
- Ivan Mulya Saputra (1301173719)
- Rochi Eko Pambudi (1301170761)
Kelas : IF-GAB-04

Dataset :
1. https://archive.ics.uci.edu/ml/datasets/Abscisic+Acid+Signaling+Network
2. https://archive.ics.uci.edu/ml/datasets/Labor+Relations

A. Dataset Abscisic Acid Signaling Network

Dataset ini merupakan kumpulan interaksi data node terhadap plant signaling network. Dalam keterangan yang tertera di link penyedia dataset, dataset ini memiliki 300 simulasi berbeda dengan pengisian data pada masing masing record adalah boolean. Terdapat masing masing 21 record pada tiap simulasi dan terdapat 43 kolom pada dataset ini. Sebelum mencari rule - rule yang bermanfaat menggunakan metode frequent pattern analysis atau analisis asosiasi , Tahap pertama yang kami lakukan terhadap dataset ini adalah sebagai berikut:

- Tahap Exploratory Data Analysis (EDA)
- Tahap Pre-Processing
- Tahap Pencarian Rule Menggunakan Algoritma Apriori

Untuk menentukan rule-rule yang bermanfaat kami melakukan analisis dengan menggunakan python, dengan platform google colaboration. Sebelum masuk kedalam tahap pertama, kami melakukan tahap upload dataset serta merapihkan data agar dapat diproses selanjutnya.

a. Tahap Exploratory Data Analysis (EDA)

Untuk langkah awal, kami melakukan tahap Exploratory Data Analysis (EDA) untuk memahami lebih dalam isi dari dataset yang akan diproses serta untuk keperluan pre-processing selanjutnya. Pada tahap ini pertama kami melakukan visualisasi data sebagai berikut:

	ABA	CLOSURE	Ca	CaATPase	CAIM	CIS	ABH1	GCR	ERA1	PEPC	OST	SPHK	PH	PLD	ROP2	KEV	AGB	RAC	RCN
0	1	0	1	1	1	0	1	1	1	0	1	0	1	1	0	1	1	0	1
1	1	1	0	0	0	0	1	1	1	0	1	1	1	1	0	1	1	0	1
2	1	1	0	0	0	1	1	1	1	0	1	1	1	1	1	0	1	0	1
3	1	1	0	0	0	0	1	1	1	0	1	1	1	1	1	0	1	0	1
4	1	1	0	0	0	0	1	1	1	0	1	1	1	1	1	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5451	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	0	1
5452	1	1	0	1	0	1	1	1	1	0	1	1	1	1	1	1	1	0	1
5453	1	1	1	0	0	1	1	1	1	0	1	1	1	1	1	0	1	0	1
5454	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	0	1
5455	1	1	0	1	0	1	1	1	1	0	1	1	1	1	1	0	1	0	1

5456 rows × 43 columns

Dari informasi yang tertera di website penyedia dataset, bahwa terdapat 300 percobaan serta 21 record di masing - masing percobaan yang berarti bahwa seharusnya dataset memiliki 300 * 21 = 6300 record, namun saat dianalisis ternyata dataset hanya memiliki 5456 record. Dataset juga bernilai boolean 0 dan 1 di semua value, jika 1 menandakan bahwa interaksi pada suatu node terjadi dan 0 adalah sebaliknya.

Lalu selanjutnya adalah pencarian missing value, duplikasi serta nilai abnormal selain boolean data pada dataset. Untuk Missing value, hasilnya adalah terdapat 0% missing value atau tidak ada missing value pada dataset ini.

```
percent_missval=round((new_df.isnull().sum().sum() / new_df.count().sum()) * 100, 2)
print('total missing values : {}%'.format(percent_missval))

total missing values : 0.0%
```

Kedua adalah pegecekan duplikasi data yang terjadi pada dataset, dimana terdapat 4014 duplikasi data yang harus ditangani untuk mendapatkan hasil serta proses pencarian rule yang efektif.

```
pd.set_option('display.max_rows',  None)
duplicate = new_df[new_df.duplicated()]
print('duplikasi data: {}'.format(len(duplicate)))

duplikasi data: 4014
```

Dan terakhir adalah pengecekan data abnormal selain nilai 0 atau 1 dimana tidak ada data selain 0 dan 1.

.

c. Tahap Pre-Processing

Selanjutnya adalah tahap processing berdasarkan informasi dari tahap pertama EDA. Pada tahap ini kami melakukan Pre-processing agar data berkualitas serta siap dan cocok untuk dimasukkan kedalam inputan algoritma Apiori. Sebagaimana informasi dari tahap EDA, kami akan melakukan penghapusan dataset guna meningkatkan tingkat efetifitas dari algoritma Apriori mengingat permasalahan dalam pencarian rule adalah kompleksitas komputasinya. Kami memutuskan untuk menghapus data duplikat karena data sudah terlalu banyak dan memperlambat proses mining. Berikut adalah kondisi dari dataset yang telah kami hapus duplikasi datanya:

	ABA	CLOSURE	Ca	CaATPase	CAIM	CIS	ABH1	GCR	ERA1	PEPC	OST	SPHK	PH	PLD	ROP2	KEV
0	1	0	1	1	1	0	1	1	1	0	1	0	1	1	0	1
1	1	1	0	0	0	0	1	1	1	0	1	1	1	1	0	1
2	1	1	0	0	0	1	1	1	1	0	1	1	1	1	1	0
21	1	0	0	0	1	1	1	0	1	1	1	1	0	1	1	0
22	1	0	0	0	0	0	1	1	1	0	1	1	1	1	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5439	1	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1
5440	1	0	1	0	0	0	1	1	1	0	1	1	1	1	1	1
5441	1	1	0	1	0	0	1	1	1	0	1	1	1	1	1	0
5444	1	1	0	1	0	1	1	1	1	0	1	1	1	1	1	0
5445	1	1	1	0	0	1	1	1	1	0	1	1	1	1	1	0

1150 rows × 43 columns

Data hasil penghapusan duplikasi sekarang berjumlah 1150 record. Setelah melakukan pre-processing data menjadi lebih berkualitas dan siap untuk masuk kedalam tahap selanjutnya.

d. Tahap Pencarian Rule Menggunakan Algoritma Apriori

Kami memutuskan untuk menggunakan Algoritma Apriori karena algoritma ini cocok dalam masalah pencarian rule dengan memaksimalkan pencarian / penelusuran yang lebih baik dibandingkan brute force. Algoritma ini juga lebih populer dalam pencarian rule yang bermanfaat. Setelah melakukan preprocessing dan cocok untuk input dari algoritma apriori selanjutnya kami melakukan analisis Algoritma Apriori serta mencari Frequent Itemset Generation. Untuk parameter yang diambil kami mengambil minimal support yaitu 0.45 , dimana kami mengasumsikan terdapat 475 data yang muncul pada simulasi yang muncul pada dataset. Pengambilan nilai minimal support ini juga dipilih karena faktor komputasi yang tinggi mengingat terdapat 43 item atau kolom yang akan diproses atau dicari rulenya.

Berdasarkan rumus berikut kami menghitung kompleksitas serta banyaknya rule yang dapat diambil.

	support	itemsets
0	1.000000	(ABA)
1	0.456522	(CLOSURE)
2	0.454783	(Ca)
3	0.477391	(CaATPase)
4	0.584348	(CIS)
...	...	...
292573	0.483478	(Actin, InsPK, AnionEM, DEPOLAR, GPA, S1P, NIA...
292574	0.456522	(PA, Actin, DEPOLAR, IP6, GPA, S1P, NIA12, KOUT)
292575	0.475652	(Actin, AnionEM, DEPOLAR, IP6, GPA, S1P, NIA12...
292576	0.461739	(PA, InsPK, Actin, DEPOLAR, IP6, GPA, S1P, KOUT)
292577	0.480870	(Actin, InsPK, AnionEM, DEPOLAR, IP6, GPA, S1P...

292578 rows × 2 columns

$$R = \sum_{k=1}^{d-1} \left[\binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \right]$$

Dari rumus tersebut dapat dihitung bahwa 3^43 - 2^44 + 1 = 3,282569e20 maksimal rule. Hal ini dapat memberatkan pencarian , maka minimal support 0.45 serta max length yang dipilih untuk itemset adalah 8 , jika lebih dari itu kami mendapatkan crash saat running program.

Selanjutnya adalah tahap Rule Generation dari hasil frequent berdasarkan metric lift dan berikut adalah hasilnya:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
0	(CLOSURE)	(ABA)	0.456522	1.000000	0.456522	1.0	1.000000
1	(Ca)	(ABA)	0.454783	1.000000	0.454783	1.0	1.000000
2	(CaATPase)	(ABA)	0.477391	1.000000	0.477391	1.0	1.000000
3	(CIS)	(ABA)	0.584348	1.000000	0.584348	1.0	1.000000
4	(ABH1)	(ABA)	0.877391	1.000000	0.877391	1.0	1.000000
...	...	...	...	...	...	...	...
435841	(Actin, RCN, AnionEM, DEPOLAR, GPA, S1P, KOUT)	(InsPK)	0.493913	0.881739	0.493913	1.0	1.134122
435842	(PA, Actin, IP6, DEPOLAR, GPA, S1P, NIA12)	(InsPK)	0.466957	0.881739	0.466957	1.0	1.134122
435843	(AnionEM, IP6, DEPOLAR, GPA, S1P, NIA12, KOUT)	(InsPK)	0.479130	0.881739	0.479130	1.0	1.134122
435844	(Actin, IP6, DEPOLAR, GPA, S1P, NIA12, KOUT)	(InsPK)	0.561739	0.881739	0.561739	1.0	1.134122
435845	(PA, Actin, IP6, DEPOLAR, GPA, NIA12, KOUT)	(InsPK)	0.460000	0.881739	0.460000	1.0	1.134122

435846 rows × 9 columns

Terdapat 435846 rule yang didapat atau dimining dalam dataset ini.

C. Dataset Labor Relations

Untuk mendapatkan rule dengan analisis asosiasi dari data tersebut kami melakukan beberapa tahapan yang dilakukan yaitu:

1. Data Collection
2. Exploratory Data Analysis
3. Data Preprocessing
4. Pencarian rule menggunakan algoritma asosiasi

a. Data Collection

Pada tahap ini dilakukan proses load data yang akan dilakukan pemrosesan analisis asosiasi, data yang digunakan adalah Dataset Labor Relations yang merupakan data yang menjelaskan tentang semua kesepakatan bersama yang dicapai di sektor bisnis dan layanan pribadi untuk penduduk setempat (Canada) dengan setidaknya 500 anggota (guru, perawat, staf universitas, polisi, dll) pada tahun 1987 sampai kuartal pertama 1988.
Atribut pada data Labor Relations:

dur: duration of agreement
wage1 : wage increase in first year of contract
wage2 : wage increase in second year of contract
wage3 : wage increase in third year of contract
cola : cost of living allowance
hours : number of working hours during week
pension : employer contributions to pension plan
stby_pay : standby pay
shift_diff : shift differential : supplement for work on II and III shift
educ_allw.boolean : education allowance
holidays : number of statutory holidays
vacation : number of paid vacation days
lngtrm_disabil.boolean : employer's help during employee long term disability
dntl_ins : employers contribution towards the dental plan
bereavement.boolean : employer's financial contribution towards the covering the costs of bereavement
empl_hplan : employer's contribution towards the health plan

```
[2] columns = ["dur","wage1","wage2","wage3","cola","hours","pension","stby_pay","shift_diff",
               "educ_allw","holidays","vacation","lngtrm_disabil","dntl_ins","bereavement","empl_hplan","class"]

    df =pd.read_csv('labor-neg.data', header=None, names=columns, skipinitialspace=True)
```

```
[3] df.head()
```

	dur	wage1	wage2	wage3	cola	hours	pension	stby_pay	shift_diff	educ_allw	holidays	vacation	lngtrm_disabil	dntl_ins	bereavement	empl_hplan	class
0	1	5.0	?	?	?	40	?	?	2	?	11	average	?	?	yes	?	good
1	2	4.5	5.8	?	?	35	ret_allw	?	?	yes	11	below average	?	full	?	full	good
2	?	?	?	?	?	38	empl_contr	?	5	?	11	generous	yes	half	yes	half	good
3	3	3.7	4.0	5.0	tc	?	?	?	?	yes	?	?	?	?	yes	?	good
4	3	4.5	4.5	5.0	?	40	?	?	?	?	12	average	?	half	yes	half	good

b. Exploratory Data Analysis (EDA)

Pada tahap EDA dilakukan untuk menganalisis data yang ada untuk mendapatkan informasi yang dapat digunakan pada tahap preprocessing data, di tahap ini dilakukan pengecekan tipe data, cek missing values dan proses visualisasi untuk kuantitas data yang ada.

#Cek missing values:

Pada tahap ini dilakukan untuk mengetahui missing values pada dataset Labor, untuk proses handling yang dilakukan adalah dengan menggunakan imputing dengan modus, hal tersebut didasarkan pada data tersebut merupakan data Multivariate yang memiliki tipe data object dan setiap atribut memiliki *missing values* sehingga akan lebih mudah jika menggunakan modus untuk inputting value nya.

#Mengubah tipe data

mengubah tipe data dari masing-masing kolom yang sebelumnya object menjadi tipe data yang sesuai dengan values yang ada yang akan mempermudah dalam proses dalam melakukan analisis rule menggunakan algoritma apriori.

```
[10] df.head()
```

	dur	wage1	wage2	wage3	cola	hours	pension	stby_pay	shift_diff	educ_allw	holidays	vacation	lngtrm_disabil	dntl_ins	bereavement	empl_hplan	class
0	1	5.0	4.0	5.0	none	40	none	2	2	no	11	average	yes	half	yes	full	good
1	2	4.5	5.8	5.0	none	35	ret_allw	2	3	yes	11	below average	yes	full	yes	full	good
2	2	2.0	4.0	5.0	none	38	empl_contr	2	5	no	11	generous	yes	half	yes	half	good
3	3	3.7	4.0	5.0	tc	40	none	2	3	yes	11	below average	yes	half	yes	full	good
4	3	4.5	4.5	5.0	none	40	none	2	3	no	12	average	yes	half	yes	half	good

```
[11] df[["dur","stby_pay","shift_diff","holidays","hours"]] = df[["dur","stby_pay","shift_diff","holidays","hours"]].astype('int')
     df[["cola","pension","educ_allw","vacation","lngtrm_disabil","dntl_ins","bereavement","empl_hplan","class"]] = df[["cola","pension","educ_allw","vacation","lngtrm_disabil","dntl_ins","bereaveme
     df[["wage1","wage2","wage3"]] =df[["wage1","wage2","wage3"]].astype('float')
```

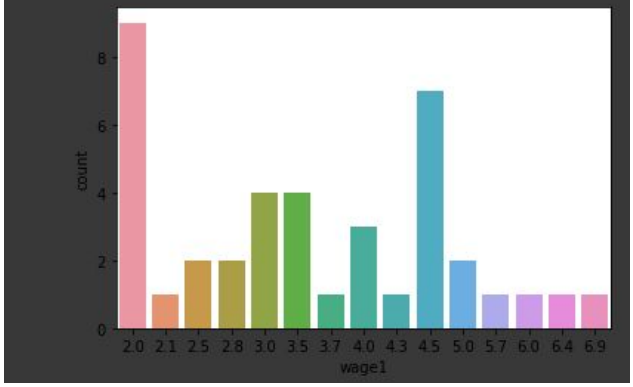
```
[12] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40 entries, 0 to 39
Data columns (total 17 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   dur             40 non-null     int64
 1   wage1           40 non-null     float64
 2   wage2           40 non-null     float64
 3   wage3           40 non-null     float64
 4   cola            40 non-null     category
 5   hours           40 non-null     int64
 6   pension         40 non-null     category
 7   stby_pay        40 non-null     int64
 8   shift_diff      40 non-null     int64
 9   educ_allw       40 non-null     category
 10  holidays        40 non-null     int64
 11  vacation        40 non-null     category
 12  lngtrm_disabil  40 non-null     category
 13  dntl_ins        40 non-null     category
 14  bereavement     40 non-null     category
 15  empl_hplan      40 non-null     category
 16  class           40 non-null     category
dtypes: category(9), float64(3), int64(5)
memory usage: 3.9 KB
```

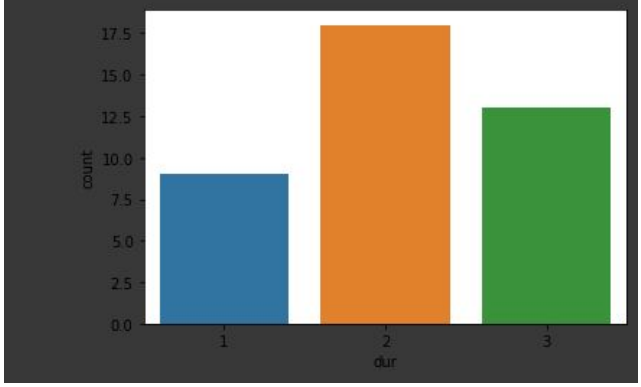
#Visualisasi Data

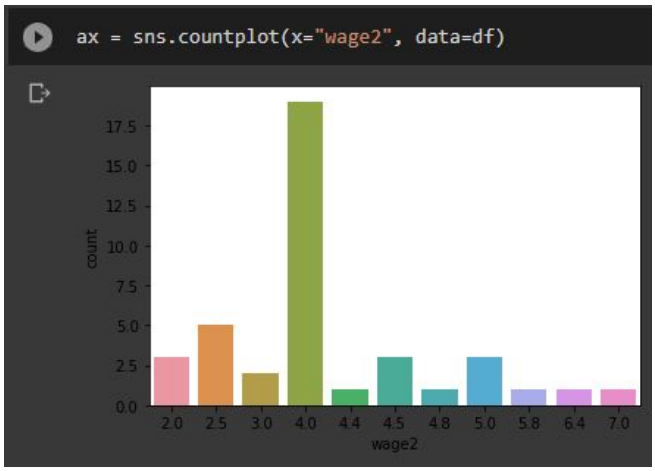
Pada tahap visualisasi digunakan untuk mengetahui persebaran nilai dan frekuensi untuk masing-masing atribut yang ada.

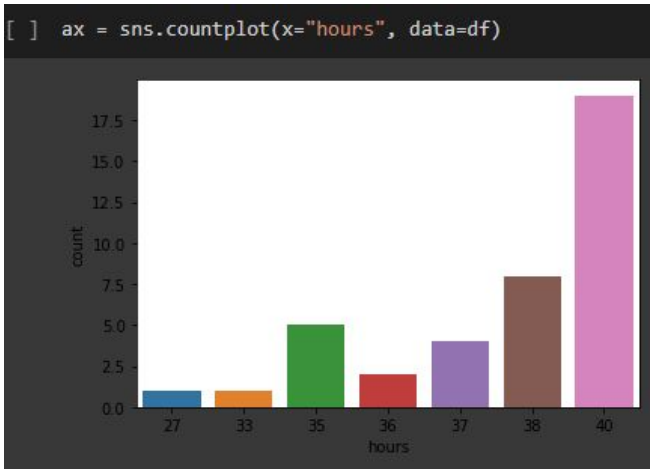
```
[ ] ax = sns.countplot(x="wage1", data=df)
```




```
[ ] ax = sns.countplot(x="dur", data=df)
```







c. Preprocessing Data

Setelah mendapatkan informasi dari visualisasi yang dilakukan dan handling *missing values*, tahapan selanjutnya adalah preprocessing data, di tahap ini akan dilakukan proses *Discretization* dan proses *encode* data, berdasarkan data yang diperoleh setelah tahapan EDA values yang dimiliki belum ke dalam bentuk 0/1 sehingga belum dalam digunakan untuk proses pencarian rule menggunakan algoritma apriori, karena algoritma apriori hanya bisa menerima inputan dalam bentuk 0/1.

```
[ ] df.head()
```

	dur	wage1	wage2	wage3	cola	hours	pension	stby_pay	shift_diff	educ_allw	holidays	vacation	lngtrm_disabil	dntl_ins	bereavement	empl_hplan	class
0	1	5.0	4.0	5.0	none	40	none	2	2	no	11	average	yes	half	yes	full	good
1	2	4.5	5.8	5.0	none	35	ret_allw	2	3	yes	11	below average	yes	full	yes	full	good
2	2	2.0	4.0	5.0	none	38	empl_contr	2	5	no	11	generous	yes	half	yes	half	good
3	3	3.7	4.0	5.0	tc	40	none	2	3	yes	11	below average	yes	half	yes	full	good
4	3	4.5	4.5	5.0	none	40	none	2	3	no	12	average	yes	half	yes	half	good

#Proses discretization

Mengubah values dengan tipe numerik ke dalam bentuk range nilai.

```
[14] pd.qcut(df['dur'], q=2).tail()

     35    (0.999, 2.0]
     36    (0.999, 2.0]
     37    (0.999, 2.0]
     38      (2.0, 3.0]
     39    (0.999, 2.0]
     Name: dur, dtype: category
     Categories (2, interval[float64]): [(0.999, 2.0] < (2.0, 3.0]]

[15] df['dur_range'] = pd.qcut(df['dur'], q=2, labels=['<=2', '>2'])

[16] pd.qcut(df['wage1'], q=3).tail()

     35    (1.999, 2.8]
     36    (1.999, 2.8]
     37    (1.999, 2.8]
     38    (1.999, 2.8]
     39      (4.3, 6.9]
     Name: wage1, dtype: category
     Categories (3, interval[float64]): [(1.999, 2.8] < (2.8, 4.3] < (4.3, 6.9]]

[17] df['wage1_range'] = pd.qcut(df['wage1'], q=3, labels=['<=2.8', '>2 <=4.3','>4.3'])
```

#Cek Duplicated

Berdasarkan hasil pengecekan data yang dilakukan, tidak ada row yang memiliki duplikat dari 40 rows yang ada.

```
[37] df_new.duplicated().sum()

     0
```

Setelah dilakukan preprocessing yang data yang ada dapat sudah siap digunakan untuk pencarian rules menggunakan algoritma apriori.

```
[34] df.shape

     (40, 17)

[35] df_new = pd.get_dummies(df, columns=df.columns)

[36] df_new.head()
```

	cola_none	cola_tc	cola_tcf	pension_empl_contr	pension_none	pension_ret_allw	educ_allw_no	educ_allw_yes	vacation_average	vacation_below average	vacation_generous	lngtrm_disabil_no	lngtrm_di
0	1	0	0	0	1	0	1	0	1	0	0	0	
1	1	0	0	0	0	1	0	1	0	1	0	0	
2	1	0	0	1	0	0	1	0	0	0	1	0	
3	0	1	0	0	1	0	0	1	0	1	0	0	
4	1	0	0	0	1	0	1	0	1	0	0	0	

d. Pencarian Rule dengan Algoritma Apriori

Pada tahap terakhir adalah melakukan pencarian rule berdasarkan data yang telah dilakukan pemrosesan, untuk proses pencarian rule kami menggunakan library mlxtend untuk menggunakan algoritma apriori.

Langkah 1: Frequent Itemset Generation

Parameter yang digunakan pada kelompok kami menggunakan semua atribut yang ada yaitu sebanyak 43 kolom, Selain itu kami menggunakan min_support = **0.15** pada algortima apriori dengan alasan bahwa nilai tersebut merupakan nilai yang cukup ideal berdasarkan pemrosesan dan percobaan yang telah kami lakukan, selain itu dengan adanya keterbatasan resources perangkat yang kami miliki. Dengan parameter tersebut kami mendapatkan item set yang memenuhi syarat sebanyak **26.779.**

```
[40] #Apriori min support
     min_support = 0.15

     #Max lenght of apriori n-grams

     frequent_items = apriori(df_new, use_colnames=True, min_support=min_support)
```

```
[41] frequent_items
```

	support	itemsets
0	0.750	(cola_none)
1	0.150	(cola_tc)
2	0.175	(pension_empl_contr)
3	0.750	(pension_none)
4	0.825	(educ_allw_no)
...	...	...
26774	0.150	(lngtrm_disabil_yes, bereavement_yes, dur_rang...
26775	0.150	(lngtrm_disabil_yes, bereavement_yes, dur_rang...
26776	0.150	(lngtrm_disabil_yes, dntl_ins_half, bereavemen...
26777	0.150	(lngtrm_disabil_yes, shift_diff_range_>3, bere...
26778	0.150	(lngtrm_disabil_yes, bereavement_yes, dur_rang...

26779 rows × 2 columns

Langkath 2: *Rule Generation*

Pada tahap ini bertujuan membentuk aturan dengan nilai confidence yang tinggi dari frequent itemset yang telah diperoleh sebelumnya. Aturan ini disebut strong rules, berikut adalah rule yang diperoleh dengan vacation average sebagai antecedent dan cola_none yang memiliki nilai confidence paling tinggi yaitu sebesar **1**. Dengan rule yang diperoleh sebanyak **1.988.719.**

```
[ ] rules = association_rules(frequent_items, metric ="lift", min_threshold = 1)
```

```
rules.head(10).sort_values(by='confidence', ascending=False)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
6	(vacation_average)	(cola_none)	0.275	0.750	0.275	1.000000	1.333333	0.06875	inf
9	(cola_none)	(lngtrm_disabil_yes)	0.750	0.875	0.675	0.900000	1.028571	0.01875	1.250000
0	(pension_empl_contr)	(cola_none)	0.175	0.750	0.150	0.857143	1.142857	0.01875	1.750000
4	(cola_none)	(educ_allw_no)	0.750	0.825	0.625	0.833333	1.010101	0.00625	1.050000
8	(lngtrm_disabil_yes)	(cola_none)	0.875	0.750	0.675	0.771429	1.028571	0.01875	1.093750
2	(pension_none)	(cola_none)	0.750	0.750	0.575	0.766667	1.022222	0.01250	1.071429
3	(cola_none)	(pension_none)	0.750	0.750	0.575	0.766667	1.022222	0.01250	1.071429
5	(educ_allw_no)	(cola_none)	0.825	0.750	0.625	0.757576	1.010101	0.00625	1.031250
7	(cola_none)	(vacation_average)	0.750	0.275	0.275	0.366667	1.333333	0.06875	1.144737
1	(cola_none)	(pension_empl_contr)	0.750	0.175	0.150	0.200000	1.142857	0.01875	1.031250

```
[ ] rules.shape
```

```
(1988718, 9)
```