پیاده سازی الگوریتم های قواعد انجمنی با استفاده از روش های یادگیری ماشین و تحلیل سیستم های داده حجیم

گردآورنده : حسین جمشیدیان

نام درس: تحلیل داده های حجیم

استاد درس : جناب آقای دکتر ملک

بهمن 1401

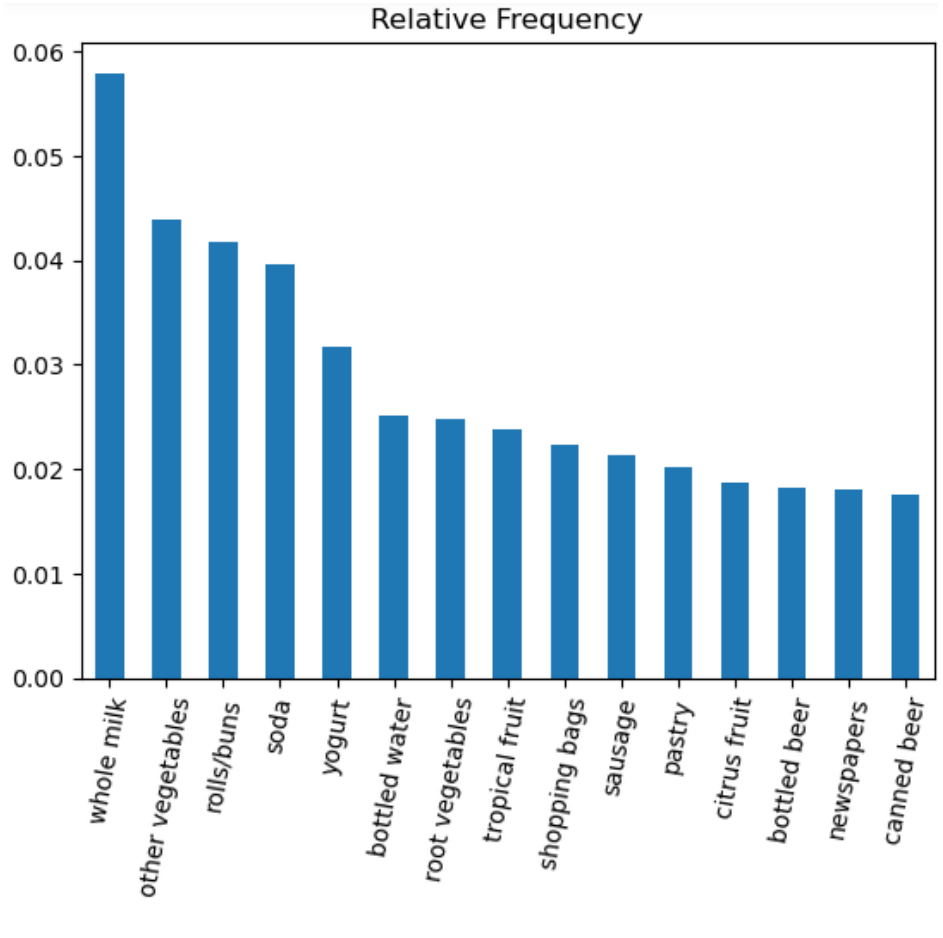
معرفی

- داده هایی مربوط به خرید از مغازه ای در یک ماه است که داده ها از کگل با آدرس زیر در دسترس است :

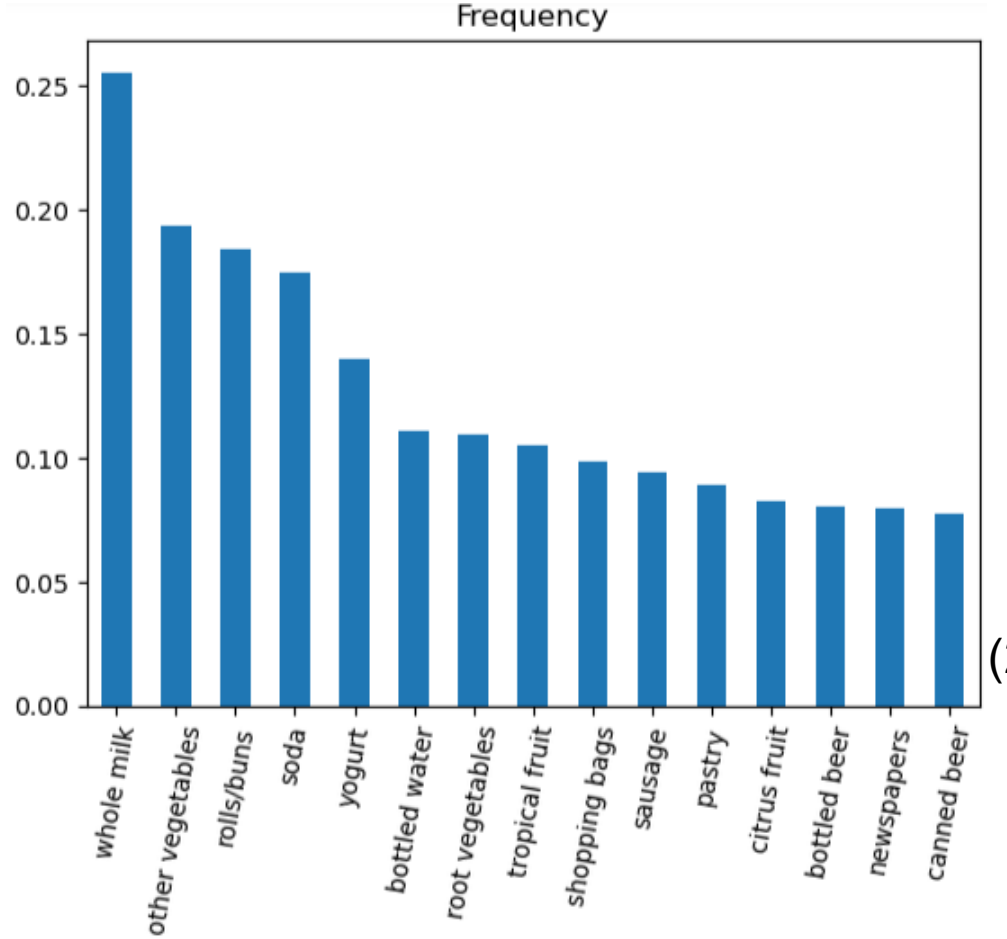
https://www.kaggle.com/datasets/muhammadglennyunifer/groceries-data-for-market-basket-analysis

- داده ها شامل ۹۸۳۵ رکورد است که که با کمی داده کاوی می بینیم که ۱۶۹ کالای منحصر به فرد در داده های ما وجود دارد .
- بیشترین آیتمی که دیده می شود شیر است با ۲۵۱۳ تکرار که تقریبا ۵٪ از کل آیتم ها است .
- شکل (۱) صفحه بعد فراوانی نسبی و شکل (۲) فراوانی مربوط ۱۵ آیتم پرطرفدار را نشان می دهد .

نمودار



شکل(1)



شکل(2)

پیش پردازش

- قبل از شروع مدل سازی باید ابتدا مقادیر NaN را حذف کنیم . به نوعی فقط مقادیر آیتم های موجود را تحت عنوان trans برای ادامه کار در نظر می گیریم .
- مدل سازی خود را ابتدا با مدل Apriori شروع می کنیم . سه مقدار مهم برای مدل بندی مورد استفاده قرار می گیرد که بصورت زیر است :

$$\boldsymbol{support}(M) = M \text{ شامل های آیتم لیست/مشتریان های آیتم لیست کل}$$

$$\boldsymbol{Confidence}(M_1 \rightarrow M_2) = M_2 \text{ و } M_1 \text{ شامل های کاربر آیتم لیست/} M_1 \text{ شامل ها آیتم لیست کل}$$

$$\boldsymbol{Lift}(M_1 \rightarrow M_2) = {Confidence(M_1 \rightarrow M_2)}/{support(M_2)}$$

Apriori

- از کتابخانه apyoriاستفاده می کنم .
- مقدار min_support را با این فرضیه که هر آیتم هر روزحداقل ۳ مرتبه در سبدهای خرید موجود باشد و همچنین ۷ روز هفته داریم پس حداقل ساپورت را $\frac{3\times7}{9835} \cong .003$ درنظر می گیرم.
- کمترین مقدار اطمینان را با توجه به مطالب درس بزرگتر از 5. در نظر می گیرم .
- و برای پارامترlift نیز طبق تجربیات حداقل مقدار3. بنظر مناسب می رسد .
- باقی پارامتر ها نیز برای تعیین مقدار k در K_candidate استفاده می شود.

Apriori

- خروجی مدل را براساس پارامترLift مرتب می کنیم .نکته ای که وجود دارد برای کاندیداهای بیشتر از ۲ ممکن است مدل در سمت راست قاعده مقدارnan را نیز جز قواعد در نظر بگیرد که آن قاعده را به کل در نظر نمی گیرم. بنابراین یک تابع با عنوان clean می سازم که قواعد نامعتبر را حذف می کند.
- به عنوان تحلیل می توان گفت که اگر فردی (سبزیجات ، مرکبات و میوه های گرمسیری) خریداری کند پس با احتمال 55%. می توان اطمینان داشت که آن مشتری (شیر و نوع های دیگر سبزیجات) را درسبد خود قرار می دهد .

	left hand side	right hand side	support	confidence	lift
0	(citrus fruit, tropical fruit, root vegetables)	(other vegetables, whole milk)	0.003152	0.553571	7.397249
3	(whole milk, herbs)	(root vegetables,)	0.004169	0.539474	4.949369

Eclat

- نکته مهم دیگری که وجود دارد مقدار ساپورتی که هرقاعده می دهد در واقع می گوید که هر سبد شامل j، i($i \rightarrow j$) شامل S درصد تراکنش های ما است به عنوان مثال در سطر ۱ جدول صفحه قبل، سبدی شامل مرکبات ،سبزیجات شیر تقریبا بیشتر از003. درصد از تراکنش های ما را شامل می شود .
- Eclat یک مدل ساده شده از Apriori است که پارامترهای confidence,lift را ندارد و فقط بر اساس support تصمیم گیری می کند . مدل Apriori از مدل Eclat تفسیرپذیر تر وقابل قبول تر است و اطلاعات بیشتری را در اختیار ما قرار می دهد. ۲ قاعده از ۱۰۰ قاعده با روش Eclat بصورت زیر است:

	products1	products2	support
0	(root vegetables, tropical fruit)	(other vegetables,)	0.012303
1	(nan, root vegetables, tropical fruit)	(other vegetables,)	0.012201

FPGrowth

- حال می خواهم با استفاده از الگوریتمی که برای اسپارک نیز وجود دارد با استفاده از این ابزار ، مدل FPGrowth را پیاده سازی کنم .
- از کتابخانه pyspark.ml.fpm استفاده می کنیم. ابتدا پیکربندی های اولیه مربوط به اسپارک را پیاده سازی می کنم. حال داده ها را آپلود می کنم . می بینید که از f که از کتابخانه pyspark.sql گرفته می شود و یک سری توابع built-in را می توان از آن برداشت که f. monotonically_increasing_id در واقع به انتهای هر سبد یک شماره منحصر بفرد می دهد و آن را بعنوان آیدی در نظر می گیرد .
- یک چارچوب داده ای با نام df_basket می سازم که شامل دو ستون با عنوان basket, id است که مقادیر

FPGrowth

- هر سبد را در این چارچوب داده می ریزم.
- با استفاده از f.array_except می توان مقادیر null را حذف کرد .
- برای مدل سازی کافیست دو پارامتر minSupport و minConfidece را تعیین کنم . در ادامه ۲۲۲۶ آیتم پرتکرار kتایی بدست می آید .
- حال با استفاده از دستوری فقط قواعدی که بالای 51. درصد اطمینان دارند را جدا می کنیم .که بعنوان اولین قاعده می توان گفت که اگر فردی سبزیجات ، مرکبات و میوه های گرمسیری خریده باشد با احتمال 78. درصد باقی سبزیجات را نیز در سبد خود قرار داده است .

FPGrowth

- سرعت الگوریتم FpGrowth بسیار بیشتر از Apriori عمل می کند .

antecedent	consequent	confidence	lift	support
[citrus fruit, tropical fruit, root vegetables]	[other vegetables]	0.7857142857142857	4.060693641618497	0.004473817996949669
[brown bread, root vegetables, other vegetables]	[whole milk]	0.775	3.0330779944289694	0.003152008813421454
[onions, butter]	[whole milk]	0.75	2.9352367688022283	0.003050330452465684
[citrus fruit, tropical fruit, root vegetables, whole milk]	[other vegetables]	0.8857142857142857	4.577509196006305	0.003152008813421454
[tropical fruit, root vegetables, yogurt, other vegetables]	[whole milk]	0.7142857142857143	2.7954635893354554	0.003558718861209964
[sugar, domestic eggs]	[whole milk]	0.7142857142857143	2.7954635893354554	0.003558718861209964
[butter, tropical fruit, yogurt]	[whole milk]	0.7333333333333333	2.8700092850510672	0.003355363497712252
[curd, tropical fruit, yogurt]	[whole milk]	0.75	2.9352367688022283	0.003965429588205389

SON

- برای باقی الگوریتم ها من جمله SON algorithm ابزاری وجود ندارد و سعی بر پیاده سازی از پایه کرده ام ولی به نتیجه مورد دلخواه نرسیدم ولی ابتدا داده ها را تمیز کرده ام و ایتم های تک عضوی پر تکرار را پیدا کرده ام . و حال کدی نوشته ام که برای جفت آیتم ها خوب جواب می دهد ولی برای چند آیتم های پر تکرار متاسفانه جواب مطلوب را نگرفته ام .
- همچنین کدی را نوشتم که با استفاده از map_reduce این الگوریتم را پیاده سازی می کند .

1. در mapاول باید ایتم های پرتکرار را به عنوان کلید ومقدار ۱ را بعنوان مقدار در یک تاپل قرار دهد .
2. در reduce اول نیز باید تمام مقادیر با کلید های یکسان را جمع زد تا تعداد تکرار

SON

3. در map دوم باید جفت (ساپورت ، کاندیدا) را برای هر باکت محاسبه کرد .
4. در نهایت در reduce دوم مقادیر را بر حسب کاندیدا ها که کلید ها هستند جمع کرده و اگر مقدار بدست امده از مقدار ساپورت داده شده توسط کاربر بزرگترمساوی بود آنگاه در قسمت output آن را مشاهده می کنیم.


THANK YOU!