FUZZY MATCHING : Conditional FUZZ PARTIAL TOKEN MATCH for Keyword2

Description :

This is a Fuzzy Matching Project. There are two Files involved.

1. Raw data and
2. Keyword List

Raw data : Includes columns like :

```
1 raw_data.head(2)
```

	HS_Code	Detailed_Product	HS_sub
0	90181100	trolley for btl -08 ecg l line baik / baru	9018
1	90183200	microcatheter.merit maestro┐ .2.8f.2.4f.130 cm ...	9018

Keyword list looks like this :

	A	B
1	**Product_Name_short_First Keyword**	**Product_Name_short_Second Keyword**
2	dialysis	catheter
3	Magnum	magnum ii
4	Biopsy	plier
5	Biopsy	nipper
6	Biopsy	pincer
7	Biopsy	forcep
8	Coseal	CARBOSEAL
9	Gelita	thp2628x150b
10	Gelita	thp2830x100b

The Keyword File has two main columns :

- Keyword 1 --> Product_Name_short_First_Keyword
- Keyword 2 --> Product_Name_short_Second_Keyword

Step 1 : Run the Fuzz Partial Match on raw_data['Detailed_Product'] for Keyword 1

We import Keyword File and run Fuzz Partial Match for matching Keyword1, that is "Product_Name_short_First_Keyword" with the "raw_data['Detailed_Product']" and get the match score.

```
In [422]: device_list_second.head(2)
```

Out[422]:

	Product_Type_First_Keyword	Product_Name_short_First_Keyword	Product_Type_Second_Keyword	Product_Name_short_Second_Keyword
0	0_EP_Accessories	402561	NaN	NaN
1	0_EP_Accessories	402566	NaN	NaN

```
In [22]: raw_data.head(2)
```

Out[22]:

	HS_Code	Detailed_Product	HS_sub
0	90181100	trolley for btl -08 ecg l line baik / baru	9018
1	90183200	microcatheter.merit maestro┐.2.8f.2.4f.130 cm ...	9018

Running Fuzz Partial Token match for Keyword1

```
In [390]: raw_data = raw_data.apply(lambda x: x.astype(str).str.lower())

In [425]: choices_brand = device_list_second['Product_Name_short_First_Keyword']

In [426]: choices_brand.shape
Out[426]: (9847,)

In [427]: def matchfunc_pr(x,choices):
              option = process.extract(x, choices, limit=1, scorer=fuzz.partial_ratio)

              if option[0][1]>75:
                  return option
              else:
                  return ""

In [428]: raw_data['Keyword1_Match_PARTIALRatio']= raw_data['Detailed_Product'].apply(lambda x: matchfunc_pr(x,choices_brand))
          raw_data['Keyword1_PARTIAL_match'] = raw_data['Keyword1_Match_PARTIALRatio'].apply(lambda x: x[0][0] if x != '' else '')
          raw_data['Keyword1_PARTIAL_score'] = raw_data['Keyword1_Match_PARTIALRatio'].apply(lambda x: x[0][1] if x != '' else '')
```

And the Output generated looks like :

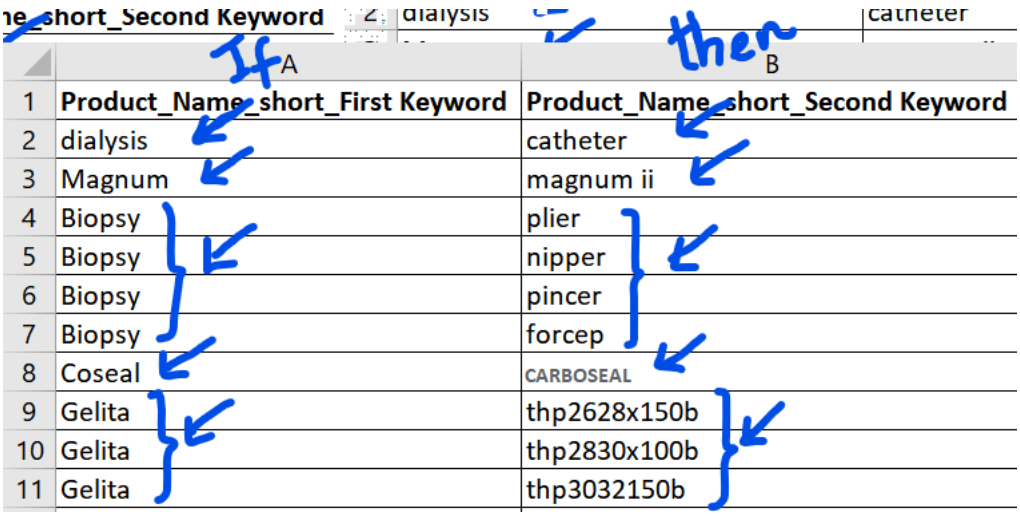
```
In [429]: raw_data.drop(['Keyword1_Match_PARTIALRatio'],axis=1, inplace = True)

In [430]: raw_data.head(2)
```

Out[430]:

	Month	HS_Code	Detailed_Product	HS_sub	Keyword1_PARTIAL_match	Keyword1_PARTIAL_score
0	202101	90181100	trolley for btl -08 ecg l line baik / baru	9018	ecg	100
1	202101	90181100	bt -08 mix/sd/spiro mains power switch with ca...	9018	malis	80

Step 2 : Run the Fuzz Partial Match on raw_data['Detailed_Product'] for Keyword 2

Logic : If Keyword 1 is present in raw_data [Keyword_PARTIAL_match] then select the corresponding Keyword 2 and run Fuzz Partial Token match for those Keywords , run Partial Token Match against "Detailed_Product" and get the Score



Approach :

1. Now our Keyword list would be raw_data: Keyword1_PARTIAL_match

	Month	HS_Code	Detailed_Product	HS_sub	Keyword1_PARTIAL_match	Keyword1_PARTIAL_score
0	202101	90181100	trolley for btl -08 ecg l line baik / baru	9018	ecg	100
1	202101	90181100	bt -08 mix/sd/spiro mains power switch with ca...	9018	malis	80

- We copy that column to an empty List :

K1list = raw_data['Keyword1_PARTIAL_match']

- Keep only the unique values:

K1list = list(set(K1list))

- Then we create a while loop :

```
- For length in range equal to the len of K1list

- If the first word in K1list matches "Product_Name_short_First_Keyword" then
extract the adjacent values of

"Product_Name_short_First_Keyword" that is "Product_Name_short_Second_Keyword"
to the

new column : **device_list_second"['K2']**

- Now these chunk of keywords in device_list_second"['K2'] will be the keyword
s to be matched.

- So, we check for the raw_data.loc where "Keyword1_PARTIAL_Match" is equal to
first word in K1list

- We now have a slice of raw_data (df_temp)

- What the While loop is doing is , getting the Keywords2 in device_list_secon
d"['K2']
```

- Matching those words against the "Detailed_Product" where first word in K1list matches the Keyword1_PARTIAL_Match

- We run Fuzz partial match for df_temp

- Keep appending this newly updated slice (df_temp) to the Dataframe list as and when the new

df_temp is generated in the While loop.

- Towards the end we have the Final df "data" with all the df_temp appended.

```
#Fuzz partial match Function that we are applying to the sclice of data in the while loop

def matchfunc_pr(x,choices):
    option = process.extract(x, choices, limit=1, scorer=fuzz.partial_ratio)

    if option[0][1]>75:
        return option
    else:
        return ""
```

In [ ]:

```
K1list = raw_data['Keyword1_PARTIAL_match']
K1list = list(set(K1list))
K1list.remove("")
data = pd.DataFrame([])

i = 0
while i < len(K1list):

        device_list_second['K2'] = device_list_second.loc
               [device_list_second['Product_Name_short_First_Keyword'] == K1list[i],'Product_Name_short_Second_Keyword']
        device_list_second['K2'] = device_list_second['K2'].astype(str)
        choices_brand =device_list_second['K2']
        df_temp = raw_data.loc[raw_data['Keyword1_PARTIAL_match'] == K1list[i]]
        df_temp = df_temp.copy()
        df_temp['Keyword2_Match_PARTIALRatio']= df_temp['Detailed_Product'].apply(lambda x: matchfunc_pr(x,choices_brand))
        df_temp['Keyword2_PARTIAL_match'] = df_temp['Keyword2_Match_PARTIALRatio'].
                                                apply(lambda x: x[0][0] if x != '' else '')
        df_temp['Keyword2_PARTIAL_score'] = df_temp['Keyword2_Match_PARTIALRatio'].
                                                apply(lambda x: x[0][1] if x != '' else '')
        data = data.append(df_temp, ignore_index=True)
        i = i + 1
```

We have the new Data Frame which looks like:

In [433]:
```
data.shape
```
Out[433]: (74441, 9)

In [434]:
```
data.head(2)
```
Out[434]:

Month	HS_Code	Detailed_Product	HS_sub	Keyword1_PARTIAL_match	Keyword1_PARTIAL_score	Keyword2_Match_PARTIALRatio	Keyword2_PARTIAL_match
202109	90189090	blade f/bv380r 4-prong 16mm x 60mm	9018	bv580r	83		
202109	90189090	blade f/bv380r 4-prong 16mm x 40mm	9018	bv580r	83		

```
In [435]:
1 raw_data = data.copy()
```

```
In [436]:
1 raw_data.drop(['Keyword2_Match_PARTIALRatio'],axis=1, inplace = True)
```

```
In [437]:
1 raw_data.head(4)
```

Out[437]:

	Month	HS_Code	Detailed_Product	HS_sub	Keyword1_PARTIAL_match	Keyword1_PARTIAL_score	Keyword2_PARTIAL_match	Keyword2_PARTIAL_score
0	202109	90189090	blade f/bv380r 4-prong 16mm x 60mm	9018	bv580r	83		

The Column "Keyword2_PARTIAL_match" will contain the matched word.

The column "Keyword2_PARTIAL_score" will contain the match Score.

Owner : Aisha Khalid