

1. **استيراد المكتبات اللازمة:** يتم استيراد المكتبات اللازمة للبايثون للتلاعب بالبيانات، والتصوير البياني، والتعلم الآلي.
2. **pd.read_csv("Market_Basket_Optimisation.csv")** باستخدام DataFrame **تحميل مجموعة البيانات:** يتم تحميل مجموعة البيانات في.
3. **لتخزين كل العناصر من كل عملية بيع.** يتم تكرار جميع الصفوف والأعمدة في البيانات، transaction يتم إنشاء قائمة تُسمى **numpy: جمع كل العناصر من كل عملية بيع في مصفوفة** numpy. ثم يتم تحويل القائمة إلى مصفوفة transaction. ويتم إضافة كل عنصر إلى قائمة items. ويتم تسمية العمود pandas باستخدام DataFrame إلى numpy يتم تحويل المصفوفة: **pandas باستخدام DataFrame تحويلها إلى**
5. **في Group By ويتم تعيين القيمة 1 لكل عنصر.** هذا يتيح لنا إجراء عملية DataFrame إلى incident_count **وضع 1 لكل عنصر لإنشاء جدول قابل للعد:** يتم إضافة عمود جديد يُسمى incident_count. ووقت لاحق لحساب عدد المرات التي يظهر فيها كل عنصر
6. **DataFrame من (NaN) من مجموعة البيانات:** يتم إزالة العناصر التي ليس لها قيمة (NaN) **حذف العناصر التي ليس لها قيمة**
7. **DataFrame لكل عنصر.** ثم يتم فرز incident_count حسب العناصر وحساب مجموع DataFrame يتم تجميع: **pandas جديد مناسب للتصورات باستخدام DataFrame إنشاء** incident_count حسب بترتيب تنازلي
8. **incident_count.** يتم إنشاء خريطة شجرية لإظهار أعلى 50 عنصرًا حسب: **plotly إنشاء خريطة شجرية باستخدام**
9. **لتخزين كل عملية بيع كقائمة منفصلة.** يتم تكرار جميع الصفوف في البيانات، transaction يتم إعادة تعيين القائمة: **numpy تحويل كل عملية بيع إلى قائمة منفصلة وجمعها في مصفوفة** numpy. ثم يتم تحويل القائمة إلى مصفوفة transaction. ويتم إضافة كل عملية بيع كقائمة منفصلة إلى القائمة
10. **يتم تحويل كل عملية بيع إلى مصفوفة من القيم الثنائية (0 و 1) حيث 1 يعني أن العنصر موجود.** TransactionEncoder. يتم ترميز العمليات باستخدام **TransactionEncoder تهيئة** TransactionEncoder. في العملية و 0 يعني أنه غير موجود
11. **لتشمل فقط هذه العناصر DataFrame ويتم تصفية incident_count تحديد أعلى 30 عنصر:** يتم تحديد أعلى 30 عنصرًا حسب
12. **يتم تعيين الدعم الأدنى إلى 0.05 ويتم استخدام أسماء الأعمدة كأسماء العناصر.** DataFrame على FP-Growth يتم تنفيذ خوارزمية: **FP-Growth تشغيل خوارزمية**
13. **والحد الأدنى إلى 1 "lift"** يتم تعيين المقياس إلى FP-Growth. **إنشاء قواعد الربط:** يتم إنشاء قواعد الربط باستخدام النتائج من خوارزمية
14. **فرز القيم حسب الثقة:** يتم فرز القواعد حسب الثقة بترتيب تنازلي
15. **الرسم البياني:** يتم إنشاء رسومات بيانية لتصور الدعم مقابل الثقة والدعم مقابل الرفع لقواعد الربط

يمكن استخدام هذا النوع من الأنظمة لتوصية العناصر للمستخدمين. FP-Growth بشكل عام، يقوم هذا الكود بتنفيذ تحليل سلة التسوق وإعداد نظام توصية بسيط يعتمد على قواعد الربط وخوارزمية الخاص بك. أيضاً، قد تحتاج إلى ضبط CSV بالمسار الفعلي لملف "Market Basket Optimisation.csv" بناءً على العناصر التي اشتروها المستخدمون الآخرون. يرجى التأكد من استبدال

بناءً على مجموعة البيانات الخاصة بك والمتطلبات الخاصة بك. تذكر، في تعدين قواعد الربط، لا يوجد مفهوم للدقة مثلما هو الحال في fpgrowth في استدعاء الدالة min_support معلمة التصنيف. الهدف هو كشف العلاقات المثيرة للاهتمام في البيانات، وليس إجراء توقعات دقيقة. يتم قياس جودة القواعد عادة بدعمها، وثقتها، ورفعها