Filtering cpdata.csv to MergeFileCrop.csv

Filtering fertilizer.csv to MergerFileFert.csv

In [ ]:
```
#pandas is used for analyzing and cleaning the data
import pandas as pd
```

In [ ]:
```
# Reading the data from specific path

crop_data_path = '../Data-raw/cpdata.csv'
fertilizer_data_path = '../Data-raw/Fertilizer.csv'

crop = pd.read_csv(crop_data_path)
fert = pd.read_csv(fertilizer_data_path)
```

In [ ]:
```
crop.head()
```

Out[ ]:

	temperature	humidity	ph	rainfall	label
0	20.879744	82.002744	6.502985	202.935536	rice
1	21.770462	80.319644	7.038096	226.655537	rice
2	23.004459	82.320763	7.840207	263.964248	rice
3	26.491096	80.158363	6.980401	242.864034	rice
4	20.130175	81.604873	7.628473	262.717340	rice

In [ ]:
```
fert.head()
```

Out[ ]:

	Unnamed: 0	Crop	N	P	K	pH
0	0	Rice	80	40	40	5.5
1	1	Jowar(Sorghum)	80	40	40	5.5
2	2	Barley(JAV)	70	40	45	5.5
3	3	Maize	80	40	20	5.5
4	4	Ragi(naachnnii)	50	40	20	5.5

In [ ]:
```
# function for changing into Lower case Letters
def change_case(i):
    i = i.replace(" ", "")
    i = i.lower()
    return i
```

In [ ]:
```
#applying the above function to crop feature in fertilizer dataset and label featur
fert['Crop'] = fert['Crop'].apply(change_case)
crop['label'] = crop['label'].apply(change_case)
```

In [ ]:
```
#making changes in ferti data
fert['Crop'] = fert['Crop'].replace('mungbeans','mungbean')
fert['Crop'] = fert['Crop'].replace('lentils(masoordal)','lentil')
fert['Crop'] = fert['Crop'].replace('pigeonpeas(toordal)','pigeonpeas')
fert['Crop'] = fert['Crop'].replace('mothbean(matki)','mothbeans')
fert['Crop'] = fert['Crop'].replace('chickpeas(channa)','chickpea')
fert['Crop'] = fert['Crop'].replace('jowar(sorghum)','jowar')
```

```
fert['Crop'] = fert['Crop'].replace('barley(jav)','jav')
fert['Crop'] = fert['Crop'].replace('ragi(naachnnii)','ragi')
```

In [ ]:

```
crop.head()
```

Out[ ]:

	temperature	humidity	ph	rainfall	label
0	20.879744	82.002744	6.502985	202.935536	rice
1	21.770462	80.319644	7.038096	226.655537	rice
2	23.004459	82.320763	7.840207	263.964248	rice
3	26.491096	80.158363	6.980401	242.864034	rice
4	20.130175	81.604873	7.628473	262.717340	rice

In [ ]:

```
crop.tail()
```

Out[ ]:

	temperature	humidity	ph	rainfall	label
3095	25.287846	89.636679	6.765095	58.286977	watermelon
3096	26.638386	84.695469	6.189214	48.324286	watermelon
3097	25.331045	84.305338	6.904242	41.532187	watermelon
3098	26.897502	83.892415	6.463271	43.971937	watermelon
3099	26.986037	89.413849	6.260839	58.548767	watermelon

In [ ]:

```
fert.head()
```

Out[ ]:

	Unnamed: 0	Crop	N	P	K	pH
0	0	rice	80	40	40	5.5
1	1	jowar	80	40	40	5.5
2	2	jav	70	40	45	5.5
3	3	maize	80	40	20	5.5
4	4	ragi	50	40	20	5.5

In [ ]:

```
#display the unique values(without repeating) for labels in crop dataset:nearly 30
crop_labels = crop['label'].unique()
crop_labels
```

Out[ ]:

```
array(['rice', 'wheat', 'mungbean', 'tea', 'millet', 'maize', 'lentil',
       'jute', 'coffee', 'cotton', 'groundnut', 'peas', 'rubber',
       'sugarcane', 'tobacco', 'kidneybeans', 'mothbeans', 'coconut',
       'blackgram', 'adzukibeans', 'pigeonpeas', 'chickpea', 'banana',
       'grapes', 'apple', 'mango', 'muskmelon', 'orange', 'papaya',
       'pomegranate', 'watermelon'], dtype=object)
```

In [ ]:

```
del fert['Unnamed: 0']
```

In [ ]:

```
#These are the unique values of crop in fertilizer dataset: nearly 100 unique crop
fert_labels = fert['Crop'].unique()
fert_labels
```

Out[ ]:

```
array(['rice', 'jowar', 'jav', 'maize', 'ragi', 'chickpea',
       'frenchbeans(farasbi)', 'favabeans(papdi-val)', 'limabeans(pavta)',
       'clusterbeans(gavar)', 'soyabean', 'blackeyedbeans(chawli)',
       'kidneybeans', 'pigeonpeas', 'mothbeans', 'mungbean', 'greenpeas',
       'horsegram(kulthi)', 'blackgram', 'rapeseed(mohri)',
       'corianderseeds', 'mustardseeds', 'sesameseed', 'cuminseeds',
       'lentil', 'brinjal', 'beetroot', 'bittergourd', 'bottlegourd',
       'capsicum', 'cabbage', 'carrot', 'cauliflower', 'cucumber',
       'corianderleaves', 'curryleaves', 'drumstick-moringa', 'chili',
       'ladyfinger', 'mushroom', 'onion', 'potato', 'pumpkin', 'radish',
       'olive', 'sweetpotato', 'fenugreekleaf(methi)', 'spinach',
       'ridgegourd', 'gooseberry(amla)', 'jambun(syzygiumcumini)',
       'ziziphusmauritiana(bor)', 'garciniaindica(kokam)', 'tamarind',
       'tapioca(suran)', 'garlic', 'lemon', 'tomato', 'ashgourd',
       'pineapple', 'pomegranate', 'banana', 'mango', 'grapes',
       'jackfruit', 'guava', 'watermelon', 'muskmelon', 'apricot',
       'apple', 'chickoo', 'custardapple', 'dates', 'figs', 'orange',
       'papaya', 'aniseed', 'asafoetida', 'bayleaf', 'blackpepper',
       'cardamom', 'cinnamon', 'cloves', 'jaiphal(nutmeg)', 'ginger',
       'turmeric', 'cashewnuts', 'raisins', 'coconut', 'almondnut',
       'arecanut', 'pistachionut', 'lemongrass', 'cotton', 'jute',
       'coffee', 'sunflower'], dtype=object)
```

In [ ]:

```
#if the crops in ferti dataset are equal to crop dataset then that crop data will l
for i in fert_labels:
    print(crop[crop['label'] == i])
```

In [ ]:

```
#3100 labels present in crop
crop['label']
```

Out[ ]:

```
0             rice
1             rice
2             rice
3             rice
4             rice
           ...    
3095    watermelon
3096    watermelon
3097    watermelon
3098    watermelon
3099    watermelon
Name: label, Length: 3100, dtype: object
```

In [ ]:

```
#if the labels in fertilizer data are in crop datasets, then that labels alone will
extract_labels = []
for i in fert_labels:
    if i in crop_labels:
        extract_labels.append(i)
print(extract_labels)
```

```
['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas', 'mothbeans', 'mungbea
n', 'blackgram', 'lentil', 'pomegranate', 'banana', 'mango', 'grapes', 'watermelo
n', 'muskmelon', 'apple', 'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffe
e']
```

In [ ]:

```
#column names of crop dataset will form as new dataframe-(new_crop)
#column names of ferti dataset will form as new dataframe-(new_fert)
new_crop = pd.DataFrame(columns = crop.columns)
new_fert = pd.DataFrame(columns = fert.columns)
print(new_crop.head(5))
print(new_fert.head(5))
```

```
Empty DataFrame
Columns: [temperature, humidity, ph, rainfall, label]
Index: []
Empty DataFrame
Columns: [Crop, N, P, K, pH]
Index: []
```

In [ ]:

```
#The data in extracted labels should match to previous crop labels, those rows alor
for label in extract_labels:
    new_crop = new_crop.append(crop[crop['label'] == label])
```

In [ ]:

```
#The data in extracted labels should match to fertilizer labels, those rows in fert
for label in extract_labels:
    new_fert = new_fert.append(fert[fert['Crop'] == label].iloc[0])
```

In [ ]:

```
#After extracting the common labels in both datasets, There are 2200 rows in new_cr
new_crop
```

Out[ ]:

	temperature	humidity	ph	rainfall	label
0	20.879744	82.002744	6.502985	202.935536	rice
1	21.770462	80.319644	7.038096	226.655537	rice
2	23.004459	82.320763	7.840207	263.964248	rice
3	26.491096	80.158363	6.980401	242.864034	rice
4	20.130175	81.604873	7.628473	262.71734	rice
...	...	...	...	...	...
895	26.774637	66.413269	6.780064	177.774507	coffee
896	27.417112	56.636362	6.086922	127.92461	coffee
897	24.131797	67.225123	6.362608	173.322839	coffee
898	26.272418	52.127394	6.758793	127.175293	coffee
899	23.603016	60.396475	6.779833	140.937041	coffee

2200 rows × 5 columns

In [ ]:

```
#here 1843 are reduced to 22 rows
new_fert
```

Out[ ]:

	Crop	N	P	K	pH
0	rice	80	40	40	5.5
3	maize	80	40	20	5.5
5	chickpea	40	60	80	5.5
12	kidneybeans	20	60	20	5.5
13	pigeonpeas	20	60	20	5.5
14	mothbeans	20	40	20	5.5
15	mungbean	20	40	20	5.5
18	blackgram	40	60	20	5.0
24	lentil	20	60	20	5.5
60	pomegranate	20	10	40	5.5
61	banana	100	75	50	6.5
62	mango	20	20	30	5.0
63	grapes	20	125	200	4.0
66	watermelon	100	10	50	5.5
67	muskmelon	100	10	50	5.5
69	apple	20	125	200	6.5
74	orange	20	10	10	4.0
75	papaya	50	50	50	6.0
88	coconut	20	10	30	5.0
93	cotton	120	40	20	5.5
94	jute	80	40	40	5.5
95	coffee	100	20	30	5.5

In [ ]:

```
new_crop.to_csv('../Data-raw/MergeFileCrop.csv')
new_fert.to_csv('../Data-raw/FertilizerData.csv')
```