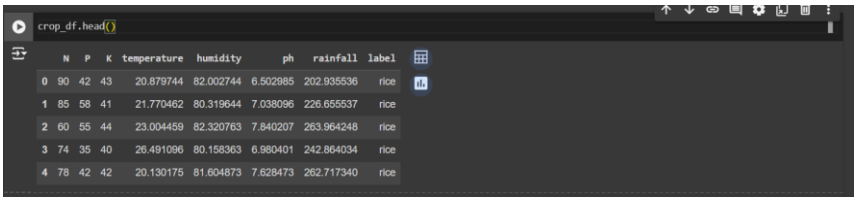
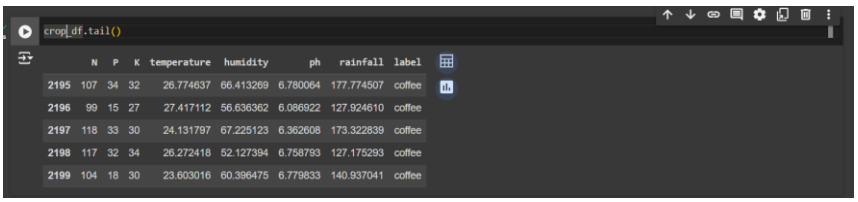
Data Collection and Preprocessing Phase

Date	15 July 2024
Team ID	739989
Project Title	Crop Prediction using machine learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	#Structure of the data:  crop_df.head() N P K 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 3 74 35 40 26.491096 80.158363 6.980401 242.864034 rice 4 78 42 42 20.130175 81.604873 7.628473 262.717340 rice  crop_df.tail() N P K temperature humidity ph rainfall label 2195 107 34 32 26.774637 66.413269 6.780064 177.774507 coffee 2196 99 15 27 27.417112 56.636362 6.086922 127.924610 coffee 2197 118 33 30 24.131797 67.225123 6.362608 173.322839 coffee 2198 117 32 34 26.272418 52.127394 6.758793 127.175293 coffee 2199 104 18 30 23.603016 60.396475 6.779833 140.937041 coffee crop_df.shape (2200, 8) #Descriptive Statistical: Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.

<table>
<tr><td></td><td>

```
crop_df.describe()
```

	N	P	K	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

</td></tr>
<tr><td>Univariate Analysis</td><td>

Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

Univariate Analysis:

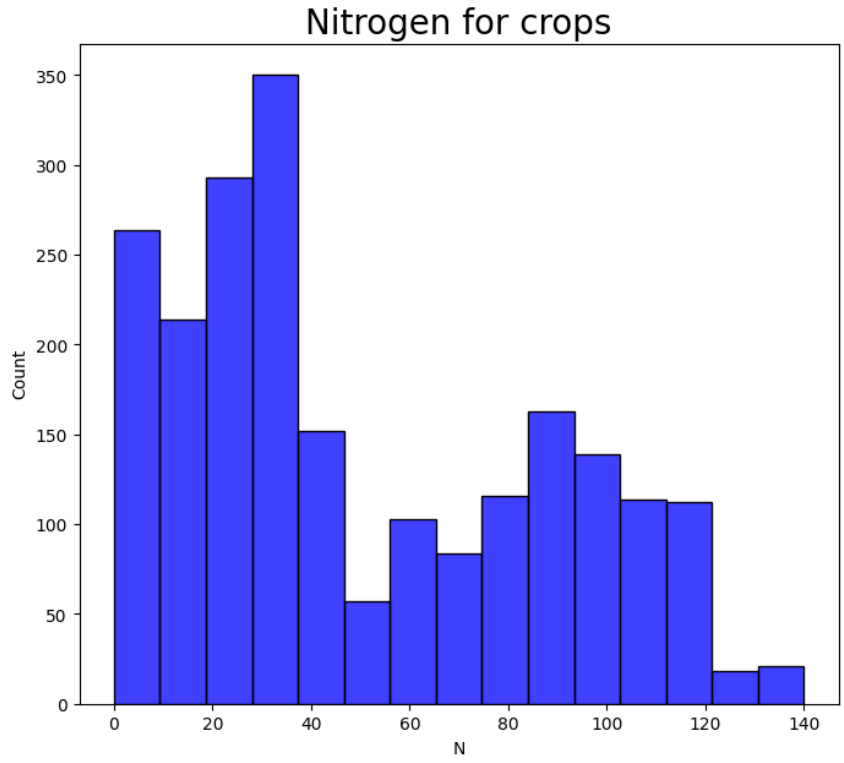
In simple words, univariate analysis is understanding the data with a single feature. We have displayed three different types of graphs and plots.

For simple visualizations we can use the matplotlib. pyplot library. Here the plt. figure() command is used to determine the size of the plot.

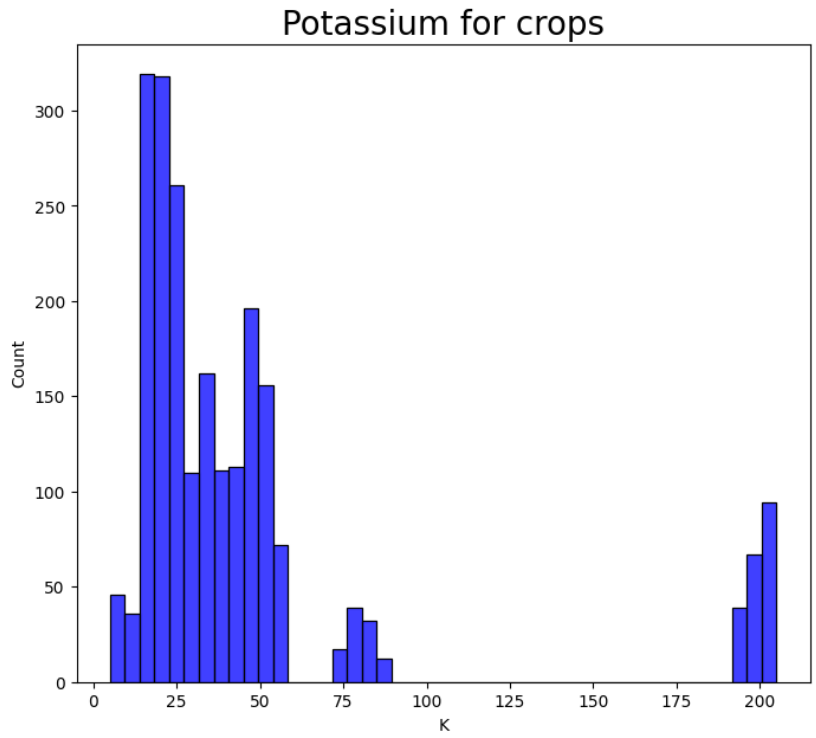
We have histogram for all features of the dataset which include phosphorus, humidity, temperature as well . The histogram shows the distribution of nitrogen fertilizers for crop.

```
#Nitrogen
plt.figure(figsize=(8,7))
sns.histplot(x='N',data=crop_df,color='b');
plt.title("Nitrogen for crops", {'fontsize':20});
```

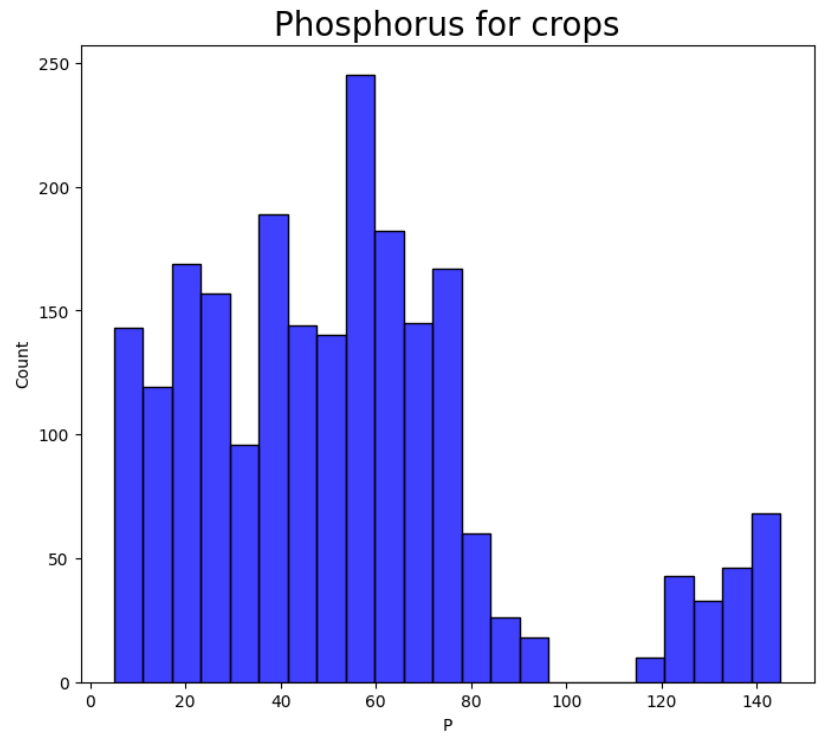
</td></tr>
</table>



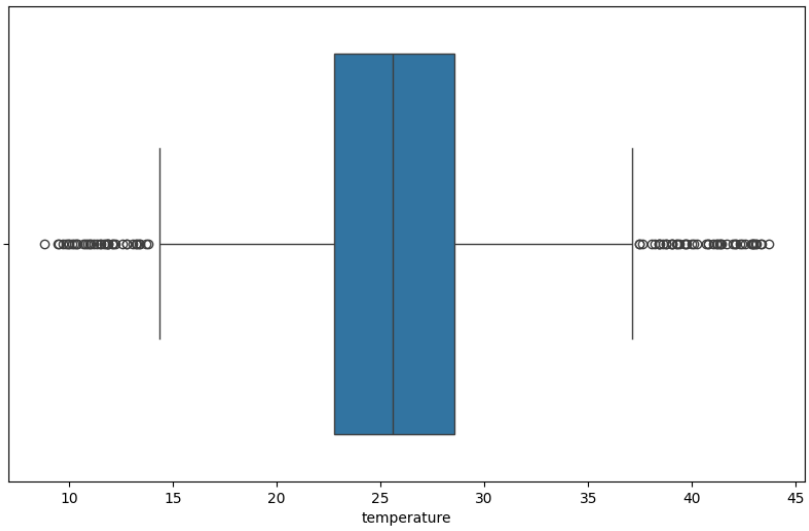

```
#Potassium
plt.figure(figsize=(8,7))
sns.histplot(x='K',data=crop_df,color='b');
plt.title("Potassium for crops",{'fontsize':20});
```



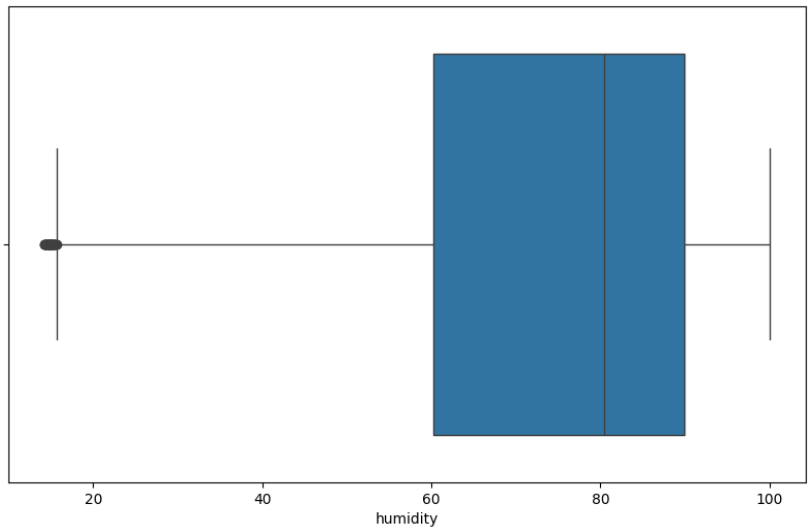
```
#Phosphorus
plt.figure(figsize=(8,7))
sns.histplot(x='P',data=crop_df,color='b');
plt.title("Phosphorus for crops",{'fontsize':20});
```




```
#Temperature
plt.figure(figsize=(10,6))
sns.boxplot(x=crop_df.temperature);
```

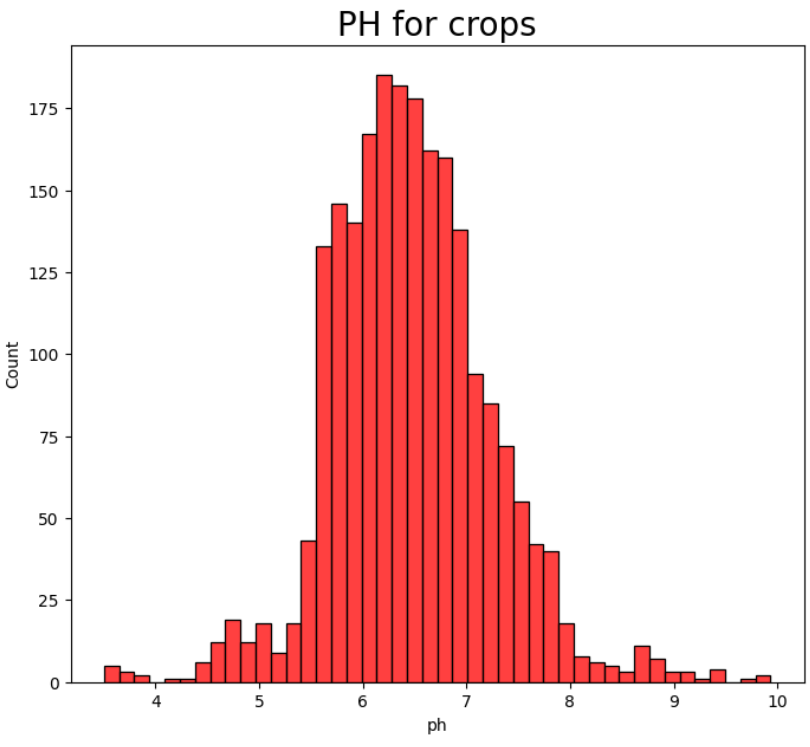



```
#Humidity
plt.figure(figsize=(10,6))
sns.boxplot(x=crop_df.humidity);
```

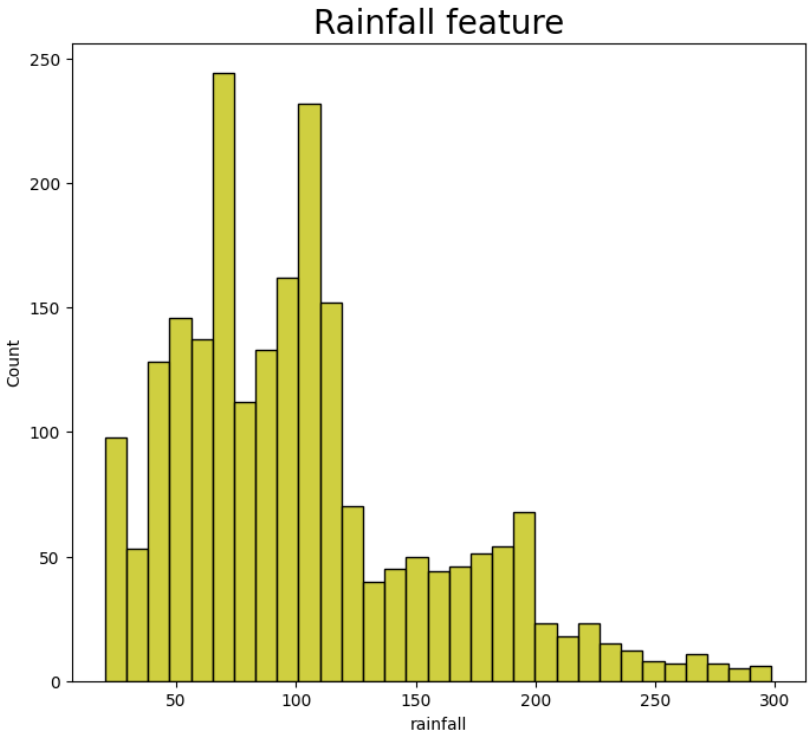


Here we have plotted Boxplot using seaborn library. These boxplots can be plotted without using any external library. We have plotted the boxplot using the inbuilt plot function in python.

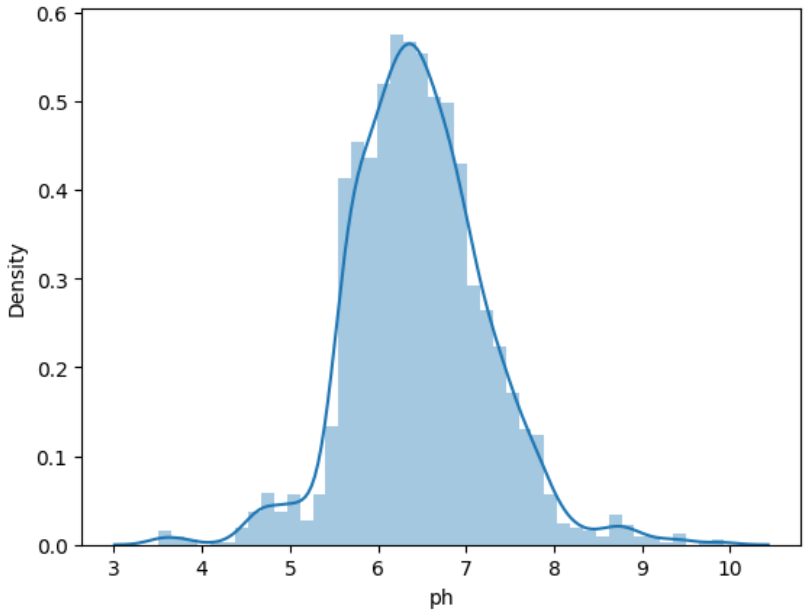
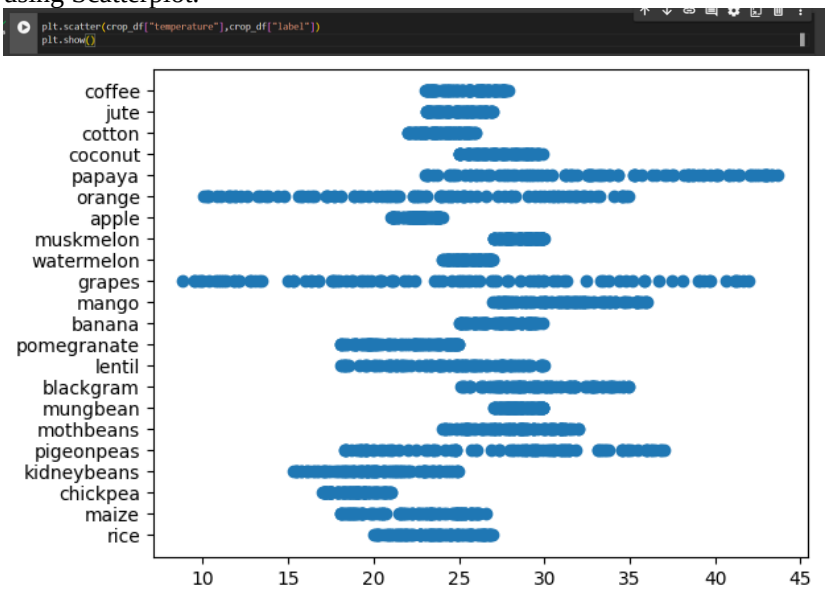
```
#ph
plt.figure(figsize=(8,7))
sns.histplot(x='ph',data=crop_df,color='r')
plt.title("PH for crops",{"fontsize":20});
```

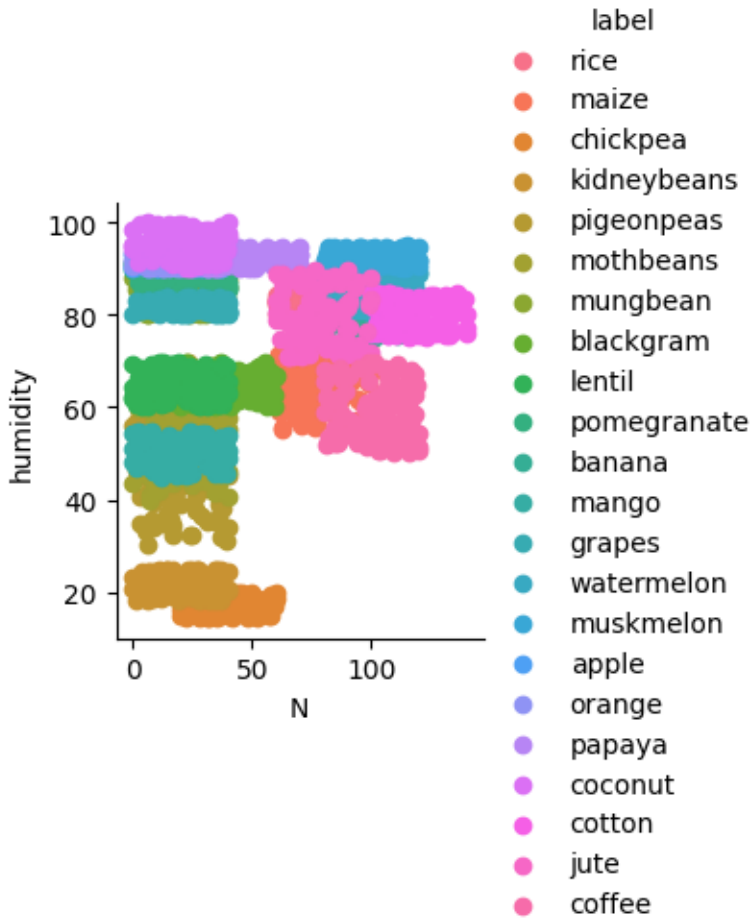
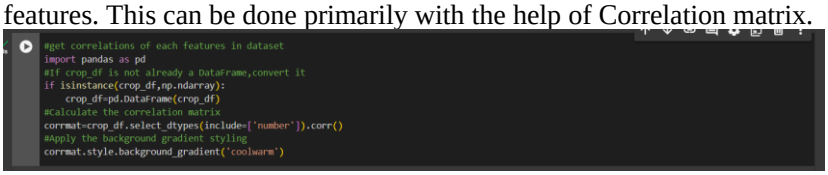



```
#Rainfall
plt.figure(figsize=(8,7))
sns.histplot(x='rainfall',data=crop_df,color='y');
plt.title("Rainfall feature",{"fontsize":20});
```



Here we have plotted distribution plot using seaborn library. These plots can be plotted without using any external library. A Distplot or distribution plot, depicts the variation in the data distribution. Seaborn Distplot represents the overall distribution of continuous data variables.

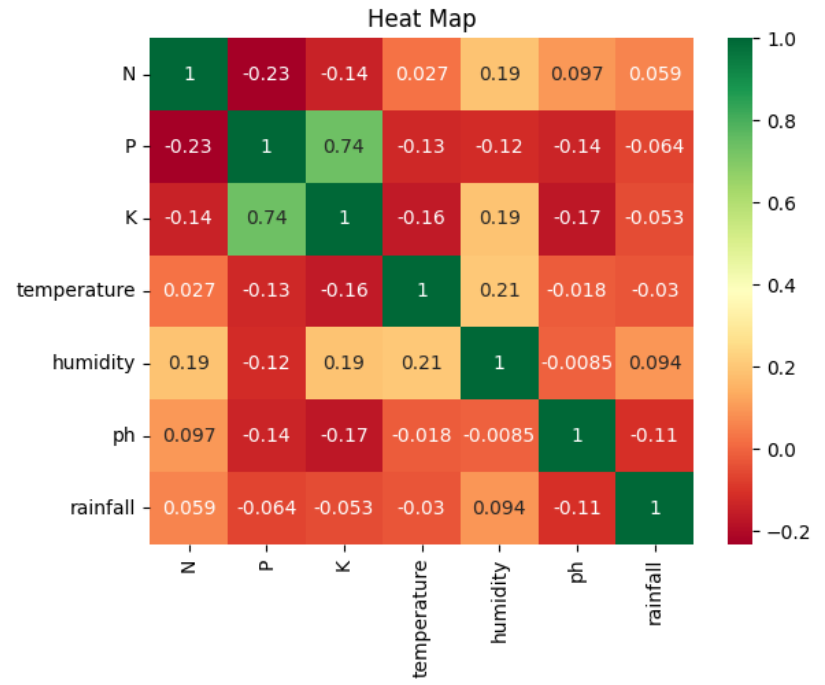
<table>
<tr>
<td></td>
<td> 
</td>
</tr>
<tr>
<td>Bivariate Analysis</td>
<td>Bivariate Analysis:
To find the relation between two features we use bivariate analysis. Here we are visualising the relationship between predicted crop and temperature.
#Scatter plot:
Scatterplot can be used with several semantic groupings which can help to understand well in a graph. They can plot two-dimensional graphics that can be enhanced by mapping up to three additional variables while using the semantics of hue, size, and style parameters. All the parameter control visual semantic which are used to identify the different subsets. Using redundant semantics can be helpful for making graphics more accessible. We have depicted the relationship between temperature and predicted crop using Scatterplot.
 
</td>
</tr>
</table>

	#FacetGrid: FacetGrid class helps in visualizing distribution of one variable as well as the relationship between multiple variables separately within subsets of your dataset using multiple panels. 
Multivariate Analysis	Multivariate Analysis: Multivariate analysis is a statistical technique used to analyse data that involves more than two variables. It aims to understand the relationships between multiple variables in a dataset by examining how they are related to each other and how they contribute to a particular outcome or phenomenon. In multivariate analysis we try to find the relation between multiple features. This can be done primarily with the help of Correlation matrix. 

	N	P	K	temperature	humidity	ph	rainfall
N	1.000000	-0.231460	-0.140512	0.026504	0.190688	0.096683	0.059020
P	-0.231460	1.000000	0.736232	-0.127541	-0.118734	-0.138019	-0.063839
K	-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

For multivariate analysis we will also plot a Heatmap
This code creates a heatmap that shows how much each column in a given data frame is related to each other column. It does this by first creating a new data frame that has all the columns of the original data frame except for 'Label' column. It then calculates the correlation between all the remaining columns and creates a matrix that shows these correlations. Finally, it generates a heatmap of this matrix using a library called Seaborn.

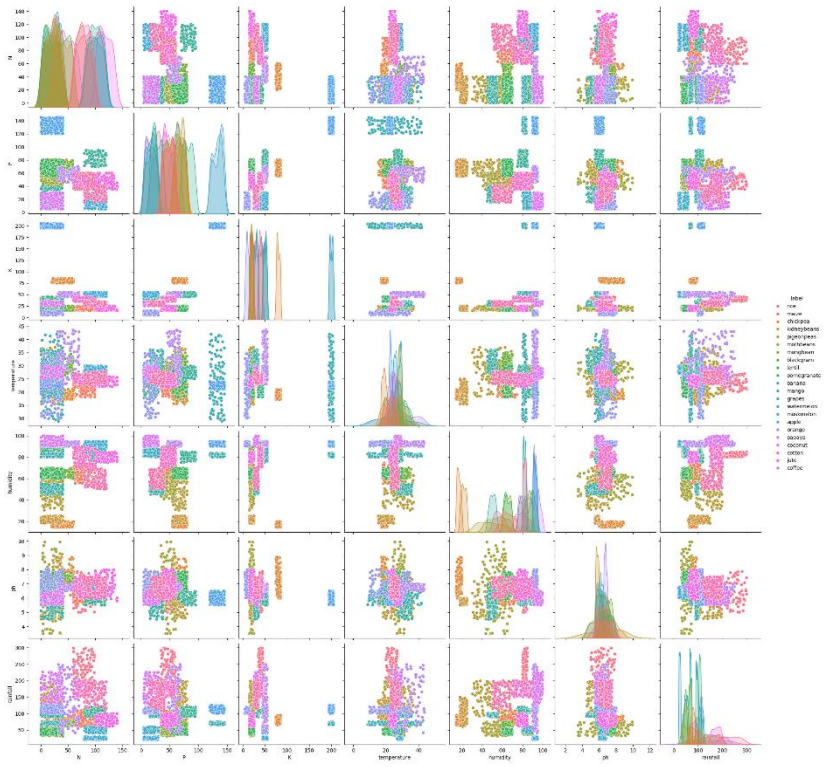
```
sns.heatmap(corrmat, annot=True, cmap="RdYlGn")
plt.title("Heat Map")
plt.show()
```



We also have plot seaborn.pairplot() :
To plot multiple pairwise bivariate distributions in a dataset, you can use the .pairplot() function.

```
sns.pairplot(crop_df,hue="label",size=3)
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:2100: UserWarning: The `size` parameter has been renamed to `height`; please update
  warnings.warn(msg, UserWarning)
<seaborn.axisgrid.PairGrid at 0x7c094814c610>
```



Outliers and Anomalies	There is no Outliers in our project.

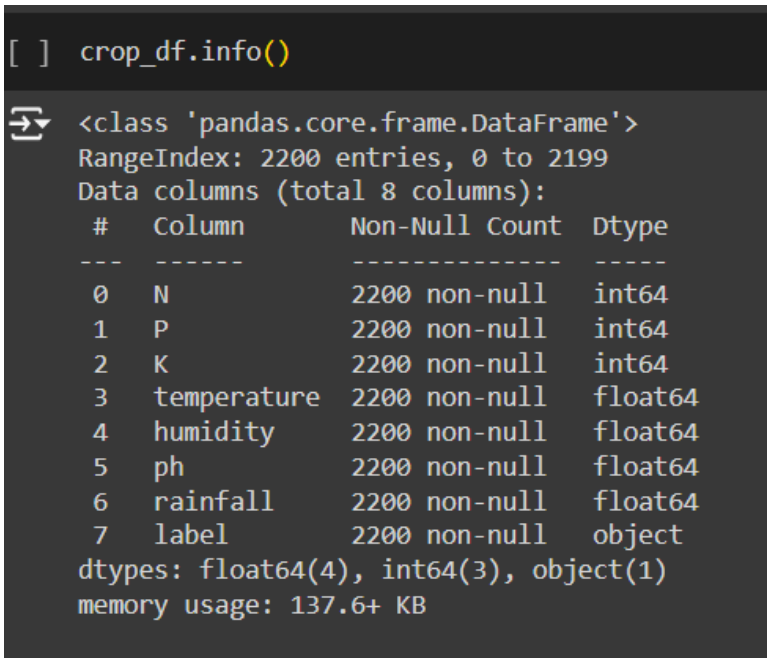
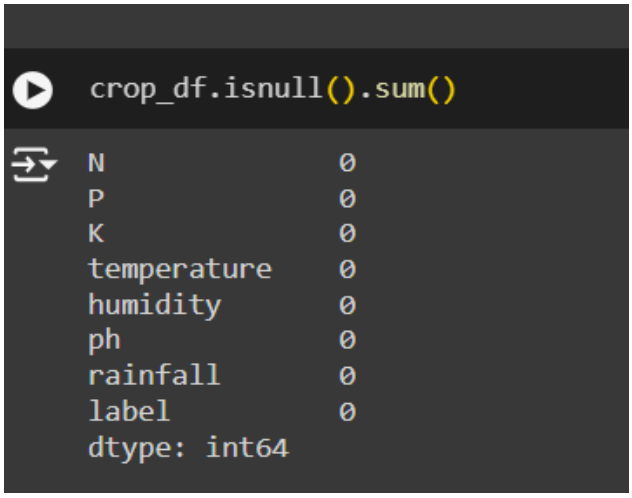
Data Preprocessing Code Screenshots

Loading Data	#Loading the data

```
crop_df=pd.read_csv("/content/Crop_recommendation.csv")
crop_df
```

	N	P	K	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
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice
...	...	...	...	...	...	...	...	...
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

2200 rows × 8 columns

Handling Missing Data	   For checking the null values . isnull() function is used. To sum those null values we use. sum() function. From the below image we found that there are no null values present in our dataset. So we can skip handling the missing values step.
Data Transformation	----------

Feature Engineering	---------
Save Processed Data	----------