ASSOCIATION RULES

- Metode yang bertujuan mencari **pola yang sering muncul** di antara banyak transaksi, dimana setiap transaksi terdiri dari beberapa ***item***.
- Pada transaksi terdapat ***item* X** terdapat kemungkinan ada ***item* Y** juga didalamnya, X→Y (X dan Y: *disjoin itemset*, dinotasikan: X ∪ Y.
    - Kumpulan dari transaksi-transaksi ini disebut dengan ***itemset*** (dinotasikan dengan I_k (k = 1, 2, … m)).
        - *Itemset* yang mempunyai item sebanyak k, maka disebut **k-*itemset***.
- Association rule menghasilkan ***rules*** yang menentukan seberapa **besar hubungan antar X dan Y**.
    - Untuk mengukur 'besar hubungan', ada dua measurement yaitu: ***support*** dan ***confidence***.

Support

- *Support*: probabilitas konsumen membeli beberapa produk secara bersamaan dari jumlah seluruh transaksi.
- *Support* merupakan **kemungkinan X dan Y muncul bersamaan**, dinotasikan sebagai:

$$Support\ (X,Y) = \frac{\Sigma\ transaksi\ yang\ mengandung\ X\ dan\ Y}{\Sigma\ transaksi} = P\ (X \cap Y)$$

- Minsup *threshold* (ambang/batas) dapat ditentukan berdasarkan pengetahuan user mengenai domain (***A minimum support threshold is given in the problem or it is assumed by the user***).

Confidence

- *Confidence*: probabilitas kejadian beberapa produk yang dibeli bersamaan, dimana salah saru produk sudah pasti dibeli.
- *Confidence* merupakan **kemungkinan munculnya Y ketika X juga muncul**, dinotasikan sebagai:

$$Confidence\ (X \rightarrow Y) = \frac{\Sigma\ transaksi\ yang\ mengandung\ X\ dan\ Y}{\Sigma\ transaksi\ yang\ mengandung\ X} = \frac{Support\ (X,Y)}{Support\ (X)}$$
$$= P\ (Y \mid X) = \frac{P\ (X \cap Y)}{P\ (X)}$$

- Nilai *confidence* memiliki rentang dari 0 sampai 1, dimana 0 mengindikasikan bahwa Y tidak pernah dibeli ketika X dibeli, dan 1 mengindikasikan bahwa Y selalu dibeli saat X dibeli.
- Nilai ***confidence* memiliki arah** (*directional*).
- Nilai *confidence* tidak menunjukkan bahwa ada hubungan antara *item*, bisa saja karena kebetulan. Untuk mengukur apakah ada hubungan antara pembelian *item*, metrik yang digunakan adalah *lift.*
- *Support and Confidence are not enough to determine how interesting a rule is. We need Correlation analysis to make it sure (Chi-Square test or Lift test).*

Lift

- *Lift*: mengindikasikan bahwa ada hubungan antara X dan Y, atau apakah dua *item* muncul bersamaan dalam transaksi yang sama secara kebetulan (*random*).
- Tidak seperti metrik *confidence* yang nilainya dapat bervariasi tergantung arahnya (contoh: *confidence* (X→Y) bisa berbeda dari *confidence* (Y→X)**),** ***lift*** **tidak memiliki arah** (*direction*). Artinya *lift* (X→Y) selalu sama dengan *lift* (Y→X).

$$Lift\ (X \rightarrow Y) = Lift\ (Y \rightarrow X) = \frac{Confidence\ (X \rightarrow Y)}{\Sigma\ transaksi\ yang\ mengandung\ Y} = \frac{P\ (Y \mid X)}{P\ (Y)}$$
$$= \frac{P\ (X \cap Y)}{P\ (X)\ P\ (Y)}$$

- *Lift* = 1, berarti **tidak ada** hubungan antara X dan Y (contoh: X dan Y muncul bersamaan karena kebetulan).
- *Lift* > 1, berarti ada hubungan positif antara X dan Y (contoh: X dan Y muncul bersamaan lebih sering daripada acak).
- *Lift* < 1, berarti ada hubungan negative antara X dan Y (contoh: X dan Y muncul bersamaan lebih jarang daripada acak).

Frequent Itemset

- *Frequent Itemset* adalah itemset yang mempunyai support ≥ minimum support yang ditetapkan oleh user.

Apriori

- Algortima Apriori (*Apriori Algorithm*) digunakan untuk **mencari *frequent itemset*** yang memenuhi ***minimum support*** (minsup) kemudian **mendapatkan *rule*** yang memenuhi ***minimum confidence*** (minconf) dari *frequent itemset* tadi.
    - Algoritma ini **mengontrol jumlah kandidat *frequent itemset*** dengan *support-based pruning* (pemangkasan jumlah *itemset* berdasarkan nilai *support*) yang tidak menarik dengan menetapkan minsup.
- Prinsip apriori: **bila *itemset* digolongkan sebagai *frequent itemset*** (memiliki support lebih dari minsup yang ditetapkan), **maka semua *subset*nya juga termasuk golongan *frequent itemset*, dan sebaliknya**.
- Cara kerja:
    - Tentukan minimum support.
    - Iterasi 1 : hitung item-item dari *support* (transaksi yang memuat seluruh item) dengan men-scan database untuk 1-*itemset*, setelah 1-*itemset* didapatkan, dari 1-*itemset* apakah diatas *minimum support*, apabila telah memenuhi *minimum support*, 1-*itemset* tersebut akan menjadi pola *frequent* tinggi.
    - Iterasi 2 : untuk mendapatkan 2-*itemset*, harus dilakukan kombinasi dari k-*itemset* sebelumnya, kemudian scan database lagi untuk hitung item-item yang memuat *support*. *Itemset* yang memenuhi *minimum support* akan dipilih sebagai pola *frequent* tinggi dari kandidat.
    - Tetapkan nilai k-*itemset* dari *support* yang telah memenuhi *minimum support* dari k-*itemset*.

- Lakukan proses untuk iterasi selanjutnya hingga tidak ada lagi k-*itemset* yang memenuhi *minimum support*.

Sources

Fauzy, M., & Asror, I. (2016). Penerapan metode association rule menggunakan algoritma Apriori pada simulasi prediksi hujan wilayah kota Bandung. *Jurnal Ilmiah Teknologi Infomasi Terapan*, *2*(3).

Ilayaraja, M., & Meyyappan, T. (2013). Mining medical data to identify frequent diseases using Apriori algorithm. *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, 194–199. IEEE.

Wandi, N., Hendrawan, R. A., & Mukhlason, A. (2012). Pengembangan sistem rekomendasi penelusuran buku dengan penggalian association rule menggunakan algoritma apriori (studi kasus Badan Perpustakaan dan Kearsipan Provinsi Jawa Timur). *Jurnal Teknik ITS*, *1*(1), A445–A449.

https://towardsdatascience.com/instacart-market-basket-analysis-part-3-which-sets-of-products-should-be-recommended-to-shoppers-9651751d3cd3

https://www.slideshare.net/dedidarwis/algoritma-apriori

http://www.codeding.com/articles/apriori-algorithm