Module 1: Data validation and pre-processing

In [1]:
```
import numpy as n
import pandas as p
from sklearn.preprocessing import LabelEncoder
```

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
df = p.read_csv('crop.csv') #load the dataset as dataframe
```

In [4]:
```
df.columns #Returns columns of dataframe
```

Out[4]:
```
Index(['nitrogen', 'phosphorus', 'potassium', 'temperature', 'humidity', 'ph',
       'rainfall', 'label'],
      dtype='object')
```

In [5]:
```
df.info() #Returns basic description of dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   nitrogen     2200 non-null   int64  
 1   phosphorus   2200 non-null   int64  
 2   potassium    2200 non-null   int64  
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   object 
dtypes: float64(4), int64(3), object(1)
memory usage: 137.6+ KB
```

In [6]:
```
df.head(3) #Returns first 3 rows of data
```

Out[6]:

	nitrogen	phosphorus	potassium	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice

In [7]:
```
df.shape #dimension of dataframe
```

Out[7]:
```
(2200, 8)
```

In [8]:
```
df.isnull() #returns True if value is NULL else returns False
```

Out[8]:

	nitrogen	phosphorus	potassium	temperature	humidity	ph	rainfall	label
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
2195	False	False	False	False	False	False	False	False
2196	False	False	False	False	False	False	False	False
2197	False	False	False	False	False	False	False	False
2198	False	False	False	False	False	False	False	False
2199	False	False	False	False	False	False	False	False

2200 rows × 8 columns

In [9]:
```
df.isnull().sum() #Return sum of missing values in each column
```

Out[9]:
```
nitrogen       0
phosphorus     0
potassium      0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64
```

In [10]:
```
df.describe() #Returns numerical description
```

Out[10]:

	nitrogen	phosphorus	potassium	temperature	humidity	ph	rainfall
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.463655
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.958389
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.211267
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.551686
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.867624
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.267508
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.560117

In [11]:
```
sum(df.duplicated()) #Returns sum of duplicate data
```

Out[11]:
```
0
```

In [12]:
```
df.nitrogen.unique() #Returns unique values of nitrogen
```

Out[12]:
```
array([ 90,  85,  60,  74,  78,  69,  94,  89,  68,  91,  93,  77,  88,
        76,  67,  83,  98,  66,  97,  84,  73,  92,  95,  99,  63,  62,
        64,  82,  79,  65,  75,  71,  72,  70,  86,  61,  81,  80, 100,
        87,  96,  40,  23,  39,  22,  36,  32,  58,  59,  42,  28,  43,
        27,  50,  25,  31,  26,  54,  57,  49,  46,  38,  35,  52,  44,
        24,  29,  20,  56,  37,  51,  41,  34,  30,  33,  47,  53,  45,
        48,  13,   2,  17,  12,   6,  10,  19,  11,  18,  21,  16,   9,
         1,   7,   8,   0,   3,   4,   5,  14,  15,  55, 105, 108, 118,
       101, 106, 109, 117, 114, 110, 112, 111, 102, 116, 119, 107, 104,
       103, 120, 113, 115, 133, 136, 126, 121, 129, 122, 140, 131, 135,
       123, 125, 139, 132, 127, 130, 134], dtype=int64)
```

In [13]:
```
df['label'].unique() #Returns unique labels
```

Out[13]:
```
array(['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas',
       'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate',
       'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple',
       'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffee'],
      dtype=object)
```

In [14]:
```
df.potassium.sort_values().unique() #Returns unique values of potassium after sortin
```

Out[14]:
```
array([  5,   6,   7,   8,   9,  10,  11,  12,  13,  14,  15,  16,  17,
        18,  19,  20,  21,  22,  23,  24,  25,  26,  27,  28,  29,  30,
        31,  32,  33,  34,  35,  36,  37,  38,  39,  40,  41,  42,  43,
        44,  45,  46,  47,  48,  49,  50,  51,  52,  53,  54,  55,  75,
        76,  77,  78,  79,  80,  81,  82,  83,  84,  85, 195, 196, 197,
       198, 199, 200, 201, 202, 203, 204, 205], dtype=int64)
```

In [15]:
```
df['label'].value_counts() #Returns number of instances of each unique label(crop ty
```

Out[15]:
```
mungbean       100
jute           100
coffee         100
orange         100
rice           100
chickpea       100
apple          100
muskmelon      100
maize          100
coconut        100
pomegranate    100
cotton         100
kidneybeans    100
grapes         100
lentil         100
banana         100
papaya         100
blackgram      100
watermelon     100
mothbeans      100
mango          100
pigeonpeas     100
Name: label, dtype: int64
```

In [16]:
```
df.corr() #Returns pairwise correlation of the columns
```

Out[16]:

	nitrogen	phosphorus	potassium	temperature	humidity	ph	rainfall
nitrogen	1.000000	-0.231460	-0.140512	0.026504	0.190688	0.096683	0.059020
phosphorus	-0.231460	1.000000	0.736232	-0.127541	-0.118734	-0.138019	-0.063839

	nitrogen	phosphorus	potassium	temperature	humidity	ph	rainfall
potassium	-0.140512	0.736232	1.000000	-0.160387	0.190859	-0.169503	-0.053461
temperature	0.026504	-0.127541	-0.160387	1.000000	0.205320	-0.017795	-0.030084
humidity	0.190688	-0.118734	0.190859	0.205320	1.000000	-0.008483	0.094423
ph	0.096683	-0.138019	-0.169503	-0.017795	-0.008483	1.000000	-0.109069
rainfall	0.059020	-0.063839	-0.053461	-0.030084	0.094423	-0.109069	1.000000

In [17]:
```
col_to_be_encoded = ['label']
le = LabelEncoder() #method to encode and set values between 0 and k-1 for k distinc
for i in col_to_be_encoded:
    df[i] = le.fit_transform(df[i]).astype(int) #Returns encoded labels as int datat
```

In [18]:
```
df['label'].unique()
```

Out[18]:
```
array([20, 11,  3,  9, 18, 13, 14,  2, 10, 19,  1, 12,  7, 21, 15,  0, 16,
       17,  4,  6,  8,  5])
```

In [19]:
```
df.tail(3) #Returns last 3 tuples of data
```

Out[19]:

	nitrogen	phosphorus	potassium	temperature	humidity	ph	rainfall	label
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	5
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	5
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	5