

حل الطلب الثاني:

المقدمة:

النتيجة المرجوة من هذا الطلب هو إيجاد أكثر خمس مسارات تكراراً لكل من هذه الأعمدة (Country & solution)، حيث أن المسار أو (path) هو تسلسل الأنشطة في عمود (types) التي مر بها (customer) مرتبة زمنياً، ولقد لاحظت أن ملف (excel) النظيف هو فعلاً مرتب زمنياً على حسب عمودي (account_id & activity_date) بحيث ليس هناك حاجة لأن أقوم بإعادة ترتيب الأعمدة بناءً على عمود التاريخ والعمود الذي يعبر عن العميل، وبذلك لكي أبدأ بالخطوات المهمة يجب عليّ أولاً أن أقوم بدمج أنشطة كل عميل في خلية واحدة بحيث تكون مرتبة زمنياً وبعدها أستطيع عمل المسارات لكل عميل ثم معرفة تكرارات كل مسار بناءً على عمودي البلد ونوع الحل ثم وضع رقم التكرارات في عمود جديد ومن ثم أقوم بإظهار أكثر خمس مسارات تكراراً للحصول على النتيجة النهائية.

البرامج المستخدمة:

استخدمت (Jupiter).

المكتبات المستخدمة:

(Pandas).

شرح الكود وطريقة الحل:

قمت باستيراد مكتبة (pandas) بواسطة الأمر (import) وقمت باستخدام (as) لعمل اختصار للمكتبة وسميته (pd) كما هو متعارف عليه في عالم البرمجة عندما يستخدمون هذه المكتبة يقومون بتسميتها (pd).

```
# Import the Library of Pandas.
import pandas as pd
```

قمت بقراءة ملف الاكسيل وتحويل الملف إلى (DataFrame) بواسطة الأمر التالي (read_excel) الموجود داخل مكتبة (pandas) وتم وضع حرف (r) قبل المسار لكي نجعل بايثون يعرف أن هذا مسار وليس شيء آخر، و (engine) هو البرنامج الداخلي الذي يستخدمه (pandas) لفتح ملف الاكسيل وبما أن ملف الاكسيل لاحقته (xlsx) قمت باستخدام (openpyxl) لكي يتم فتح الملف.

```
# Read The path of Excel file and save it in a variable called (df) --> it means DataFrame.
df = pd.read_excel(r"D:\MLT\data_clean.xlsx", engine="openpyxl")
```

بواسطة الأمر (df.columns) أستطيع معرفة أسماء الأعمدة الموجودة في ملف الاكسيل، واحتاج هذه التعليمة لكي أستطيع كتابة أسماء الأعمدة بشكل صحيح عندما أحتاج.

```
df.columns
```

```
Index(['account_id', 'SourceSystem', 'activity_date', 'who_id',
       'opportunity_id', 'opportunity_stage', 'is_lead', 'types', 'Country',
       'solution'],
      dtype='object')
```

ولكي أستطيع عمل مسار لكل عميل قمت باستخدام أمر (groupby) لكي يقوم بتجميع الصفوف المتشابهة بناءً على هذه الأعمدة الثلاثة (Country & solution & account_id)، ثم في عمود (types) بواسطة الأمر (apply) قمت بعمل قائمة حيث هذا الأمر يقوم بتطبيق ذلك على كل العملاء بحيث يتم وضع كافة أنشطة العميل الواحد داخل قائمة بناءً على الثلاث الأعمدة السابقة، ثم قمت باستخدام (reset.index()) لكي أعيد الجدول إلى وضعه الطبيعي حيث بعد استخدام

الأمر (groupby) يجعل ترتيب الجدول مختلفاً لذلك قمت باستخدام أمر يعيد الجدول إلى شكله الطبيعي ثم قمت بحفظ الأوامر التالية داخل متغير اسميته (path_for_every_customer).

```
path_for_every_customer = ( df.groupby(["account_id", "Country", "solution"])["types"].apply(list).reset_index() )
```

ولكي أرى نتيجة التنفيذ السابق قمت باستخدام أمر (head) لرؤية أول خمس أسطر بالجدول حيث أن عمود (type) يحتوي على قائمة بأنشطة كل عميل.

```
path_for_every_customer.head()
```

	account_id	Country	solution	types
0	0010L00001hVmFhQAK	United States	MRS	[Follow Up, Inbound Call, Inbound Call, Inboun...
1	0010L00001hVxd6QAC	United States	MRS	[Meeting, Meeting, Review, Email, Email, Email...
2	0010L00001hVyJQQA0	United States	MRS	[Inbound Call, Inbound Call, Inbound Call, Inb...
3	0010L00001hW1cAQAS	United States	MRS	[Inbound Call]
4	0010L00001ijzeFQAQ	United States	MRS	[Email, Email, Email, Inbound Call, Inbound Ca...

لكي أقوم بتحويل القائمة داخل عمود (types) إلى مسار نصي قمت باستخدام الأمر (str) واستخدمت (join) لتضمن هذه العلامة (→) بين عناصر كل صف.

```
path_for_every_customer["types"] = path_for_every_customer["types"].str.join(" --> ")
```

ولكي أستطيع رؤية نتيجة الأمر السابق قمت باستخدام الأمر التالي لرؤية أول خمس أسطر.

```
path_for_every_customer.head()
```

	account_id	Country	solution	types
0	0010L00001hVmFhQAK	United States	MRS	Follow Up --> Inbound Call --> Inbound Call --...
1	0010L00001hVxd6QAC	United States	MRS	Meeting --> Meeting --> Review --> Email --> E...
2	0010L00001hVyJQQA0	United States	MRS	Inbound Call --> Inbound Call --> Inbound Call...
3	0010L00001hW1cAQAS	United States	MRS	Inbound Call
4	0010L00001ijzeFQAQ	United States	MRS	Email --> Email --> Email --> Inbound Call -->...

لكي أستطيع حساب عدد تكرار كل مسار قمت باستخدام أولاً أمر (groupby) لوضع الأسطر المتشابهة لثلاث أعمدة ثم استخدمت أمر (size) لمعرفة عدد الصفوف في كل مجموعة، ثم استخدمت الأمر الذي يعيد الجدول إلى حالته الطبيعية (reset.index) ثم قمت بتسمية عمود جديد اسميته (duplicates) حيث يحتوي على عدد مرات تكرار المسارات وقمت بحفظ هذا الخطوات من الأوامر داخل متغير اسميته (path_duplicates).

```
path_duplicates = ( path_for_every_customer.groupby(["Country", "solution", "types"]).size().reset_index(name = "duplicates") )
```

ولرؤية أول خمس أسطر من نتيجة تنفيذ الأمر السابق قمت باستخدام الأمر (head) حيث نلاحظ مثلاً أن المسار (Email) لدولة (Austria) ولنوع الحل (MRS) لقد تكرر هذا المسار (22) أي أنه هناك (22) عميل قاموا بعمل نفس المسار لنفس البلد ونوع الحل.

```
path_duplicates.head()
```

	Country	solution	types	duplicates
0	Austria	MRS	Email	22
1	Austria	MRS	Email --> Email	19
2	Austria	MRS	Email --> Email --> Email	5
3	Austria	MRS	Email --> Email --> Email --> Email	6
4	Austria	MRS	Email --> Email --> Email --> Email --> Email	4

لكي أقوم بمعرفة أكثر خمس مسارات تكرر لكل بلد ولكل نوع حل قمت أولاً بترتيب جدول المسارات السابق حيث قمت بترتيب عمودي البلد ونوع الحل ترتيباً تصاعدياً وقمت بترتيب عمود (duplicates) ترتيباً تنازلياً لكي يكون الأكثر تكراراً من الأعلى للأسفل وبعدها قمت باستخدام الأمر (groupby) لكلا العمودين (Country & solution) ثم قمت باستخدام أمر (head(5)) لرؤية أعلى خمس مسارات تكرر لكل بلد ونوع حل وقمت بحفظ هذه التنفيذ داخل المتغير (top_five_paths).

```
top_five_paths = (
    path_duplicates.sort_values(["Country", "solution", "duplicates"], ascending = [True, True, False]).groupby(["Country", "solution"]).head(5)
)
```

لعرض أعلى خمس مسارات تكرر لكل بلد ونوع حل استخدمت الأمر التالي حيث كتبت اسم المتغير ومن ثم ظهرت النتائج التالية حيث أصبح الجدول عبارة عن 116 صف وأربع أعمدة.

top_five_paths				
	Country	solution	types	duplicates
0	Austria	MRS	Email	22
1	Austria	MRS	Email --> Email	19
3	Austria	MRS	Email --> Email --> Email --> Email	6
2	Austria	MRS	Email --> Email --> Email	5
6	Austria	MRS	Email --> Email --> Email --> Email --> Email ...	5
...
7294	United States	PLS	Inbound Call	6
7276	United States	PLS	Email --> Email	4
7295	United States	PLS	Inbound Call --> Email	4
7275	United States	PLS	Email	3
7280	United States	PLS	Email --> Inbound Call	3

116 rows × 4 columns

وفي النهاية قمت بحفظ النتائج داخل ملف اكسيل (Excel) وسميته (top_five_paths) وجعلت (index = False) لكي لا يظهر رقم (index) داخل ملف الاكسيل.

```
# Transform (top fie paths) into Excel file.
top_five_paths.to_excel("top_five_paths.xlsx", index= False)
```