

Evaluating similarity based on correlation

تقييم التشابه على اساس العلاقة

يتم تقييم التشابه على اساس التشابه في تقييم العملاء بحيث الاخذ بتفضل العميل للمنتجات بعين الاعتبار .

ماهو مدى الترابط بين منتجين بناء على تقييم العملاء ؟

هنا نستخدم R-Persona لاستخراج التشابه بين منتجات قام العملاء باختيارها
بمعنى : استخراج منتج له نفس قيمة التقييم مع منتج اختاره العميل بالفعل

{ اخترت فلم نعطيك فلم له نفس تقييمات الفلم الذي اخترته }
{ اخترت وجبة نعطيك وجبة لها نفس تقييمات الوجبة التي اخترتها }
{ اخترت سياره نعطيك سياره لها نفس تقييمات السيتره التي اخترتها }

ماهو R-Persona :

١ = علاقة **ايجابية** قوية
٠ = لا يوجد علاقة
-١ = علاقة **سلبية** قوية

هذا النوع من التحليل سوف يطابق المنتجات بناء على مراجعات العملاء ، من الممكن ان نجد في الداتا نس العميل قام بتقييم اكثر من منتج

الخطوات :

١-تحميل المكتابات وترتيب المخرجات :

بعد تحميل البيانات يجب ان نستخرج المخرجات التي نحتاجها منها فقط عن طريق عمل Object جديد يتضمن ما نريده من كل داتا :

The Data :

fream = [رمز العميل ، رمز المطعم ، تقييم المطعم ، تقييم الاكل ، تقييم الخدمه]
geo = ['name' , 'placeid'] , places = geo [رمز المطعم ، نوع المطبخ]
coisine = [رمز المطعم ، نوع المطبخ]

مجموع البيانات التي نريدها

{ رمز العميل ، رمز المطعم ، تقييم المطعم ، تقييم الطعام ، تقييم الخدمه ، اسم المطعم ، نوع المطبخ }

٢- استخراج متوسط التقييمات :

نقوم باستخراج المتوسط mean لكل تقييمات العملاء لكل مطعم ، عن طريق عمل داتا جديده تضم مجموعه من { رمز المطعم ، تقييم المطعم }

Object	new data	من داتا	method	العمودين	method
rating	pd.datafram	frame	groupby	('placeid'),['rating']	mean()

rating = pd.DataFrame (fream.groupby('placeid')['rating'].mean())

النتيجه : متوسط تفقييمات العملاء لكل مطعم

٣- استخراج عدد التقييمات لكل مطعم :

- كم عدد التقييمات التي حصل عليها كل مطعم ؟ وهنا نستخدم طريقة الاكثر شهرة

العمود الذي نريد اضافته للداتا	الداتا التي عملناها	الداتا	من داتا	method	الحقول التي نريدها في الداتا	method
['rating_count']	rating	pd.dataframe	fream	groupby	('placeid')['rating']	count()

**rating ['rating_count'] = pd.DataFrame (fream.groupby ('placeid') ['rating']
.count())**

النتيجه : جدول يتضمن { اسم المطعم ، التقييم ، عدد التقييمات التي حصل عليها }

٤- التحليل الوصفي :

rating.describe()

ما يقدمه لنا :

العدد : عدد المطاعم التي تم تقييمها في الداتا التي عملناها
ماكس : اكبر قيمة تقييم في الداتا و العدد هو عدد التقييمات

٥- المطعم الاعلى تقييم :

يمكننا استخراج المطعم الاعلى تقييم عن طريق جمع عدد التقييمات لكل مطعم وترتيبها تنازليا فيكون الاول هو الاعلى تقييم .

اول 5	تنازليا	العمود	method	الداتا اللي جمعنا فيها
head()	ascending=fals	('rating_count')	sort_values	rating .

rating.sort_values ('rating_count' , ascending = false).head()

النتيجة : المطعم الاعلى تقييما مرتب بشكل تنازلي

٦- استخراج اسم المطعم الاعلى تقييما :

نقوم بتطبيق فلتر يعطينا قيمة True ونمرر له العمود لسيتخرج لنا اسم المطعم

رمز المطعم اللي نريد اسمه	يساوي صحيحه	من عمود	من هذه الداتا	الداتا اللي عملناها بالاول اللي تتضمن الاسم
1350038	==	' places id '	places	places

places [places [' placeld'] == 1350038

النتيجة : جدول { الاتدكس . رمز المطعم ، اسم المطعم }

٧- استخراج نوع المطبخ للمطعم الاكثر عدد تقييم

نقوم بتطبيق فلتر لاستخراج نوع المطعم نمرر له العمود من الداتا مثل السابق

cuisine [cuisine [' placeld'] == 13582238

النتيجة : جدول يتضمن { الاتدكس ، رمز المطعم ، نوع المطبخ }

٨- تجهيز البيانات للتحليل :

- عمل مصفوفة تقييمات العملاء :

مصفوفة تقييمات العملاء هي جدول يوضح تقييمات العملاء بالكامل في كل الداتا وتشير الى تقييم كل عميل امام المطعم الي قام بتقييمه بقيمة التقييم ويستخدم Poivot Tabel لعرض المصفوفة .

اسم العمود الجديد	العمود	من داتا	method	Object
columns=places	values='rating'	(data=fream	pd.poviot_table	places_crostab

places_crosstab = pd.pivot_table (data = fream, values = 'rating' , columns = ' places')

النتيجة : جدول يتضمن الكثير من **NAN** وهذا بسبب ان ليس الجميع قاموا بتقييم جميع المطاعم ، ويتضمن الجدول { التقييم ، رمز المطعم ، رمز العميل } القيمة موجد فقط عند المطعم الذي قام العميل بتقييمه .

٩- ايجاد الاماكن ذات العلاقة :

١- يجب ان نقوم بحذف البيانات التي تتضمن **NAN** وذلك عن طريق انشاء داتا جديده تتضمن مانرديه فقط وهو مطعمنا .

Object	الداتا اللي عملناها سابقا • من هذه الداتا	رمز المطعم الاكثر شهرة ، مطعمنا
Tortas_rating	places_crosstab	1385002

٢- نقوم بعمل فلتر لكي يظهر لنا التقييمات الخاصه بمطعمنا فقط واستخراج التقييمات الي اعلى من / تساوي الصفر

Tortas_rating [Tortas_rating >= 0]

النتيجة : يظهر لنا جميع التقييمات التي اكبر من او تساوي صفر (جميع التقييمات بالتفصيل)

١٠- لايجاد علاقته التشابه :

سوف نستخرج الان العلاقة بين مطعمنا والمطاعم الباقية لنعرف الاكثر تشابهها عن طريق استخراج R-Persona correlation

Object	الداتا اللي قمنا بعملها سابقا	method	الداتا التي نريد المقارنه بها
similer_to_tortas	places_crosstab	corrwith	(tortas_rating)

similer_to_tortas = places_crosstab , corrwith (tortas_rating)

النتيجة : سوف يستخرج لنا مصفوفه نقوم بتحويلها الى داتا فريم

Object	new Data	الداتا اللي عملناها	اضافه العمود	اسم العمود الجديد للاضافه
corr_tortas	pd.dataframe	(similer_to_torta	colmun	['pearsonsR'])

corr_tortas = pd.DataFrame (similer_to_tortas , columns = [' pearsonR']

النتيجة : جدول يتضمن {رمز المطعم ، **Pearsons R** } ، القيم في همود **Persona R** هي قيمة التقييم **R-Persona** , العلاقة بين كل المطاعم مع مطعمنا .

ملاحظة مهمة جدا :

يمكن ان نحد مطعم له نفس العلاقة مع مطعمنا ولكنه يتضمن عدد قليل من التقييمات . لذلك وصل الى هذه العلاقة وهذا غير صحيح .

وهنا يجب ان نأخذ بالاعتبار مدى شهرة المطعم { ليس اي مطعم صغير يشابه مطعمنا }

١١- لاستخراج المطاعم ذات التقييمات المناسبه :

نقوم بدمج الداتا مع داتا التقييم لكل المطاعم وهي rating

دمج في العمود	من الداتا	method	الداتا السابقة	Object
['rating_count']	(rating	join	corr_tortas.	tortas_corr_summary

tortas_corr_summary = corr_tortas.join (rating [' rating_count'])

الان سنقوم بانشاء فلتر يتضمن من هذه الداتا المطاعم الحاصله على اكثر من 10 تقييمات وترتيبه انتازليا

**tortas_corr_summary (tortas_corr_summary [' rating_count'] >=10
) .sort_values (' personaR' , ascednce = False) .head()**

النتيجة : جدول يتضمن { مز المطعم ، تقييم R , عدد التقييمات اكثر من / تساوي 10 }

الان بعد هذه النتيجة نلاحظ ان هناك مطاعم لديها **persona R = 1**

وهي ليست ذات معنى لانها تعني ان هناك العميل مكرر ، اي انه كان هناك عميل واحد قام بمراجعة كلا المطعمين (مطعمنا ومطعم اخر) واعطى نفس النتيجة لكلا المطعمين

١٢- حذف المخرجات القليله :

سنقوم بحذف الاماكن التي تقييماتها قليلة عن طريق عمل داتا فريم جديده نتضمن الباقي فقط (7) من ضمنهم مطعمنا .

اسم العمود في الداتا الجديد	اندكس بعدد الصفوف	رمز ال ٧ مطاعم التي نريد	new Data Fram	Object
colmun ['places']	index = np.arrange(7)	([id ,id ,id, ..]	pd.datafream	places_corr_tortas

places_corr_tortas = pd.DataFrame ([id,id,id,..] , index = np.arrange(7) , colmun ['placeId'])

١٣ - دمج الداتا المستخرجه مع الدتا التي تتضمن نوع المطبخ :
نقوم بدمج الداتا لكي نعرف نوع المطبخ لكل مطعم و ايجاد المشابه لمطبخ مطعمنا

Object	method	الداتا عملناها سابقا	داتا التي تتضمن نوع المطبخ	الدمج في عمود
summury	pd.merge	(places_corr_tor tas	cuisine	on='placeId')

summury = pd.merge (places_corr_torats , cuisine , on = 'placeId')

النتيجة : جدول يتضمن { اندكس ، رمز المطعم ، نوع المطبخ }

عندما نطبع هذا الكود نحصل على ٥ نتائج فقط مع اننا قمنا بادخال ٧ مدخلات والسبب هو :
انه لم يتم ادراج جميع المطاعم في الدتا **cuisine**
مع ذلك :
ان ما نراه هنا انه من بين افضل 6 مطاعم مرتبطة بمطعمنا يوجد على الاقل واحد لديه نفس المطبخ

- الان سنقوم بتطبيق فلتر للحصول على اسم هذا المطعم المرتبط مع مطعمنا

places (places [' placeId'] = 12500023)

النتيجة : اسم المطعم الذي يرتبط وله نفس العلاقة مع مطعمنا

١٤ - التقييم :

لتقييم مدى ملائمة مقياس التشابه حقا ، يجب الاخذ بمجموعة الاحتمالات المتباكمها مثل عدد المانكولات في قائمه الطعام لذلك سوف نقوم بالتحليل الوصفي :

cuisine [' Rcuisine '].describe()

النتيجة :

العدد : عدد المطابخ بالكامل
يونييك : عدد المطابخ الاصلي
توب : افضل المطابخ
فريك : التردد

النتيجة التحليله :

حصلنا على 6 من افضل المطاعم التي كانت متشابهه مع مطعمنا على اساس الارتباط والشهره .
من بين هذه المطاعم يوجد مطعم واحد يقدم نفس مطبخنا .
وهنا يبدو ان نضامنا يعمل بشكل رائع في بناء التوصيه
والان نستطيع ان نقدم التوصيه للعملاء مطعمنا بزيارة المطعم الجديد الذي تم استخراجه في الجدول .