DataFrame : Column : Text : Tokenize ::: Shuffle ::: Detokenize

Re-order or Shuffle the words randomly in a Sentence or a String:

Applying the function to the DataFrame column.

Note : Always make sure the column on which you are going to apply text functions to is an "str" type column.

How to set the column dtype to str ?

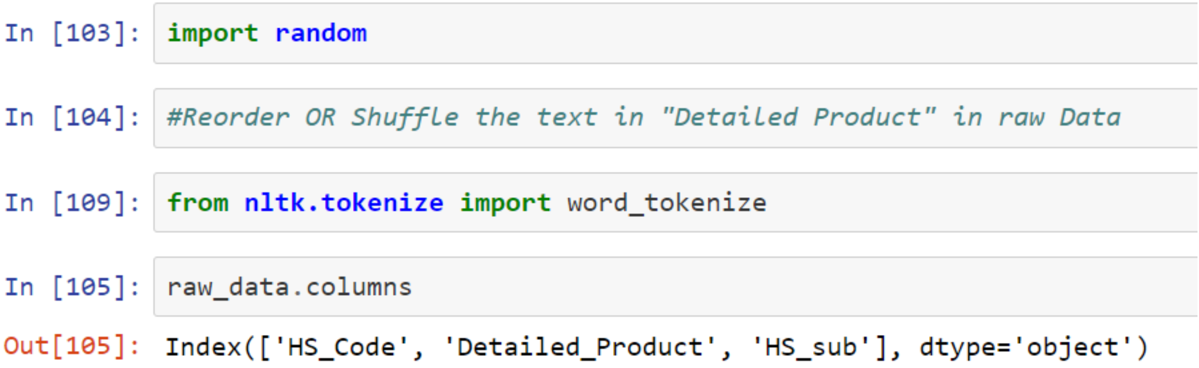
Convert the whole df to str type as well as lower case :

```
df = df.apply(lambda x: x.astype(str).str.lower())
```

Convert the df column to str type and lower case:

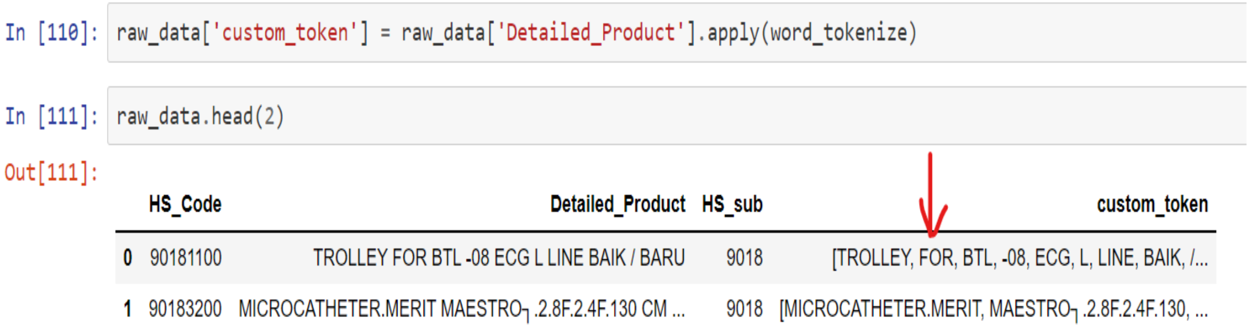
```
df['column'] = df['column'].apply(lambda x: x.astype(str).str.lower())
```

We are applying random shuffle on "Detailed_Product" column:



```
In [103]: import random

In [104]: #Reorder OR Shuffle the text in "Detailed Product" in raw Data

In [109]: from nltk.tokenize import word_tokenize

In [105]: raw_data.columns

Out[105]: Index(['HS_Code', 'Detailed_Product', 'HS_sub'], dtype='object')
```

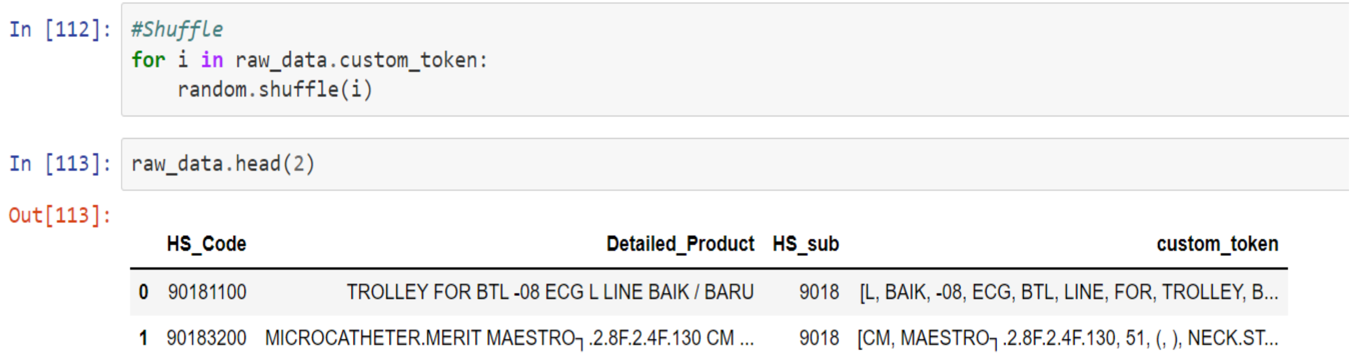
First we tokenize the text and save it in a new column called "custom_token"



```
In [110]: raw_data['custom_token'] = raw_data['Detailed_Product'].apply(word_tokenize)

In [111]: raw_data.head(2)

Out[111]:
```

	HS_Code	Detailed_Product	HS_sub	custom_token
0	90181100	TROLLEY FOR BTL -08 ECG L LINE BAIK / BARU	9018	[TROLLEY, FOR, BTL, -08, ECG, L, LINE, BAIK, /...
1	90183200	MICROCATHETER.MERIT MAESTRO┐.2.8F.2.4F.130 CM ...	9018	[MICROCATHETER.MERIT, MAESTRO┐.2.8F.2.4F.130, ...

Random shuffle : custom_token



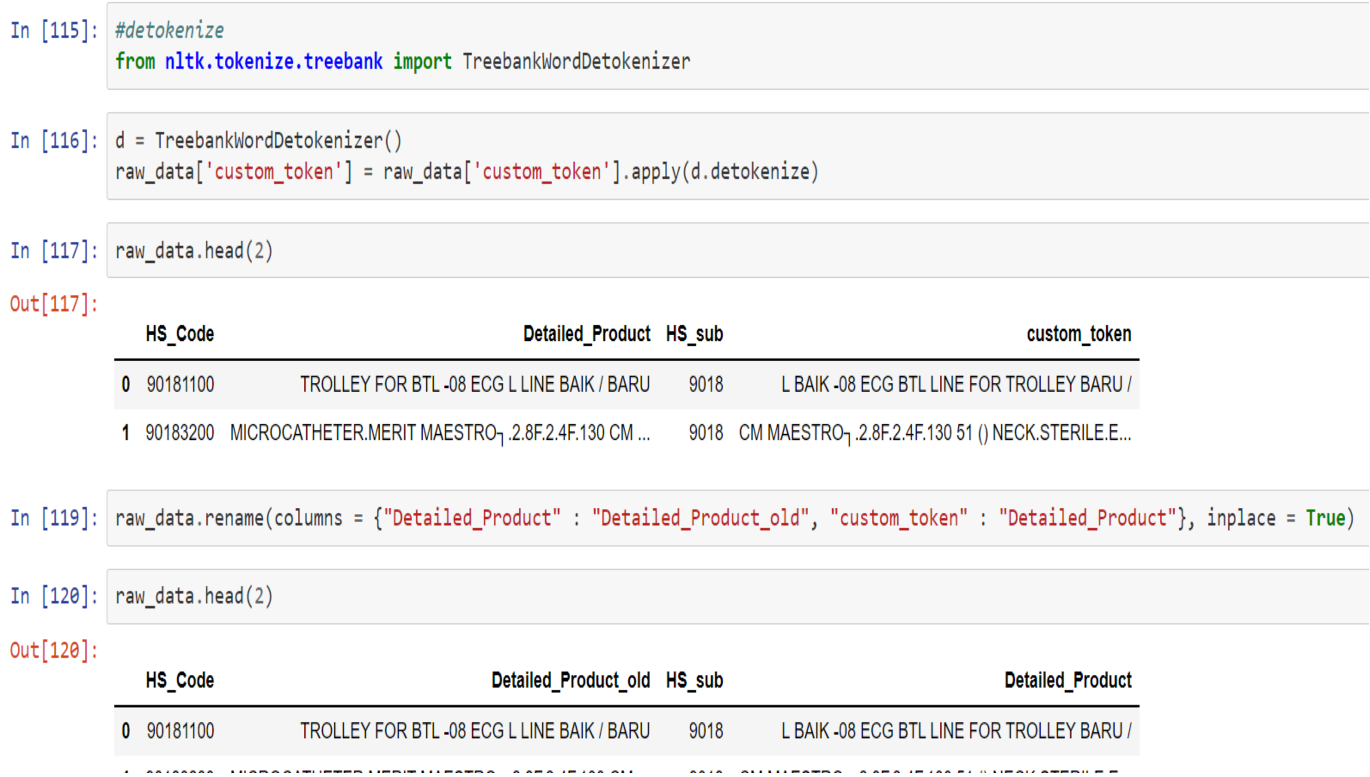
```
In [112]: #Shuffle
          for i in raw_data.custom_token:
              random.shuffle(i)
```

```
In [113]: raw_data.head(2)
```

Out[113]:

	HS_Code	Detailed_Product	HS_sub	custom_token
0	90181100	TROLLEY FOR BTL -08 ECG L LINE BAIK / BARU	9018	[L, BAIK, -08, ECG, BTL, LINE, FOR, TROLLEY, B...
1	90183200	MICROCATHETER.MERIT MAESTRO┐ .2.8F.2.4F.130 CM ...	9018	[CM, MAESTRO┐ .2.8F.2.4F.130, 51, (,), NECK.ST...

Then we de-Tokenize :



```
In [115]: #detokenize
          from nltk.tokenize.treebank import TreebankWordDetokenizer
```

```
In [116]: d = TreebankWordDetokenizer()
          raw_data['custom_token'] = raw_data['custom_token'].apply(d.detokenize)
```

```
In [117]: raw_data.head(2)
```

Out[117]:

	HS_Code	Detailed_Product	HS_sub	custom_token
0	90181100	TROLLEY FOR BTL -08 ECG L LINE BAIK / BARU	9018	L BAIK -08 ECG BTL LINE FOR TROLLEY BARU /
1	90183200	MICROCATHETER.MERIT MAESTRO┐ .2.8F.2.4F.130 CM ...	9018	CM MAESTRO┐ .2.8F.2.4F.130 51 () NECK.STERILE.E...

```
In [119]: raw_data.rename(columns = {"Detailed_Product" : "Detailed_Product_old", "custom_token" : "Detailed_Product"}, inplace = True)
```

```
In [120]: raw_data.head(2)
```

Out[120]:

	HS_Code	Detailed_Product_old	HS_sub	Detailed_Product
0	90181100	TROLLEY FOR BTL -08 ECG L LINE BAIK / BARU	9018	L BAIK -08 ECG BTL LINE FOR TROLLEY BARU /

De-Tokenize and Rename the columns , Now we are ready to apply functions like String match or any kind of Text Analysis on this shuffled text.

Thank you – Aisha Khalid