خوارزميات التسويق من واحد إلى واحد

* فهرس محتويات المشروع:

1. **توصيف المشروع ------------------------------------------------------------------------------- 2**
2. **دراسة نظرية: -----------------------------------------------------------------------------------2**

**2-1- Collaborative Filtering -------------------------------------------------------------3**

1. **تحليل المشروع :**

**3-1- المتطلبات الوظيفية ------------------------------------------------------------------------ 4**

**3-2- المتطلبات اللاوظيفية ---------------------------------------------------------------------- 7**

1. **تصميم المشروع:**

**4-1- بنية المعطيات (Recommendation Tree):**

**4-1-1- شكل الشجرة ----------------------------------------------------------------------- 7**

**4-1-2- أنواع عقد الشجرة ------------------------------------------------------------------ 8**

**4-2- خوارزمية الـRecTree:**

**4-2-1- دخل الخوارزمية -------------------------------------------------------------------- 9**

**4-2-2- خطوات عمل الخوارزمية ------------------------------------------------------------ 9**

**4-2-3- الخوارزميات المستخدمة في الشجرة:**

**4-2-3-1- ConstructRecTree ------------------------------------------------- 10**

**4-2-3-2- KMeans+ ------------------------------------------------------------ 11**

**4-2-3-3- ComputeCorrelationSimilarity+ ------------------------------- 15**

**4-2-3-4- QueryRecTree ----------------------------------------------------- 16**

**4-2-3-5- ComputeDeviationFromMeanPrediction -------------------- 17**

**4-2-3-6- InsertUser -----------------------------------------------------------18**

**4-2-4- ملاحظات عامة حول الخوارزمية ---------------------------------------------------- 19**

**4-3- مخطط الصفوف المستخدمة --------------------------------------------------------------- 20**

**4-4- قاعدة بيانات الموقع:**

**4-4-1- مخطط الـERD-------------------------------------------------------------------- 21**

**4-4-2- شرح الكيانات وواصفاتها ---------------------------------------------------------- 22**

1. **مراحل تنفيذ واختبار النظام --------------------------------------------------------------------- 24**
2. **دليل استخدام النظام ---------------------------------------------------------------------------- 24**
3. **خاتمة ----------------------------------------------------------------------------------------- 26**
4. **آفاق مستقبلية --------------------------------------------------------------------------------- 26**
5. **المراجع المستخدمة ---------------------------------------------------------------------------- 27**

* توصيف المشروع:
* تقوم فكرة التسويق من واحد إلى واحد على الانتقال من سياسة كسب الزبائن إلى سياسة الحفاظ على الزبائن، ولتحقيق ذلك أصبح من الضروري أن يتم ترغيب الزبائن بمنتجات أكثر حتى تصبح عملية الحفاظ عليهم كزبائن دائمين للموقع أسهل، ولكن هذا الشيء يعني أن من مهام الموقع الذي يعتمد فكرة التسويق من واحد إلى واحد أن يقوم بتوقع المنتج الذي قد يحبه الزبون ويرغب بشرائه، وهذه العملية تتطلب إيجاد خوارزميات خاصة من أجل دراسة وتحليل شخصية الزبون واستنتاج ميوله ورغباته حتى يتم الحصول على المنتج الصحيح، وهذا أمر طبيعي فلا تكون عملية التوقع عشوائية، وإنما وفقاً لقوانين وخوارزميات معينة تقوم بهذه العملية.
* ومشروعنا هو بناء إحدى خوارزميات التسويق من النمط واحد لواحد، حيث تم استخدام موقع ويب يتم تسويق كافة أنواع الأفلام العالمية فيه بطريقة تفاعلية مرتبطة بذاتية الزبون المرتبطة بذاتها مع زبائن آخرين متقاربين إلى حد كبير بالميول والأهواء.
* وما اختيار الويب كبيئة لتوظيف هذه الخوارزمية إلا من أجل تسهيل وصول المستخدمين المختلفين من كافة أنحاء العالم إلى المنتجات ولتحقيق الفائدة المرجوة من الخوارزمية المستخدمة كون الشبكة العنكبوتية كونت رابطاً كبيراً بين جميع المستخدمين من كافة أنحاء العالم، كما وسهلت شبكة الانترنت من عملية التسويق الإلكتروني وذلك لأنها متاحة لجميع الناس في كافة أنحاء العالم مما سهل على الشركات المنتجة في عملية تسويق بضائعها إلكترونياً.
* يرتبط هذا الموقع بقاعدة بيانات ضخمة تحوي معلومات عن كافة المستخدمين أصحاب الحسابات على هذا الموقع وقاعدة بيانات ضخمة أخرى تحوي كافة المنتجات (الأفلام) التي يقدمها الموقع، بإجراء بعض العمليات الحسابية وباستخدام بعض قوانين الإحصاء الرياضي على كلا قاعدتي البيانات يتم إيجاد التقارب بين المستخدمين وحشرهم ضمن عناقيد (Clusters) بحيث عناصر العنقود الواحد متقاربين إلى مدى كبير بالأهواء والأفكار والرغبات (like-minded people) ومعرفة هذا التقارب تم من خلال إجراء تقييم لمنتجات الموقع من قبل المستخدمين.
* وبعد التقييم يتم استخدام قوانين مخصصة لحساب الترابط بين المستخدمين من أجل الحصول على هذه العناقيد وتقديم Recommendation لكل منهم بحسب العنقود الذي ينتمي إليه، وكل هذا يتم من خلال استخدام خوارزمية خاصة بهذه العملية تسمى RecTree والتي تعتمد على مبدأ الـCollaborative filtering.
* دراسة نظرية:
* تقسم هذه الخوارزميات المستخدمة من أجل الحصول على توقعات لمنتجات قد يرغب بها الزبون recommendations إلى العديد من الأنواع، ومن أهمها:

1. Collaborative Filtering: تعتمد هذه الطريقة على خوارزميات تسعى للتنبؤ بأذواق الأشخاص اعتماداً على تحليل الأفراد المتشابهين بالآراء والبيئة (Like-Minded People)، مما يعني أن اهتمام أحد الأفراد بمنتج جديد ينعكس على كافة الأفراد المتماثلين بالاهتمامات.
2. Content-Based Filtering: تختلف هذه الفلترة عن سابقتها بأن السابقة تعتمد على إيجاد علاقة بين الزبون مع غيره من الزبائن، في حين في هذه الطريقة يتم الاعتماد على الكلمات المفتاحية التي يستخدمها هذا الزبون والتي تعبر عن اهتماماته نظراً لأنه قد بحث عنها فيما سبق أو زار صفحتها الخاصة من قبل، مما يعني أنه مهتم بهذا النوع من الأفلام فنقوم في المرة التالية التي يزور بها الموقع بإعطائه منتجات من نوع ومحتوى مشابه لما قام بالبحث عنه سابقاً.
3. Rule-Based Filtering: وهنا تعتمد عملية التوقع على عمليات ربط لإيجاد مجموعات مرتبطة مع بعضها، فمثلاً قد نوجد مجموعة مرتبطة من المنتجات كما في المثال التالي:

إذا اشترى الزبون المنتج 1 والمنتج 2 فإنه سيشتري المنتج 3

وهذه العملية فادتنا بأننا في الزيارة التالية له للموقع نقوم بإعطائه المنتج الثالث كـrecommendation.

وقد تكون عملية الربط وفقاً لمجموعة بين زبائن وليس بالضرورة بين منتجات، فمثلاً:

إذا اشترى الزبون A المنتج 1 والزبون B كذلك فإن الزبون C سيقوم بذلك

وهذه أيضاً إحدى الطرق التي نستنتج بها المنتجات التي سنرغب الزبون بها.

* ويوجد الكثير من الطرق الأخرى التي يتم استنتاج ماهية المنتج المرغوب من قبل الزبون.

Collaborative Filtering:

* ذكرنا أن هذه الطريقة من الفلترة تعتمد على خوارزميات تتوقع المنتجات التي قد يرغب بها الزبون وفقاً لرغبات الزبائن المشابهين لهم بالاهتمامات like-minded-people، وتتميز عن بقية الطرق الأخرى للفلترة بأنها تقدم تنوعاً كبيراً للمنتجات التي يتم ترغيب الزبون فيها، كما وأثبتت التجارب عليها بأنها تعطي دقة أكبر في عملية التوقع، وتقسم إلى منحيين أو طريقين مختلفين:

1. منحى الترابط Correlation (memory-based): وفي هذا المنحى يتم حساب ترابط الزبون مع الزبائن الذين يشبهونه باهتماماتهم فقط دون النظر إلى معلومات ذاتياتهم الأخرى، حيث يقوم كل زبون منهم بعملية rating لأي منتج وهذه القيمة هي التي تعكس مدى اهتمامه بأحد المنتجات أو أحد الأنواع، ولكن له بعض المساوئ التي يعاني منها وهي:
2. قلة المعلومات والمستندات Data Sparsity: وذلك لأن عدد الزبائن الذين يقومون بتقويم المنتجات قليل، وكذلك المنتجات التي يقومون بتقويمها عادةً ما يكون قليلاً، وهذه المنتجات أيضاً عادةً ما تكون متفرقة أي كل منتج منها قد يكون من نوع.
3. عدم المقدرة على التوسع Disability Of Scalability: أي أنها لا تعطي نتائج فعالة في حال كثرة المستخدمين.

وهذا لا يعني أنها غير جيدة، فهي تتمتع بشيء هام وهو الدقة في التوقع فغالباً ما تكون نتائجها أقرب للصحة.

1. منحى التصنيف Classification (model-based): وتقوم الفكرة هنا على تصنيف الزبائن إلى مجموعات وذلك بحسب اهتماماتهم وتشابهها، فمثلاً عند وجود زبونين يحبان أفلام الرعب وجاء زبون آخر يطلب أحد أفلام الرعب ففي المرة القادمة التي سيزور بها الموقع يظهر له أفلام رعب كـrecommendations من اهتمامات الأشخاص الذين تم ضمه إلى مجموعتهم، أي أننا نقوم بتصنيف الزبائن وفقاً للـGenre أو نوع المنتج المشترك الذي يحبونه دونما حساب للترابط بينهم كما في المنحى السابق، ومن المشاكل التي يعاني منها هذا المنحى هي تضييع الوقت في عملية البناء والتحديث للنظام.

* تحليل المشروع:
* لبناء النظام يلزمنا معرفة بعض المتطلبات التي يتم تحقيقها من قبل الموقع، بعض هذه المتطلبات وظيفية ترتبط ارتباطاً مباشراً بمستخدمي الموقع وزواره ويجب على النظام تأمينها، ومتطلبات أخرى غير وظيفية تصب في مصلحة المستخدمين ولكن لا يشعرون وإنما يتم إجرائها بهدف تحسين الأداء والحفاظ على الخصوصية.

1. المتطلبات الوظيفية:
2. **خدمات الموقع:** بالاعتماد على التجارة الإلكترونية وخوارزميات التسويق واحد لواحد يقوم الموقع بتقديم عدد من النصائح (Recommendations) حول بعض المنتجات (الأفلام) لكل user بالاعتماد على ذاتيته وترابطه مع غيره من المستخدمين ضمن العنقود الواحد، والـRecommendations بمفهومها هي تقديم بعض المنتجات للمستخدم التي لم يسبق أن قام بمشاهدتها أو تقييمها أبداً.

كيف يتم حساب هذه الـRecommendation؟

بعد حساب ترابط المستخدمين وحشرهم ضمن عناقيد حيث أن العنقود الواحد يحوي المستخدمين الأقرب لبضعهم، بفرض أن هذا العنقود يحوي عشرة مستخدمين، تسعة منهم قاموا بتقييم فيلم matrix والمستخدم العاشر لم يقم بتقييمه حينها وكخدمة تسويقية يتم تقديم هذا الفيلم لهذا المستخدم وباحتمال كبير يمكن أن يقوم بشرائه، ذلك لأن اختياره تم وفقاً لما يرغبه باقي المستخدمين المشتركين معه بنفس العنقود والأكثر منه تقارباً، بمعنى آخر إذا تقارب الزبائن A,B,C والزبونان A,B أحبا المنتج x حينها ببساطة الزبون C سوف يحب المنتج x.

ويتم التعبير عن هذه الخدمة التسويقية بتقديمها عن طريق واجهات الموقع ذاته حيث عند كل دخول لمستخدم ما يتم تقديم النصائح له (المنتجات التي قد يرغب بها)، وبذلك نكون فعلاً قد حققنا مبدأ أن المنتج يسعى وراء الزبون وليس العكس.

1. **الدقة:** صحيح أن دراسة التقارب تمت باستخدام قوانين إحصائية احتمالية الوقوع إلا أن التجارب أكدت أن نتائجها إلى حد كبير صحيحة وقريبة جداً من ميول الزبون، وخصوصاً عند وجود قاعدة بيانات مستخدمين ضخمة جداً ويتم منها إنشاء عناقيد بشكل أكبر، وبكثرة العناقيد يصبح التقارب بين الزبائن بشكل أفضل في كل عنقود وذلك لوفرة المستخدمين بشكل أكبر.

ومن ذلك نكون قد حققنا الدقة بانتقاء الـRecommendation المناسب لكل زبون.

1. **طرق تحديد اهتمامات الزبون:**

لمعرفة ميول زبون ما يتم إيجاد بعض الطرق التي تحقق لنا ذلك، طبعاً الهدف من ذلك تقديم الـRecommendations لهذا المستخدم، بعض هذه الطرق هي:

1. Rating: هي عملية اختيارية يجريها الزبون لكل منتج يمكن أن يصادفه في الموقع ويعبر عنها بالقيم التي تتراوح بين الـ1 والـ5 والتي تمثل مقدار رغبة الزبون بالمنتج الموافق، وهذه الدرجات هي:

5 تعني I love it

4 تعني I like it

3 تعني It 's OK

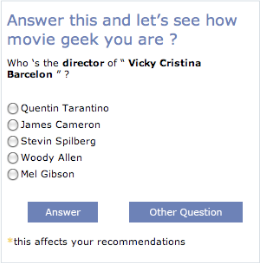
2 تعني I don't like it

1 تعني I hate it

تعتبر عملية الـrating هي العملية الأولى والأساسية التي تفيدنا في حساباتنا ضمن خوارزمية التسويق المتبعة (RecTree) ذلك لأنها الطريقة الأكثر استخداماً من أجل ترجمة مدى إعجاب الزبون بالمنتج من خلال الأرقام.



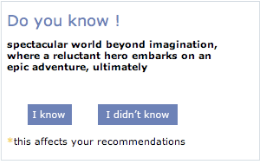
1. المسابقة: وهذه الطريقة هي عبارة عن بعض الأسئلة التي تطرح على الزبون بصيغة الاختيار من متعدد، كأن يُسأل عن شخصية مخرج فيلم معين، أو شخصية بطل الفيلم، أو غيرها من الأسئلة المستنبطة من قاعدة الأفلام التي يحويها الموقع، وتفيد هذه المسابقة في أخذ انطباع عن هذا الزبون وبالتالي ترجمة هذا الانطباع إلى أرقام كتقييم للأفلام التي يحبها بسبب إجابته على سؤالها في المسابقة بشكل صحيح، مما يعني أنه فعلياً يتم الاستفادة من إجابات هذا الزبون على الأسئلة التي تطرح عليه وتترجم إلى تقييم يفيد في الخوارزمية.



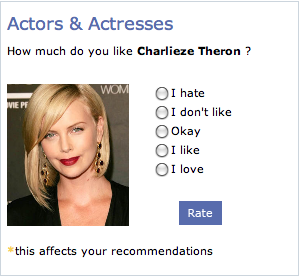
1. Session: ويقصد بها تتبع مسار المستخدم ضمن الموقع لأخذ انطباع أولي عن ميوله والأفلام التي يفضلها، وكمثال لو أن المستخدم x دخل صفحة الأفلام ذات النمط Action وبقي يتصفح العديد منها دون أن يقيم أيٍّ منها حينها يمكن لنا أن نأخذ انطباع بأن هذا المستخدم يفضل هذا النوع من الأفلام، ذلك أنه لو لم يكن يفضلها لم يبحث عنها.

إن هذا المسار يتم ترجمته إلى قيم عددية تقابل عملية تقييم لمجموعة من المنتجات والتي بدورها تفيدنا في معرفة العنقود الذي سيضاف إليه هذا المستخدم ذلك لأن هذا الرقم هو انطباع شبه صحيح عن ميول المستخدم.

1. Do you know: وهذه الطريقة هي عبارة عن سؤال سريع على شكل yes/no question يتم فيه الاستفسار من الزبون إن كان يعرف معلومة معينة أم لا في زاوية من زوايا صفحة الموقع الرئيسية، وهذه المعلومة تخص فيلماً معيناً أو مخرجاً معيناً أو أي معلومة تتعلق بأحد الأفلام التي يحويها الموقع، وإن كان جواب الزبون بالإيجاب أي أنه يعرف هذه المعلومة المتعلقة بأحد الأفلام فهذا ينعكس على الانطباع العام عن هذا الزبون فمعرفته بهذه المعلومة دليل على اهتمامه بهذا النوع من الأفلام، مما يدعو إلى ترجمة هذا الانطباع العام عنه على تقييم لهذا النوع من الأفلام ككل، فينعكس في المرات التالية التي يدخل فيها إلى الموقع على نوعية الأفلام التي ستظهر له على أنها Recommendations، وهذه الطريقة تعطينا انطباع عن سلوك هذا الزبون بهذا الأسلوب.



1. Actor: وفي هذه الطريقة يقوم الزبون بتقييم أحد الممثلين أو الممثلات كتعبير عن مدى إعجابه أو محبته بأفلام هذا الممثل أو الممثلة، وذلك من أجل تطبيق القاعدة التي تقول أن الشخص الذي يحب ممثلاً معيناً فإنه يحب عادةً الأفلام التي يتواجد فيها، فهذا من الطبيعي أن يكون من متتبعيه ليشاهد أفلامه، وبالتالي يتم ترجمة هذا التقييم إلى أرقام تمثل مدى إعجاب الزبون بهذا الممثل أو الممثلة فينعكس أيضاً على الأفلام التي ستظهر له كـRecommendations في صفحة الزبون الخاصة.



* نلاحظ من الطرق المعروضة سابقاً أنه من أجل أخذ انطباع عام عن الزبون تم توظيف كل جزء من الموقع تقريباً، حتى أن الصفحة الرئيسية للزبون قد وظف كل جزء فيها من أجل إلتقاط سلوك الزبون لينعكس فيما بعد على الـRecommendations التي تختار له.

1. المتطلبات اللاوظيفية:

أو ما تسمى الخدمات التي تقدمها إدارة الموقع لكافة الزبائن دون أن يتم التخاطب مع المستخدمين فيها، وهذه الخدمات هي:

1. **الأمن Security:** أهم سمة من سمات المواقع على شبكة الويب أن تكون مدعمة لحمايتها من الاختراق من قبل hackers على الموقع، وذلك من أجل حماية البيانات التي تحويها قاعدة بيانات الموقع من الانتشار والسرقة.
2. **أداء النظام Performance:** إن الخوارزمية المتبعة لتقديم الـRecommendations للزبائن تم الإثبات رياضياً أنها تحقق الدقة المطلوبة بتعقيد أقل من غيرها من الخوارزميات التسويقية الاخرى، وبالتالي فإن سرعة النظام هي جيدة جداً بالرغم من ضخامة البيانات التي تعالجها.
3. **الخصوصية Privacy:** يضمن الموقع أن تكون البيانات المتعلقة بزبون ما خاصة به لا يمكن الاطلاع عليها من قبل الزبائن الآخرين، وهذه الخصوصية يمكن أن تزيد ثقة كل زبون بالموقع.
4. **قابلية التوسع:** يضمن الموقع سهولة توسع مجال المنتجات المراد تسويقها ذلك أن الخوازرمية التي يستخدمها هي خوارزمية عامة يمكن أن تستخدم في جميع أنواع المنتجات دون استثناء.

* تصميم المشروع:
* بنية المعطيات المستخدمة (Recommendation tree):
* شكل الشجرة:
* هذه الشجرة هي شجرة ثنائية، كل عقدة منها تشكل عنقوداً Cluster من الزبائن المتشابهة فيما بينها، وكل عقدة أب لها عقدتان أبناء، والعقدتان الأبناء هما عنقودان جزئيان من العنقود الأب، كما في الشكل التالي:



* سنقوم بشرح كيفية التقسيم وسبب توقف نمو الشجرة في بعض الفروع وما هي العوامل التي توقف عملية النمو فيما بعد.
* أنواع عقد الشجرة:
* تقسم عقد الشجرة في هذه الخوارزمية إلى ثلاث أنواع مبينة بالشكل التالي:



1. العقد الداخلية internal nodes: وفي هذه العقد يتم تخزين الـdatabase أو أجزاء منها وذلك من أجل القيام بتقسيمها في إلى أجزاء أصغر في العقد الأبناء.
2. الأوراق leaves: وهذه العقد تحوي أيضاً جزءاً من الـdatabase ولكن فقط للزبائن المتشابهين في الاهتمامات بشكل كبير، ويتم في هذه العقدة فقط حساب مصفوفة الترابط similarity matrix بين الزبائن والتي تحمل الشكل التالي:

|  |  |  |  |
| --- | --- | --- | --- |
| u3 | u2 | u1 |  |
| w13 | w12 | w11 | u1 |
| w23 | w22 | w21 | u2 |
| w33 | w32 | w31 | u3 |

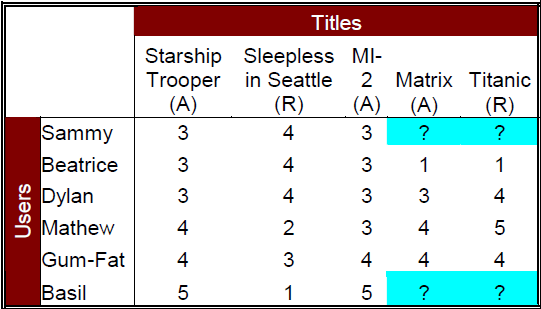
حيث هذه المصفوفة من أجل ثلاث زبائن فقط، ويمكن تعميمها على n زبون، والعامل w12 هو ترابط الزبون 1 مع الزبون 2 ويسمى similarity coefficient، ومقداره دوماً بين -1 و1، وهذه المصفوفة متناظرة طبعاً كون الترابط بين الزبون 1 مع الزبون 2 هو نفسه بين الزبون 2 مع الزبون 1، والترابط بين الزبون ونفسه دوماً 1، لذلك قطرها الرئيسي لهذه المصفوفة كله واحدات.

1. العقد الخارجية outlier nodes: وهذه العقد هي حالة خاصة من الاوراق، وتختلف عنها بأن عدد الزبائن في هذه العقد قليل مقارنةً بالحد الأدنى المسموح لعدد الزبائن في الورقة، ويتم معاملتها معاملة خاصة في عملية حساب التوقع من أجل الزبائن الذين ينتمون إليها.

* خوارزمية الـRecTree:
* دخل الخوارزمية:
* مما يجدر ذكره أن الشجرة يتم بناؤها مرة واحدة فقط، ويمكن أن تتم عملية إعادة بنائها مرة أخرى في الحالات النادرة، ذلك لأن عملية بناءها ليست بالأمر الجيد كونها تستهلك وقتاً من المعالج في عملية البناء خصوصاً إن كانت قاعدة البيانات كبيرة، وبالتالي فإن الدخل يقسم إلى قسمين:

1. الدخل عند البناء: وفي هذه المرحلة يتم جلب كل قاعدة البيانات التي تحوي جميع الزبائن وجميع المنتجات وتقويم هؤلاء الزبائن لهذه المنتجات (طبعاً لا نجلب جميع الذاتيات التي تخص الزبائن لأن الخوارزمية لا تهتم سوى بتقييمات هؤلاء الزبائن للمنتجات).

أي أن دخل الخوازرمية هو جدول (table) مشابه لهذا الجدول:



حيث أن الأماكن الفارغة والملونة بالأزرق هي منتجات لم يقم الزبون المقابل لها بالسطر لتقييمها، ومن مهمة الخوارزمية أن تقوم بإعطائه إياها كـrecommendation إن كان الزبائن الآخرون في عنقوده يرغبون بها، طبعاً الجدول السابق هو مثال صغير جداً كون قاعدة بيانات الزبائن والمنتجات في الموقع تكون أكبر بكثير ولكن المبدأ واحد.

1. الدخل أثناء العمل: وفي هذه المرحلة قد نقوم بإضافة زبون جديد إلى الشجرة قد سجل من جديد على الموقع، أو قد نقوم بإضافة منتج جديد إلى قاعدة بيانات منتجات الموقع، كون الخوارزمية تضمن هذه الديناميكية في الإضافة والحذف والتعديل دون عملية إعادة بناء لها.

* خطوات الخوازرمية:

1. يتم تقسيم الزبائن إلى عناقيد أو مجموعات أصغر من الزبائن (cliques) المتشابهين في الاهتمامات.
2. تتم عملية التقسيم بشكل عودي في كل عقدة ابن في هذه الشجرة وهكذا حتى نصل إلى شروط التوقف لعملية التقسيم والعنقدة.
3. عند الوصول إلى عقد الأوراق يتم حساب مصفوفة الترابط بين الزبائن.
4. يتم استخدام عوامل الترابط similarity coefficients بين الزبائن من أجل حساب التوقع من أجل كل منتج من المنتجات التي لم يقم بتقييمها الزبون، وهذا الكلام يتم من أجل جميع الزبائن في هذا العنقود طبعاً.

* الخوارزميات المستخدمة في الشجرة:
* ConstructRecTree(parent ,y ,β ,g ,OutlierSize):

في هذه الخوارزمية يتم توليد الشجرة كاملة، ويتم استدعاؤها كما ذكرنا مرةً واحدة، إلا في الحالات الخاصة التي نحتاج فيها إلى إعادة بناء الشجرة كلها.

* الدخل:

1. Parent وهو عقدة الأب، وهي العقدة الحالية التي سنقوم بتقسيمها أو ربما حساب الترابط فيها.
2. Y وهي قاعدة البيانات التي تحوي الزبائن وتقييماتهم لكل المنتجات والتي مثلناها بالجدول في الصفحة السابقة.
3. β وهو العدد الأعظمي للزبائن في كل عقدة، ويمثل الحد الذي نوقف عنده تقسيم العقدة الحالية (وهو الحد الأعظمي للزبائن في الورقة).
4. g ويمثل الـMax Depth أي العمق الأعظمي المسموح لأحد فروع الشجرة أن يصل له.
5. OutlierSize ويمثل الحد الأدنى للزبائن في الورقة، والعقد التي تحوي زبائن أقل منه تصنف على أنها عقدة خارجية outlier node.

* الخرج: شجرة الـRecTree.
* الخطوات:

1. إذا كان الـparent=NULL عندها نضع القيمة 0 بعداد الـglobal iterations المستخدم من أجل حساب العمق الحالي للشجرة.
2. ننشئ عقدة V ونقوم بربطها مع العقدة الأب بوصلة ثنائية أي يمكن الصعود أو الهبوط باستخدام هذه الوصلة الثنائية الأطراف.
3. نضع جميع الزبائن في Y مع تقييماتهم للمنتجات في هذه العقدة الحالية.
4. إذا كان عدد الزبائن في Y أقل من الحد OutlierSize نميز هذه العقدة على أنها outlier، ونوقف النمو في هذا الفرع لأننا وصلنا إلى outlier.
5. إذا أصبح عدد الزبائن في Y أقل أو يساوي الحد الأدنى المسموح لنوقف العنقدة في هذه العقدة والذي يساوي أيضاً الحد الأعظمي لعدد الزبائن في الورقة leaf المسمى β، أو إذا تجاوزنا العمق الأعظمي للشجرة g عندها نوقف النمو في هذا الفرع ونقوم بتمييز هذه العقدة على أنها leaf، ثم نقوم بحساب مصفوفة الترابط similarity matrix باستدعاء التابع ComputeCorrelationSimilarity.
6. نحسب مركز العقدة أو ما يسمى Centroid العقدة ويمثل متوسط تقييمات زبائن هذه العقدة لجميع المنتجات ويمثل على شكل شعاع vector يحوي عناصر بعدد المنتجات وكل خانة منه تمثل متوسط تقييمات الزبائن في هذه العقدة للمنتج المقابل.
7. نزيد عداد الـglobal iterations بمقدار واحد.
8. نستدعي خوارزمية العنقدة KMeans+ لتقسيم الـdataset إلى قسمين (عنقودين) ونسمي المجموعة الأولى y1 والثانية y2.
9. من أجل الابن اليميني للعقدة الحالية نستدعي هذه الخوارزمية عودياً مع إضافة الـdataset الأولى y1 إليه، وكذلك من أجل الابن اليساري:

ConstructRecTree (right son of parent ,y1 ,β ,g ,OutlierSize)

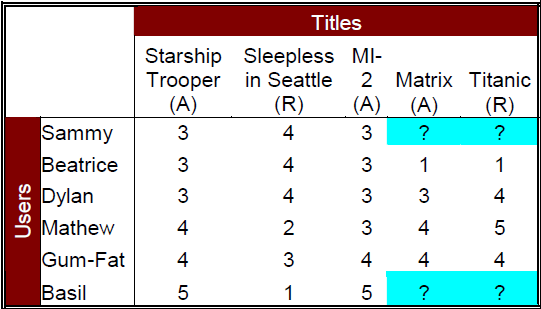
ConstructRecTree (left son of parent ,y1 ,β ,g ,OutlierSize)

* يوجد بعض العمليات التي قمنا بها في الخوارزمية السابقة ولم نقم بذكر سبب القيام بذلك وسيتم تبرير ذلك في الخوارزميات القادمة والتي تعتبر مساعدة من أجل إكمال استدعاء هذه الخوارزمية الأساسية لبناء الشجرة.
* KMeans+(Y):

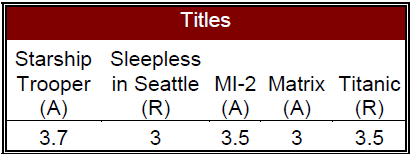
هذه الخوارزمية هي إحدى خوارزميات العنقدة المشهورة، وتستخدم من أجل تقسيم مجموعة من البيانات إلى عدد من الأقسام يساوي إلى العدد k، وهو أحد قيم الدخل لهذه الخوارزمية، ولكن بما أننا لا نقوم بتقسيم الـdataset التي لدينا إلا إلى مجموعتين لن نقوم بإدخاله وسنفترض دوماً بأن k=2.

* الدخل: Y والتي تمثل مصفوفة من تقييمات الزبائن للمنتجات، حيث كل سطر منها يمثل زبوناً وكل عمود هو منتج والتقاطع هو تقييم الزبون للمنتج المقابل، وهي الـdataset التي سنقوم بتقسيمها في هذه الخوارزمية.
* الخرج: مجموعتان منفصلتان من الزبائن كل مجموعة منها تسمى عنقود، ويحوي مجموعة من الزبائن المتشابهين بالاهتمامات، وسنقوم بالتعبير عن انتماء الزبائن إلى العناقيد الناتجة باستخدام مصفوفة ثنائية البعد عدد أسطرها هو عدد العناقيد (وهنا هو 2) وعدد أعمدتها هو بعدد الزبائن الذين كانو في المصفوفة الأصلية Y التي تعتبر دخلاً للخوارزمية، والتقاطع بين كل سطر وعمود يحوي Boolean ليدل إن كان المستخدم ينتمي إلى هذا العنقود المقابل في السطر أو لا.
* الخطوات:

1. نحسب r0 والذي يسمى the default vector ويعبر عن متوسط تقييمات كل الزبائن لكل المنتجات, وعدد أعمدته يساوي عدد المنتجات في Y، ويحوي متوسط تقييمات الزبائن لكل منتج من المنتجات، أي من أجل المثال التالي:



يكون r0 المقابل هو:



والفائدة منه هو أن نقوم بترميم الـratings المفقودة لدى بعض الزبائن، وذلك لأن خوارزمية العنقدة KMeans تتطلب أن تكون جميع الخانات ممتلئة في الـdataset، وهذه العملية في الترميم لا تضر بنتائج الخوارزمية لأنها أصلاً ترمم بقيم تمثل متوسط تقييمات الزبائن لأحد المنتجات وهؤلاء الزبائن هم أصلاً مترابطون في عنقود واحد وإلا لما قدموا دفعة واحدة، ومهمة الخوارزمية هنا الآن هو تقسيم هذه المجموعة من الزبائن المترابطين إلى مجموعتين أصغر هما أيضاً مترابطان بشكل أكبر.

1. نختار الآن مركزين للعناقيد، حيث المركز هنا هو أحد الزبائن من الـdataset يتم اختياره من أجل دراسة تقارب الزبائن الآخرين إليه، ومن ثم في المرحلة التالية للخوارزمية يتغير هذا المركز حتى يتم تقسيم الزبائن إلى مجموعات متقاربة أكثر مما سبق وهكذا. والمركز هنا كمفهوم هو vector من التقييمات ويتم اختيار المركز الأول والثاني بالطريقة التالية:
2. نحسب المسافة الإقليدية Eqlidis distance بين كل زبون من الزبائن والصفر (origin) وهو vector فيه كل التقييمات تساوي الصفر وعدد أعمدته يساوي عدد المنتجات في Y، والقانون المستخدم في ذلك هو:

حيث rik هو تقييم الزبون i للمنتج k وrjk هو تقييم الزبون j للمنتج k.

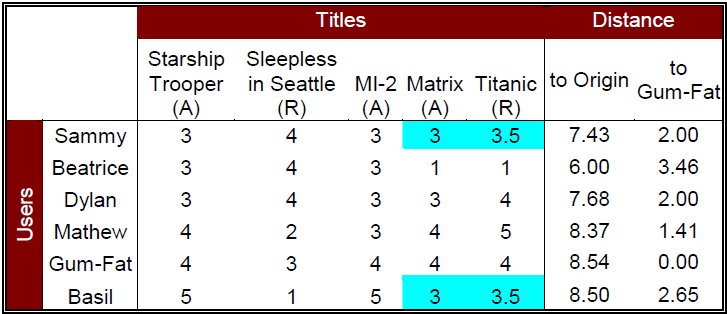
وn هو عدد المنتجات الكلي.

والزبون الأبعد عن الـorigin (صاحب المسافة الأكبر) هو يتم اختياره على أنه المركز الأول c1.

1. يتم حساب المسافة الإقليدية أيضاً ولكن هذه المرة بين جميع الزبائن والمركز الأول الذي اخترناه لأنه الأبعد عن الصفر، والزبون الأبعد عن المركز الأولى نعتبره أنه المركز الثاني c2.

مثال:

بحساب المسافة الإقليدية بين الزبائن والصفر ومن ثم بين الزبائن والمركز الأول في المثال السابق لـY نفسه يكون الناتج هو:



C1

C2

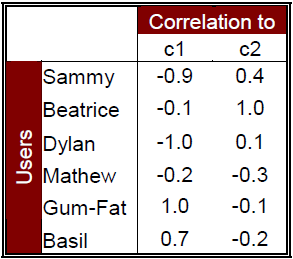
حيث حسبنا المسافة الإقليدية بين Sammy والصفر origin بالطريقة التالية:

*وحسبناها بين* Beatrice *و*Gum-Fat*:*

*وهكذا من أجل جميع الزبائن.*

1. *نضع عداد الـ*iterations=0.
2. بعد أن قمنا باختيار المركزين الابتدائيين c1 وc2 أصبح بالإمكان أن نقوم بتقسيم الزبائن في Y إلى مجموعتين مترابطتين، الأولى مترابطة مع المركز c1 والأخرى مع المركز c2، ولكن مما يجدر ذكره أن العملية السابقة لاختيار c1 وc2 تتم فقط مرة واحدة في خوارزمية KMeans أي هي عملية الـinitialize لكل من المركزين c1 وc2، ومن هذه الخطوة يتم التكرار.

في هذه الخطوة نقوم بحساب الترابط بين الزبائن وكل من المركزين c1 وc2، والترابط بين كل زبون وآخر يتم حسابه باستخدام قانون خاص لحساب الترابط بين زبونين يتم استخدام الخوارزمية ComputeCorrelationSimilarity من أجل ذلك باستدعائها، ويتم تخزين هذه القيم في مصفوفة مرحلية تسمى مصفوفة المسافات distances عدد أعمدتها هو اثنان كل عمود منها يمثل مركزاً من المراكز والأسطر بعدد الزبائن والتقاطع يخزن فيه الترابط بين الزبون والمركز المقابل، أي لها الشكل التالي:



وبعدها نستطيع تحديد إلى أي من العنقودين ينتمي كل زبون من الزبائن، وذلك بإضافته إلى العنقود الذي يرتبط مع مركزه أكثر، فمثلاً الزبون Sammy قيمة ارتباطه مع المركز c2 (وهي 0.4) أكبر من ارتباطه مع المركز c1 (وهي -0.9) لذلك هو ينتمي إلى العنقود الثاني وليس إلى الأول.

وبالتالي يمكن تكوين مصفوفة مؤقتة لخرج الخوارزمية ككل وهي:

Sammy Beatrice Dylan Mathew Gum-Fat Basil

C1

C2

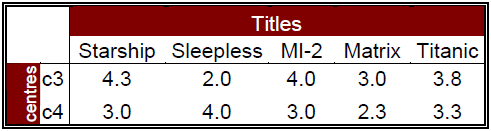
وذلك يعني أن Sammy ينتمي إلى العنقود الأول الذي يقوده المركز c1 وهكذا.

لنسمي هذه المصفوفة بالإسم curr\_result وتحسب في كل iteration.

1. بعد الخطوة السابقة أصبح الزبائن مقسومين إلى عنقودين لهما الشكل التالي:



وفي هذه الخطوة سنقوم بإيجاد المراكز الجديدة c3, c4 والتي سيتم العنقدة من جديد وتوزيع الزبائن وفقاً للترابط معها، يتم حساب c3 الآن من العنقود الأول C1 بإيجاد متوسط تقييمات كل الزبائن في العنقود C1 لكل المنتجات، وكذلك نوجد c4 من العنقود C2، ولهما الشكل التالي:



وهذان المركزان الجديدان هما اللذان سيتم توزيع الزبائن إلى مجموعتين وفقهما في التكرار التالي للخوارزمية next iteration، حيث نحسب الترابط بين الزبائن وهذين المركزين الجديدين.

1. Iterations++.
2. إذا كان iterations<Max\_iterations نعود إلى الخطوة الثالثة لتنفيذ الـiteration الجديدة، وذلك إذا كانت المصفوفة curr\_result تختلف عن المصفوفة السابقة prev\_result، أما إذا كانت هذه المصفوفة تحمل نفس القيم لسابقتها في التكرار السابق ولم يتغير تقسيم الزبائن على العناقيد نوقف الخوارزمية، وبالتالي شرط توقف الخوارزمية هو واحد من الحالتين:
3. Iterations>Max\_iterations.
4. وصل العناقيد إلى حالة استقرار أي لم ينتقل ولا أي زبون من الزبائن من عنقود إلى آخر.
5. نرد مصفوفة الخرج الآن curr\_result من الخوارزمية.

* ونلاحظ أن خوارزمية Kmeans+ هي خوارزمية تكرارية ويتم استدعاؤها بشكل عودي في خوارزمية بناء الشجرة ConstructRecTree على كل dataset في كل عقدة داخلية internal node كونها هي العقد الوحيدة التي تحتاج إلى تقسيم، وتم تسميتها بإسم Kmeans+ بدلاً من Kmeans ذلك لأنها معدلة عنها لتعطي نتائج أفضل للخوارزمية RecTree، ففي خوارزمية Kmeans الأصلية يتم استخدام قانون المسافة الإقليدية في كل مرة يتم فيها توزيع الزبائن على العناقيد بدلاً من قانون الترابط Correlation Similarity بينما في خوارزميتنا هنا اعتمدنا قانون الترابط بين الزبونين ذلك أنه يعطي نتيجة أدق لترابط شخصين.
* ComputeCorrelationSimilarity+(Y):

هذه الخوارزمية تقوم بحساب الترابط بين كل زبون والآخر، وذلك وفقاً لقانون الترابط للعالم Pearson.

* الدخل: Y والتي تمثل الـdataset لتقييمات الزبائن للمنتجات.
* الخرج: مصفوفة مربعة تحوي معاملات الترابط بين الزبائن لها الشكل التالي وتسمى Correlation Matrix:

|  |  |  |  |
| --- | --- | --- | --- |
| u3 | u2 | u1 |  |
| w13 | w12 | w11 | u1 |
| w23 | w22 | w21 | u2 |
| w33 | w32 | w31 | u3 |

* الخطوات:

1. نحسب r0 والذي يمثل الـdefault vector وسبق ذكره فيما سبق في خوارزمية KMeans، ولكن السبب وراء حسابه هنا ليس نفس السبب لحسابه هناك.
2. نرمم مصفوفة الـY ونضع القيم المقابلة من الـdefault vector لكل منتج لم يقم الزبائن بتقييمه.
3. نحسب متوسط تقييمات كل زبون من الزبائن ونحتفظ به لنستخدمه في قانون الترابط، ويتم حساب متوسط تقييمات الزبون من القانون:

حيث يمثل ru,k تقييم الزبون u للمنتج k، وYu هي قائمة المنتجات التي قام الزبون بتقييمها و هو عددها.

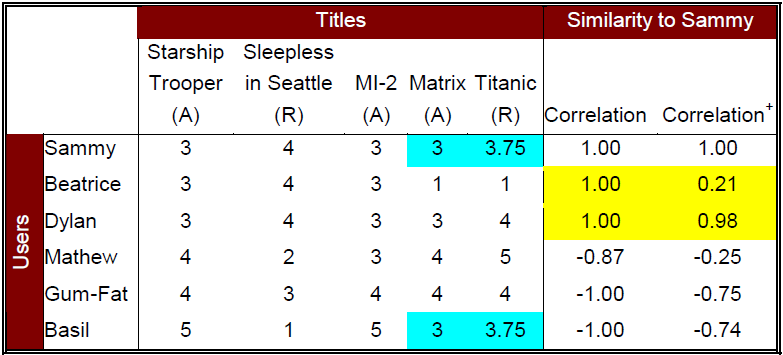
1. نحسب الانحراف المعياري لتقييمات كل زبون من الزبائن، وهو أيضاً مستخدم في قانون إيجاد الترابط بين كل زبون والآخر، نوجد التباين أولاً من القانون:

ثم نوجد الجذر فنحصل على الانحراف المعياري لكل زبون.

1. نحسب الترابط بين كل زبون والآخر باستخدام القانون:

حيث ra,i يمثل تقييم الزبون a للمنتج i، وi ينتمي إلى Ya,u وهي مجموعة المنتجات المشتركة بين الزبونين u وa و هو عددها.

* أما عن سبب تسمية هذه الخوارزمية بالإسم ComputeCorrelationSimilarity+ بدلاً من ComputeCorrelationSimilarity هو أننا هنا قمنا بترميم مصفوفة Y وملأنا التقييمات الفارغة باستخدام الـdefault vector، وقمنا بهذا الشيء ذلك لأن القيم المفقودة والناقصة من التقييمات تمنع من الحصول على قيم دقيقة في معظم الأحيان، وبالحساب بدون الترميم نحصل على القيم المبينة في العمود Correlation ومع الترميم نحصل على القيم المبينة في العمود Correlation+ من الجدول التالي:



نلاحظ أنه بدون ترميم يظهر لنا أن كل من Beatrice وSammy وDylan هم متطابقون تماماً مع العلم أن تقييماتهم مختلفة! لذلك جاءت فكرة الترميم لتحل هذه المشكلة.

* QueryRecTree(S):

هذه الخوارزمية هي وسيلة تخاطب الموقع مع الخوارزمية، فالشجرة بعد أن تبنى ويتم توزيع الزبائن على عقدها يصبح من الضروري أن يتم حساب التوقع وتحديد المنتجات التي سنطرحها كـrecommendation لكل زبون من الزبائن، وهذه الخوارزمية هي التي تقوم بحساب هذا التوقع وتحدد ما نسبة أن يرغب كل زبون بالمنتجات التي لم يشاهدها من قبل وفقاً لموقعه في الشجرة، ذلك لأن موقع الزبون في الشجرة يحدد طريقة حساب التوقع الخاص به وكذلك يحدد ماهية المنتجات التي ستطرح له كـrecommendation كون هذه المنتجات ترتبط بالزبائن الذين يشاركونه العقدة (like-minded-people).

* الدخل: S وتمثل dataset من الزبائن الذين لم يقومو بتقييم بعض المنتجات، والهدف تعبئة توقعات الـRecTree في مصفوفة الـrecommendations الخاصة بكل زبون.
* الخرج: هو database من الزبائن وكل زبون منهم قد ملئت مصفوفة الـrecommendations الخاصة به.
* الخطوات:

من أجل كل زبون من الزبائن في S

من أجل كل منتج في سطر الزبون لم يقم بتقييمه

1. اذهب إلى العقدة التي يتوضع فيها هذا الزبون في الشجرة من خلال المؤشر الذي يوجد في بنية معطيات الزبون والذي يحوي عنوان العقدة التي يوجد فيها.
2. إذا كانت العقدة:
3. داخلية internal: عندها يكون الـrecommendation للـitem الذي وصلنا له في الحلقة هو متوسط تقييمات باقي الزبائن لهذا المنتج.
4. ورقة leaf: يتم استدعاء إجرائية حساب التوقع من الانحراف المعياري المثقل ComputeDeviationFromMeanPrediction.
5. عقدة خارجية Outlier Node: نصعد إلى عقدة الأب لهذه العقدة (كون الوصلة من الطرفين بين العقد) ونحسب الـrecommendation بمتوسط تقييمات باقي الزبائن لهذا المنتج (كون عقدة الأب لهذه العقدة هي داخلية).

* إن السبب وراء وجود زبائن في عقدة داخلية لم يصلو إلى عقدة خارجية أو ورقة أي لم يصلو إلى آخر الشجرة هو وجود حالة خاصة عند إضافة زبون إلى الشجرة، سيتم شرحها في خوارزمية الإضافة لاحقاً.
* ComputeDeviationFromMeanPrediction(Y):

هذه الخوارزمية يتم فيها حساب التوقع الرياضي لتحديد نسبة أن يحب الزبون أحد المنتجات، وهذه الخوارزمية لا يتم استدعاؤها إلا على زبائن موجودون في عقدة ورقة.

* الدخل: عبارة عن زبون ومنتج مقابل له، حيث أن هذا الزبون لم يسبق وأن قيم هذا المنتج ومهمة هذه الخوارزمية تحديد ما هو التقييم المتوقع للزبون إن سبق وقيم هذا المنتج، كذلك يتم إدخال جميع الزبائن الموجودين في نفس الورقة ذلك لأن حساب التوقع يعتمد على مقدار ترابطه بهم، وكذلك على مقدار تقييم كل واحد منهم لهذا المنتج (كونهم يشبهونه بالاهتمامات like-minded-people).
* الخرج: prediction للزبون u مع المنتج i
* الخطوات:

1. نحسب α والذي يمثل الانحراف المعياري المثقل:

حيث هو القيمة المطلقة للترابط بين الزبون a والزبون u (حيث a هو الزبون الذي سننصحه ونحسب التوقع له، وu هو الناصح).

1. نحسب التوقع من القانون:

حيث هو متوسط تقييمات الزبون a والذي نريد حساب التوقع له.

هو الانحراف المعياري المثقل - هو تقييم الزبون u الناصح للمنتج i المراد نصح a به - هو متوسط تقييمات الزبون u - هو مقدار الترابط بين u وa.

* Insert\_user(u, r):

هذه الخوارزمية هي التي تقوم بإضافة زبون جديد إلى الشجرة في موقعه المناسب وفقاً للتقييمات التي قام بتقييمها، وربما هذا الزبون لم يقم بأي تقييم، وكل ذلك يؤثر على موقعه في الشجرة.

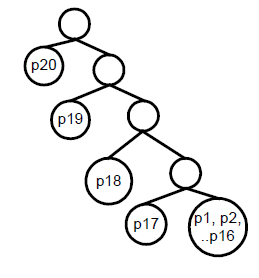
* الدخل: هو زبون u (رقمه الخاص id وبعض المعلومات الأخرى)، شعاع r من التقييمات التي قام بها عند زيارته للموقع (قد يكون شعاعاً فارغاً).
* الخرج: شجرة الـRecTree نفسها ولكن بعد إضافة الزبون عليها.
* الخطوات:

1. نضيف جميع معلوماته على العقدة الحالية (ratings وname وid و..)
2. إذا كانت العقدة الحالية هي ورقة وزاد عدد المستخدمين فيها عن الحد β نقوم باستدعاء الإجراء ConstructRecTree عليها ليقوم بتقسيمها.
3. إذا كانت outlier node وأصبح عدد المستخدمين فيها أكبر من OutlierSize تصبح ورقة ونحسب الـCorrelation Matrix فيها باستدعاء الإجراء ComputeCorrelationSimilarity.
4. إذا كانت عقدة داخلية عندها نقوم بحساب المسافة الإقليدية بين تقييمات هذا الزبون والمركز Centroid لكل عقدة من عقدتي الأبناء لهذه العقدة الداخلية (هذا ما تم حسابه في خوارزمية ConstructRecTree)، ونذهب بهذا الزبون إلى العقدة الأقرب إليه (التي تعطي مسافة إقليدية d أقل)، ويتم ذلك باستدعاء هذا التابع للإضافة عودياً على أحد العقدتين الأبناء right or left، ولكن الحالة الخاصة التي ذكرناها ولم نشرحها من قبل هي أن تكون المسافة الإقليدية بين هذا الزبون وعقدتي الأبناء متساوية تماماً، عندها لا نستطيع تحديد إلى أي طريق سيذهب كونه مترابط مع الزبائن على اليمين وكذلك على اليسار، لذلك يبقى في مكانه في هذه العقدة الداخلية ولا ينتقل إلى أي مكان، كما في الشكل التالي :



* ملاحظات عامة حول الخوارزمية:

1. عندما يقوم أحد الزبائن بتعديل تقييمه على أحد المنتجات أو تقييم منتج جديد لم يسبق له أن قيمه، نقوم بحذفه من العقدة التي يتواجد فيها وكذلك من العقد الآباء لها وصولاً إلى الجذر، وبعدها يتم إضافته من جديد إلى الشجرة، ذلك لأن تقييمه الجديد الذي قام به قد يغير من مكانه في الشجرة كلها، لذلك يجب القيام بذلك.
2. الحدود β وOutlierSize وg يتم تحديدها وفقاً لعدد الزبائن الكلي في الموقع، فالحد β عادةً ما يكون اللوغاريتم الثنائي لعدد الزبائن في الموقع، أما الـOutlierSize يكون عادةً ثلث الرقم β، والعمق الأعظمي للشجرة g يكون عدد الزبائن مقسوماً على β.
3. الهدف من الحد من عمق الشجرة منع تواجد عقد في الشجرة تحوي عدد قليل جداً من الزبائن في أحد الأفرع (عميقة وكل منها صغيرة الحجم) وفرع أخرى تحوي عدد كبير من الزبائن، كما في الشكل التالي:



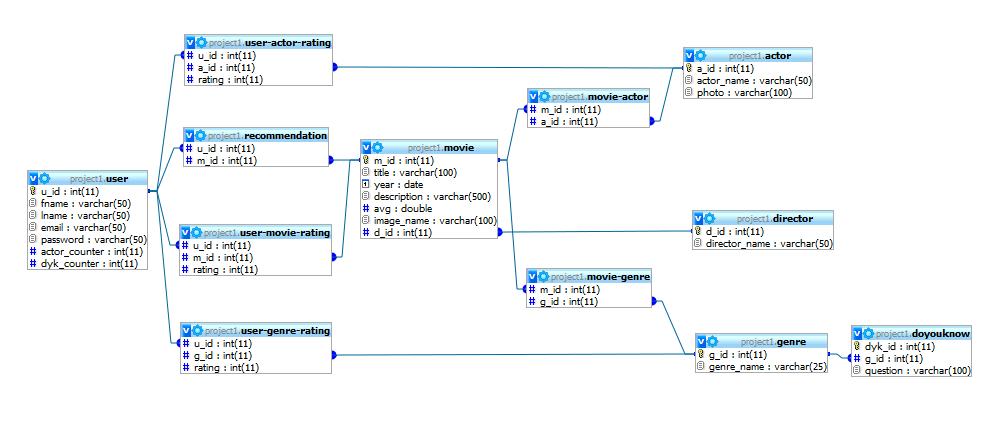
حيث p20 وp19 وp18 وp17 لا تحوي إلا عدد قليل من الزبائن ولو استمرينا بتقسيم الباقي (p1,p2,…) لأصبح الجميع كذلك، لذلك نحد من هذا النمو في العمق للشجرة.

1. يوجد في كل عقدة من العقد مؤشر على عقدة الأب، ويوجد في كل profile لزبون مؤشر على العقدة التي يوجد هو فيها (آخر عقدة وصل لها أثناء فرزه على العقد).

* مـخطط الصفوف المستخدمة:



* قاعدة بيانات الموقع:

1. مخطط الـERD: 
2. شرح الكيانات وارتباطاتها:

* تتألف من عدد من الكيانات المتربطة ببعضها البعض بعلاقات متعددة

1. **الكيان user :** جدول يحوي كافة المستخدمين الذين اشتركوا بالموقع ويحوي الواصفات التالية :

* u\_id : الرقم التسلسلي للمستخدم .
* fname : الأسم الأول للمستخدم .
* lname : الاسم الثاني للمستخدم .
* email : البريد الالكتروني للمستخدم .
* password : كلمة سر دخول المستخدم .
* actor\_counter : عدد الممثلين الذين تم اختبارهم وتقديم تقييم لهم والذين تم جلبهم من جدول actor وبهذه الحالة نكون قد منعنا التكرار .
* dyk\_counter : عدد الأسئلة المطروحة على المستخدم كي نمنع التكرار أيضا ً .

1. **الكيان movie :** جدول يحوي معلومات عن الأفلام الموجودة في الموقع ويحوي الواصفات التالية :

* m\_id : الرقم التسلسلي للفيلم .
* title : عنوان الفيلم .
* year : تاريخ انتاج الفيلم .
* description : توصيف الفيلم ببعض الجمل القصيرة .
* avg : متوسط تقييم الفيلم من جميع المستخدمين ويفيدنا بإظهار بعض النتائج على الصفحة الرئيسية .
* image\_name : صورة الفيلم التسويقية .
* d\_id : رقم مخرج الفيلم ويتم جلبه من جدول المخرجين .

1. **الكيان actor :** جدول يحوي كافة الممثلين لجميع الأفلام ويحوي الواصفات التالية :

* a\_id : رقم التسلسلي للممثل .
* actor\_name : اسم الممثل .
* photo : صورته .

1. **الكيان director :** جدول يحوي مخرجي الأفلام الموجودة بالموقع ويحوي الواصفات التالية :

* a\_id : الرقم التسلسلي للمخرج .
* director\_name : الاسم الكامل للمخرج .

1. **الكيان genre :** ويحوي تصنيف الأفلام . مثلا : رومانسي – أفلام قصيرة - رعب و يحوي الواصفات التالية :

* g\_id : الرقم التسلسلي للتصنيف .
* genre\_name : اسم التصنيف .

1. **الكيان movie\_actor :** كيان يحوي الممثلين ضمن فيلم معين ويحوي الواصفات التالية :

* m\_id : رقم الفيلم .
* a\_id : رقم الممثل ضمن الفيلم .

1. **الكيان movie\_genre :** ويحوي كافة الأفلام مع تصنيفاتها ويحوي الواصفات التالية :

* m\_id : رقم الفيلم .
* g\_id : رقم التصنيف .

1. **الكيان user-actor-rating :** يحوي كافة التقييمات الخاصة بمستخدم ما لممثل ما ويحوي الواصفات التالية :

* u\_id : رقم المستخدم .
* a\_id : رقم الممثل .
* rating : تقييم المستخدم لهذا الممثل .

1. **الكيان user-movie-rating :** يحوي كافة التقييمات الخاصة بمستخدم ما لفيلم ما ويحوي الواصفات التالية :

* u\_id : رقم المستخدم .
* m\_id : رقم الممثل .
* rating : تقييم المستخدم لهذا الفيلم .

1. **الكيان user-genre-rating :** يحوي التقييمات الخاصة بمستخدم ما لتصنيف ما ويحوي الواصفات التالية :

* u\_id : رقم المستخدم .
* g\_id : رقم التصنيف .
* rating : تقييم المستخدم لهذا التصنيف .

1. **الكيان recommendation :** كيان يحوي كافة النصائح لمستخدم ما لأفلام معينة ويحوي الواصفات التالية :

* u\_id : رقم المستخدم .
* m\_id : رقم الممثل .

1. **الكيان doyouknow :** و يحوي كافة المعلومات التي سيتم عرضها على المستخدم لأخذ انطباع عنه ويحوي الواصفات التالية :

* dyk\_id : رقم المعلومة .
* g\_id : ويحوي رقم تصنيف الفيلم الذي تم عرض المعلومة عنه .
* question : ويحوي نص السؤال .
* مراحل تنفيذ واختبار النظام:

1. بناء الخوارزمية وتطبيقها باستخدام لغة الـ++C بالمراحل التجريبية من أجل اختبار صحة خرج الخوارزمية.
2. تحويل هذه الخوارزمية إلى لغة برمجة الويب php.
3. تصميم الواجهات باستخدام برنامج التصميم Photoshop.
4. تحويل التصاميم السابقة إلى صفحات ويب باستخدام كل من لغتي html وcss.
5. جعل هذه الصفحات وظيفية بربطها مع قواعد معطيات الموقع وخوارزمية التسويق المستخدمة باستخدام كل من لغتي php وsql.

* وظيفية تعني أنها ليست ثابتة دوماً وإنما هي ديناميكية ترتبط بماهية الزبون الذي يتصفح الصفحة ونوعية الأفلام التي يتم عرضها له وماهية الصفحة التي يتصفحها.

1. ربط قاعدة المعطيات التي تحوي تقييمات الزبائن (ratings) على المنتجات مع الخوارزمية وجعل هذه التقييمات دخلاً لها، وتخزين خرج الخوارزمية الذي يمثل المنتجات التي ننصح الزبون بها أي الـRecommendations في جدول الـRecommendations ضمن قاعدة بيانات الموقع.

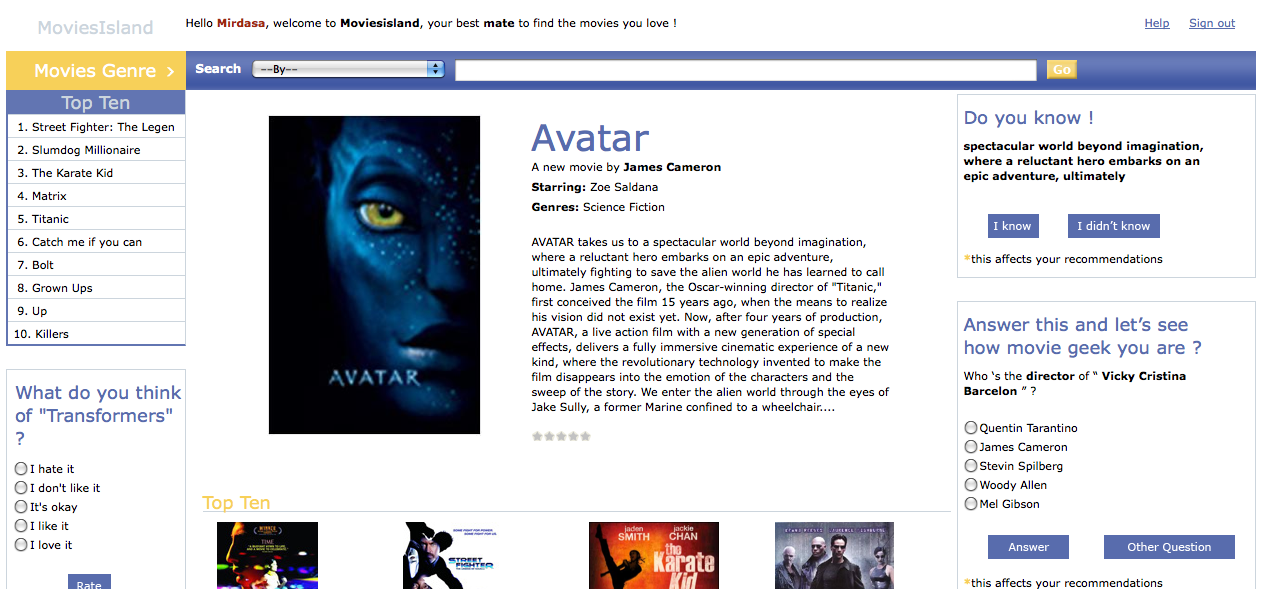
**Top ten:**

The most popular movies in the website

* دليل استخدام النظام:

**Advertisement:**

It's the newest movie in the website



**Do you know:**

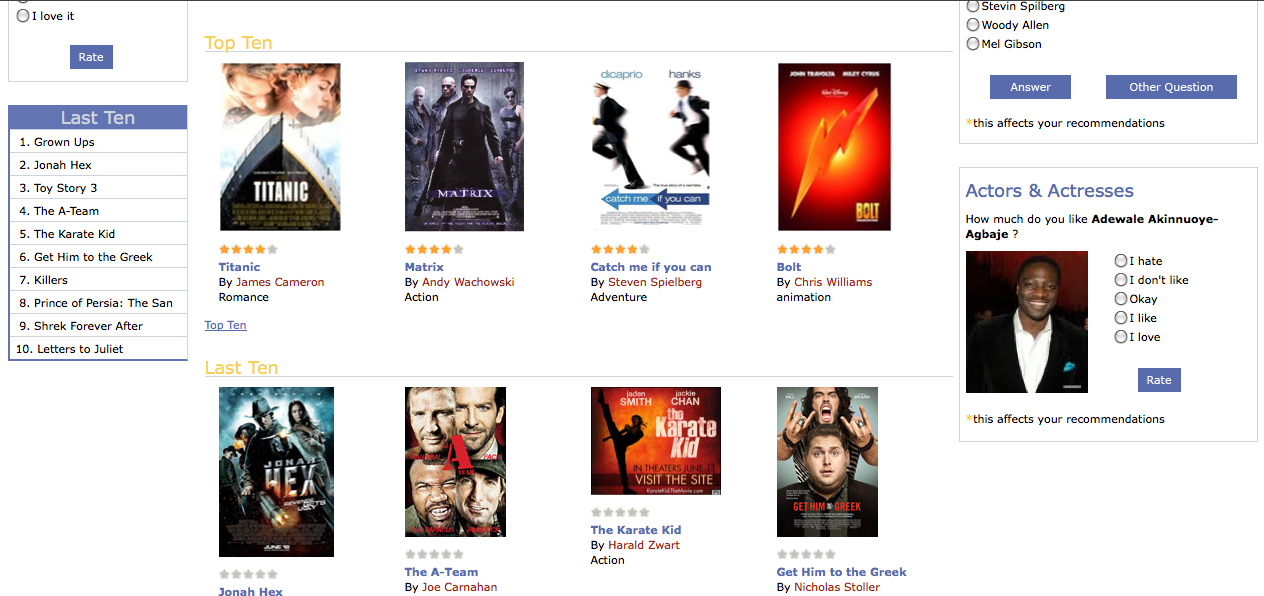
Yes/no question about some info affects the recommendations

**Competition:**

Some questions about some movie, actor or director which generated automatically

**Random Rate:**

We pick a random movie for the user to rate it



**Actors & Actresses:**

Randomly chosen actor or actress to be rated from the user which affects the recommendations

**Last ten:**

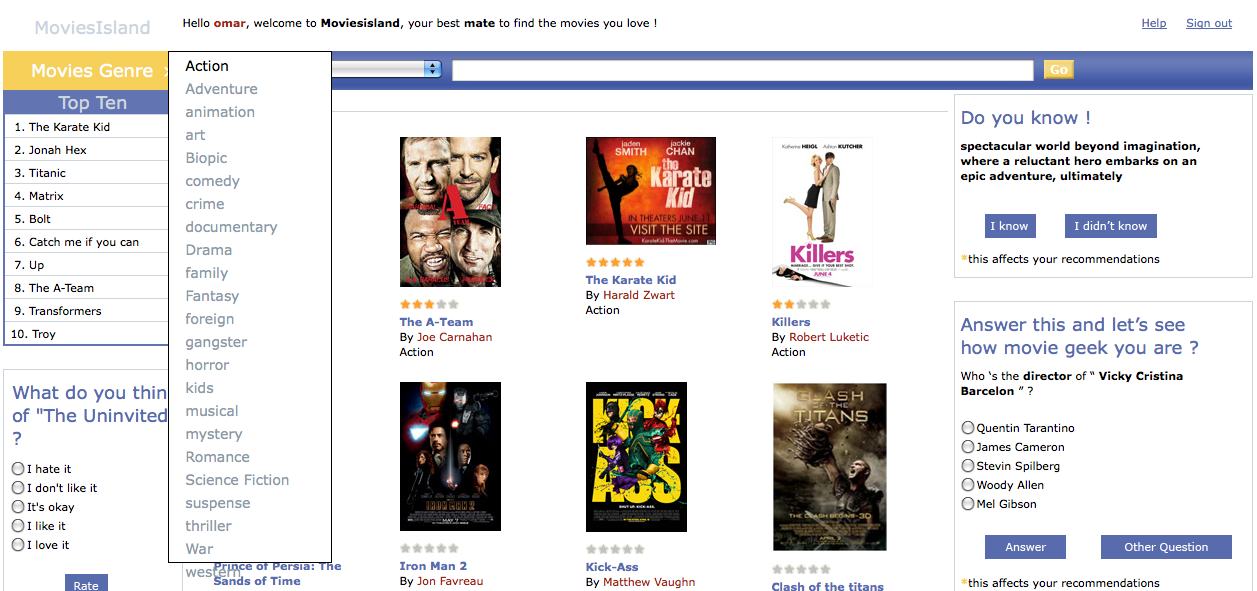
The latest movies in the website

**Header:**

The current user's name

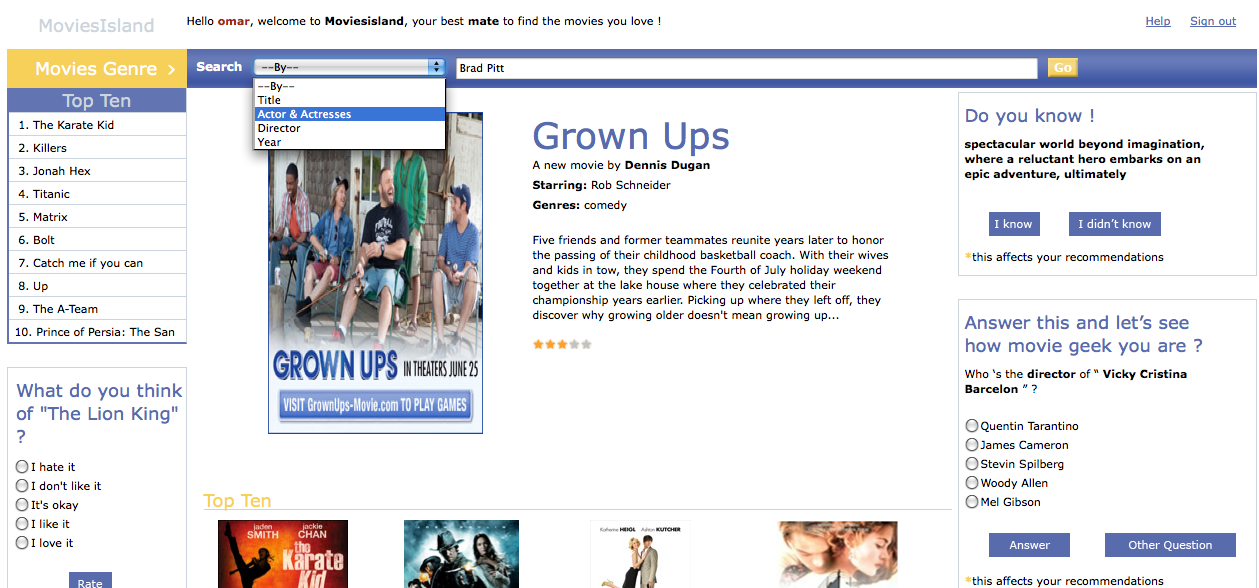
**Genre list:**

The list of movies' genres which can be clicked to show all the movies which their genre is the chosen



**Search list:**

The search keywords which the user can search for. The result of the current search example would show the movies that Brad Pitt took part in.



* خاتمة:
* في نهاية هذا التقرير سنذكر بشكل مختصر عما قمنا ببنائه فيما سبق، حيث قمنا باستخدام خوارزمية تسويق تخضع للفلترة التعاونية Collaborative filtering تسمى بـRecTree التي هي اختصار للإسم Recommendation Tree وذلك من أجل تحقيق سياسة التسويق من واحد لواحد التي تضمن أن يتم اقتراح أو نصح كل زبون على حدا بالمنتجات التي قد يرغب فيها، مما يعني أن الزبون x ليس بالضرورة أن يتم نصحه بنفس المنتجات التي يتم نصح الزبون y فيها وإنما يتبع ذلك كله إلى سلوك كل واحد منهما على الموقع وكيفية تقييمه للمنتجات المعروضة على الموقع.
* كما ونذكر بأن اختيار الويب كبيئة لبناء الخوارزمية يرجع إلى متاحية الويب وتوفره في جميع أنحاء العالم تقريباً كوسيلة جيدة للتواصل والتسويق.
* آفاق مستقبلية:
* مما يجدر ذكره بأن المنتجات التي يتم تسويقها في الموقع حالياً هي عبارة عن أفلام، وما اختيار هذا الصنف من المنتجات إلا لأنه متداول بشكل كبير فيما بين جميع فئات المجتمع تقريباً، فالكتب ليست الأكثر تداولاً في مجتمعاتنا مثلاً، ولكن هذا لا يعني أن الموقع لا يمكن أن يحوي أنواع المنتجات الأخرى، فذلك لا مشكلة فيه لأن الخوارزمية المختارة تعمل على جميع أنواع المنتجات بغض النظر عن طبيعتها، فكما نلاحظ من مخطط الصفوف المستخدمة أن صف المنتجات قد وصف بأنه item مجرد في الخوارزمية، وهذا الشيء الذي يعني إلى أن الخوارزمية لا ترتبط بنوعية معينة من المنتجات مما قد يساعد في تطوير الموقع مستقبلاً ليحوي أي نوعية من المنتجات سواءً أفلام أو كتب أو غيرها.
* المراجع المستخدمة:

1. RecTree: A Linear Collaborative Filtering Algorithm

Sonny Han Seng Chee -

M.Sc., University of Toronto, 1992.

1. Constrained K-means Clustering with Background Knowledge

Kiri Wagsta, Claire Cardie -

Department of Computer Science, Cornell University, Ithaca, NY 14853 USA

Seth Rogers, Stefan Schroedl, DaimlerChrysler -  
 Research and Technology Center, 1510 Page Mill Road, Palo Alto, CA 94304 USA

انتهى التقرير