In [1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from IPython import get_ipython
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```python
data = pd.read_csv('crop.csv')
```

In [3]:

```python
data.head()
```

Out[3]:

	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

In [4]:

```python
data.tail()
```

Out[4]:

	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

In [5]:

```
1  data.shape
```

Out[5]:

(2200, 8)

In [6]:

```
1  data.columns
```

Out[6]:

Index(['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'labe
l'], dtype='object')

In [7]:

```
1  data.duplicated().sum()
```

Out[7]:

0

In [8]:

```
1  data.isnull().sum()
```

Out[8]:

```
N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64
```

In [9]:

```
1  data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   N            2200 non-null   int64
 1   P            2200 non-null   int64
 2   K            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 [10]:

```
1  data.describe()
```

Out[10]:

	N	P	K	temperature	humidity	ph	r:
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.00
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.46
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.95
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.2'
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.55
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.86
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.26
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.56

In [11]:

```
1  data.nunique()
```

Out[11]:

```
N               137
P               117
K                73
temperature    2200
humidity       2200
ph             2200
rainfall       2200
label            22
dtype: int64
```

In [12]:

```
1  data['label'].unique()
```

Out[12]:

```
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 [13]:

```
1  data['label'].value_counts()
```

Out[13]:

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

In [14]:

```
1  crop_summary = pd.pivot_table(data,index=['label'],
2                                aggfunc='mean')
```
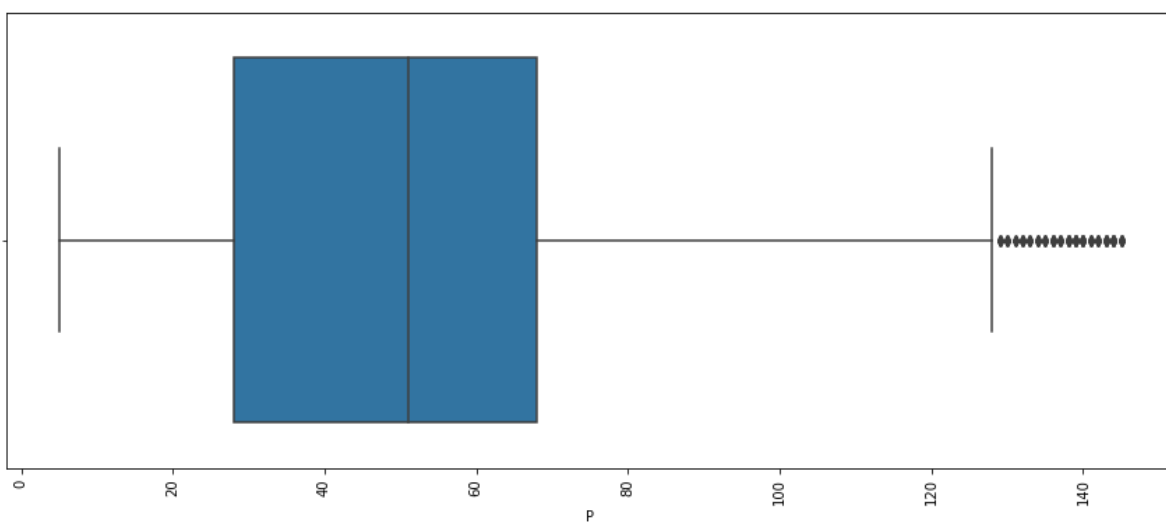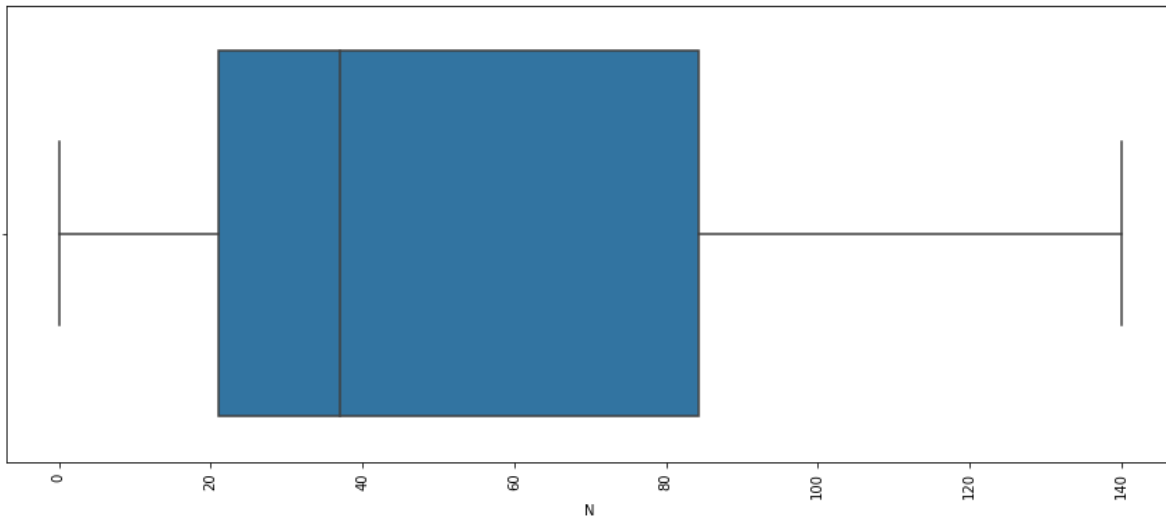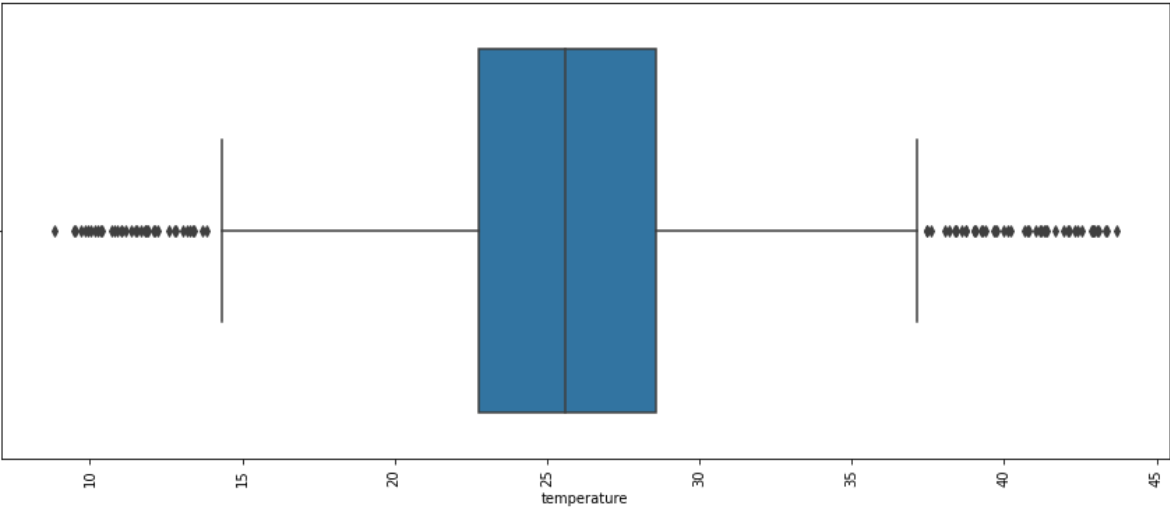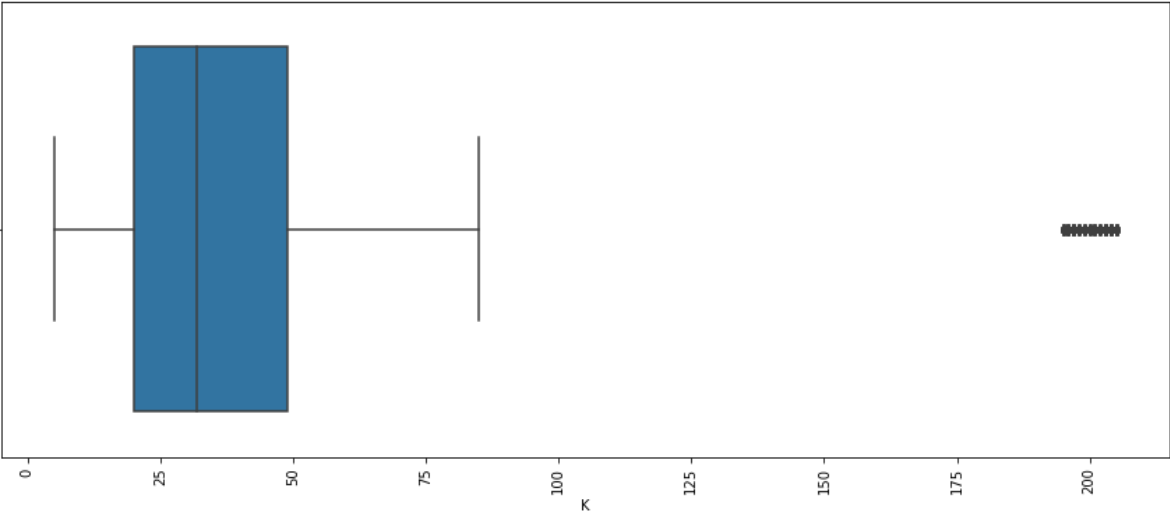
In [15]:

```
1  crop_summary
```

Out[15]:

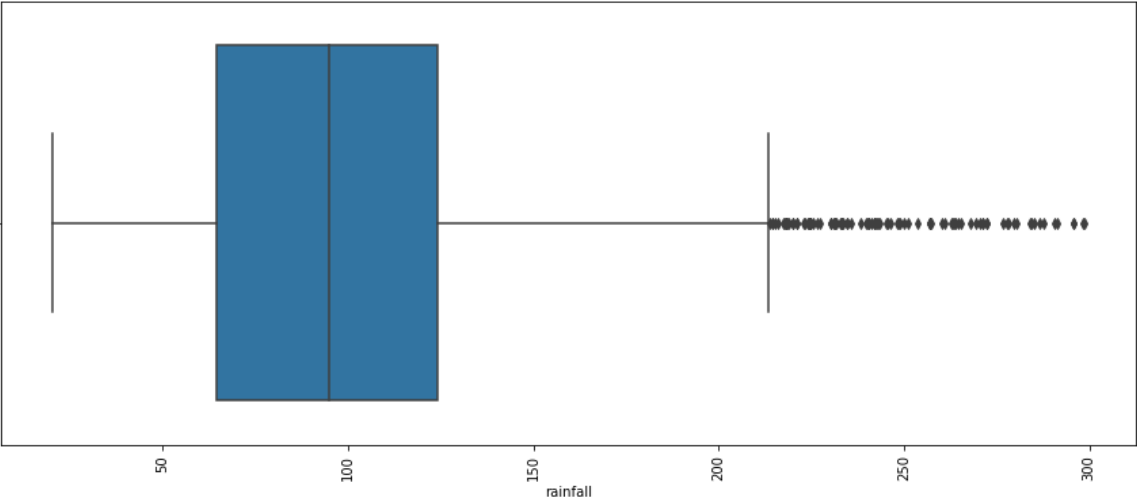| label	K	N	P	humidity	ph	rainfall	temperature
apple	199.89	20.80	134.22	92.333383	5.929663	112.654779	22.630942
banana	50.05	100.23	82.01	80.358123	5.983893	104.626980	27.376798
blackgram	19.24	40.02	67.47	65.118426	7.133952	67.884151	29.973340
chickpea	79.92	40.09	67.79	16.860439	7.336957	80.058977	18.872847
coconut	30.59	21.98	16.93	94.844272	5.976562	175.686646	27.409892
coffee	29.94	101.20	28.74	58.869846	6.790308	158.066295	25.540477
cotton	19.56	117.77	46.24	79.843474	6.912675	80.398043	23.988958
grapes	200.11	23.18	132.53	81.875228	6.025937	69.611829	23.849575
jute	39.99	78.40	46.86	79.639864	6.732778	174.792798	24.958376
kidneybeans	20.05	20.75	67.54	21.605357	5.749411	105.919778	20.115085
lentil	19.41	18.77	68.36	64.804785	6.927932	45.680454	24.509052
maize	19.79	77.76	48.44	65.092249	6.245190	84.766988	22.389204
mango	29.92	20.07	27.18	50.156573	5.766373	94.704515	31.208770
mothbeans	20.23	21.44	48.01	53.160418	6.831174	51.198487	28.194920
mungbean	19.87	20.99	47.28	85.499975	6.723957	48.403601	28.525775
muskmelon	50.08	100.32	17.72	92.342802	6.358805	24.689952	28.663066
orange	10.01	19.58	16.55	92.170209	7.016957	110.474969	22.765725
papaya	50.04	49.88	59.05	92.403388	6.741442	142.627839	33.723859
pigeonpeas	20.29	20.73	67.73	48.061633	5.794175	149.457564	27.741762
pomegranate	40.21	18.87	18.75	90.125504	6.429172	107.528442	21.837842
rice	39.87	79.89	47.58	82.272822	6.425471	236.181114	23.689332
watermelon	50.22	99.42	17.00	85.160375	6.495778	50.786219	25.591767

In [18]:

```python
data1 = data[['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall']]
for i in data1.columns:
            plt.figure(figsize=(15,6))
            sns.boxplot(data1[i])
            plt.xticks(rotation = 90)
            plt.show()
```
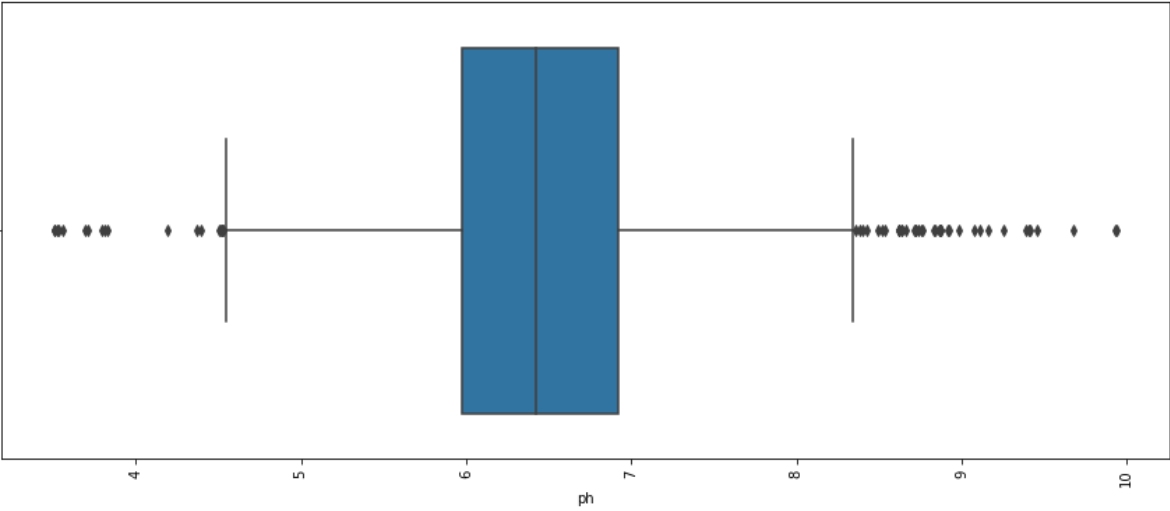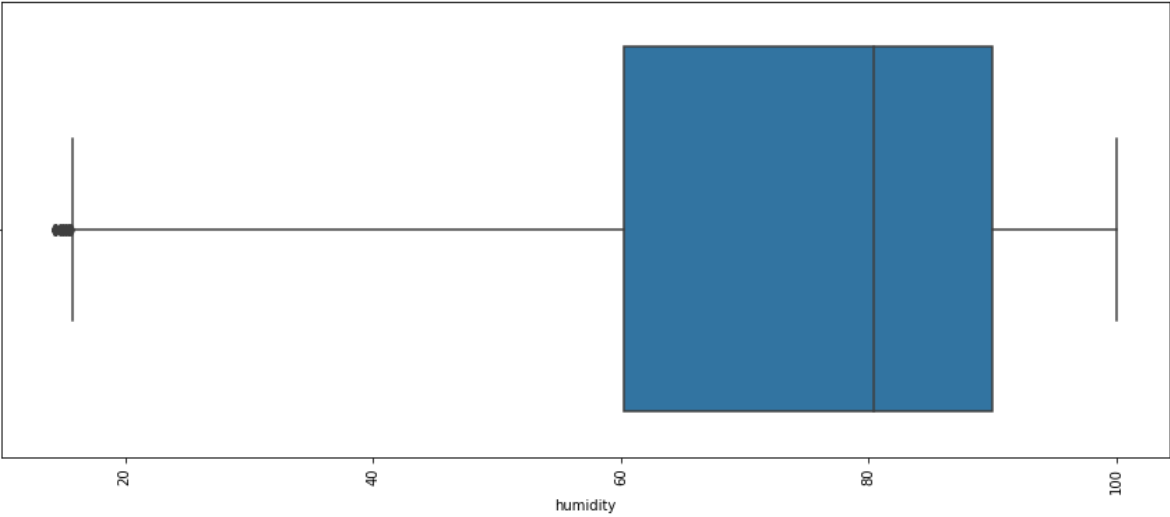
In [31]:

```
1  crop_summary_new = crop_summary.reset_index()
```

In [32]:

```
1  crop_summary_new
```

Out[32]:

	label	K	N	P	humidity	ph	rainfall	temperature
0	apple	199.89	20.80	134.22	92.333383	5.929663	112.654779	22.630942
1	banana	50.05	100.23	82.01	80.358123	5.983893	104.626980	27.376798
2	blackgram	19.24	40.02	67.47	65.118426	7.133952	67.884151	29.973340
3	chickpea	79.92	40.09	67.79	16.860439	7.336957	80.058977	18.872847
4	coconut	30.59	21.98	16.93	94.844272	5.976562	175.686646	27.409892
5	coffee	29.94	101.20	28.74	58.869846	6.790308	158.066295	25.540477
6	cotton	19.56	117.77	46.24	79.843474	6.912675	80.398043	23.988958
7	grapes	200.11	23.18	132.53	81.875228	6.025937	69.611829	23.849575
8	jute	39.99	78.40	46.86	79.639864	6.732778	174.792798	24.958376
9	kidneybeans	20.05	20.75	67.54	21.605357	5.749411	105.919778	20.115085
10	lentil	19.41	18.77	68.36	64.804785	6.927932	45.680454	24.509052
11	maize	19.79	77.76	48.44	65.092249	6.245190	84.766988	22.389204
12	mango	29.92	20.07	27.18	50.156573	5.766373	94.704515	31.208770
13	mothbeans	20.23	21.44	48.01	53.160418	6.831174	51.198487	28.194920
14	mungbean	19.87	20.99	47.28	85.499975	6.723957	48.403601	28.525775
15	muskmelon	50.08	100.32	17.72	92.342802	6.358805	24.689952	28.663066
16	orange	10.01	19.58	16.55	92.170209	7.016957	110.474969	22.765725
17	papaya	50.04	49.88	59.05	92.403388	6.741442	142.627839	33.723859
18	pigeonpeas	20.29	20.73	67.73	48.061633	5.794175	149.457564	27.741762
19	pomegranate	40.21	18.87	18.75	90.125504	6.429172	107.528442	21.837842
20	rice	39.87	79.89	47.58	82.272822	6.425471	236.181114	23.689332
21	watermelon	50.22	99.42	17.00	85.160375	6.495778	50.786219	25.591767

In [33]:

```python
plt.figure(figsize=(15,6))
sns.barplot(y = 'N', x = 'label', data=crop_summary_new, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```
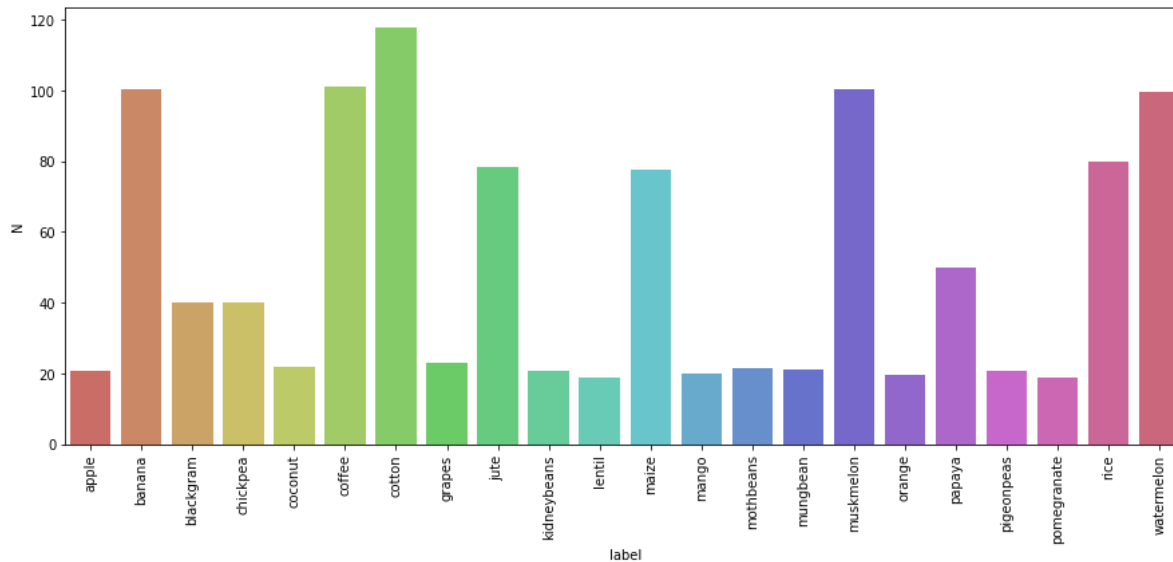

In [34]:

```python
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
```

In [35]:

```python
fig1 = px.bar(crop_summary_new, x='label', y='N')
fig1.show()
```

In [36]:

```python
plt.figure(figsize=(15,6))
sns.barplot(y = 'P', x = 'label', data=crop_summary_new, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```

In [37]:
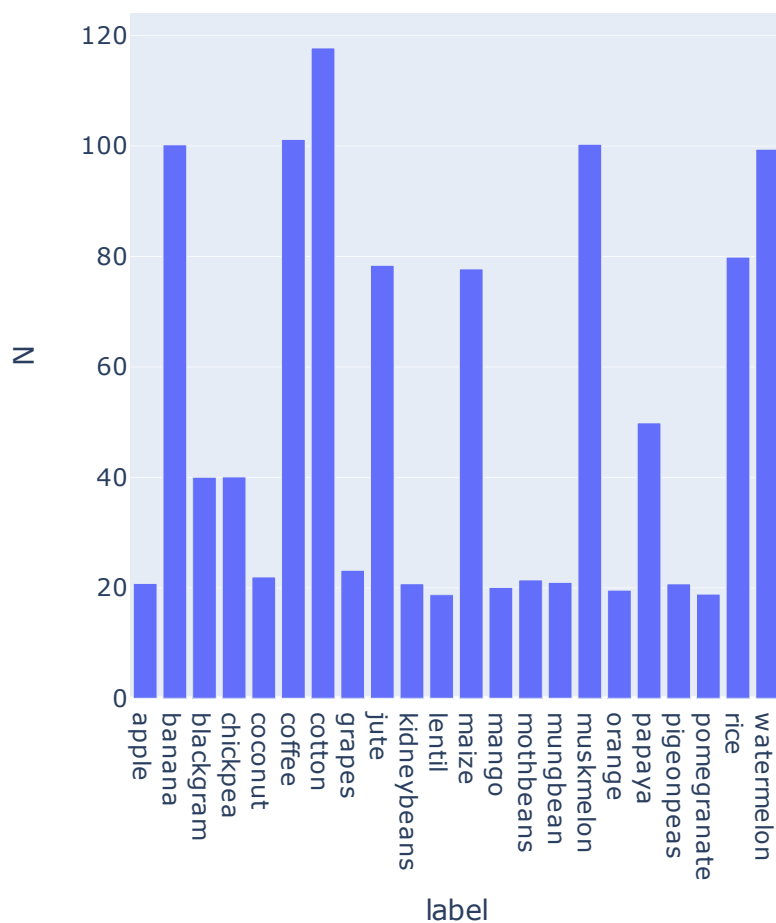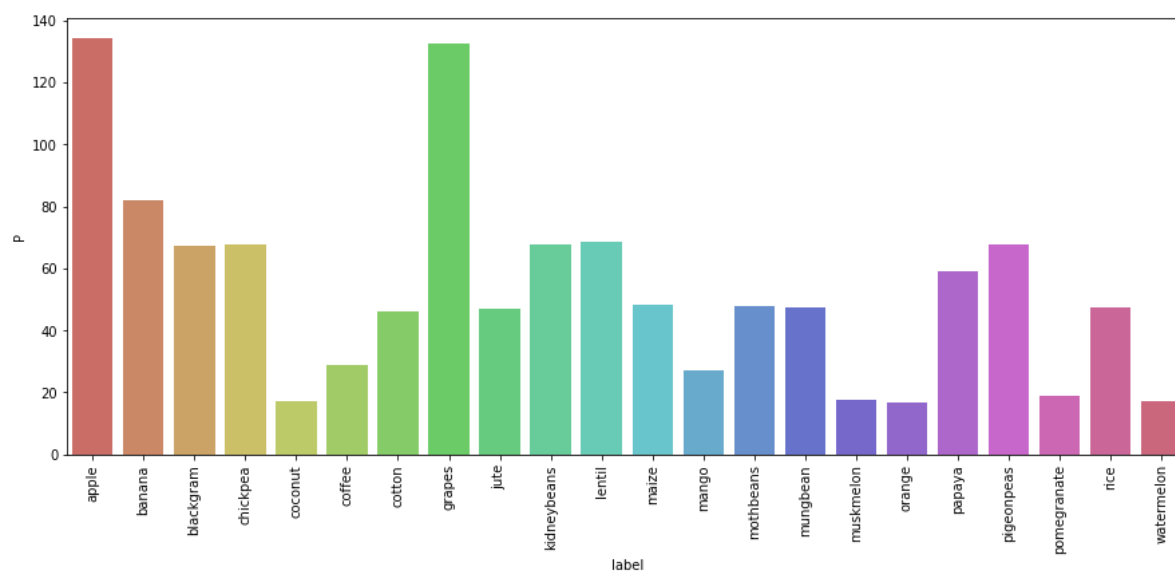
```python
fig2 = px.bar(crop_summary_new, x='label', y='P')
fig2.show()
```
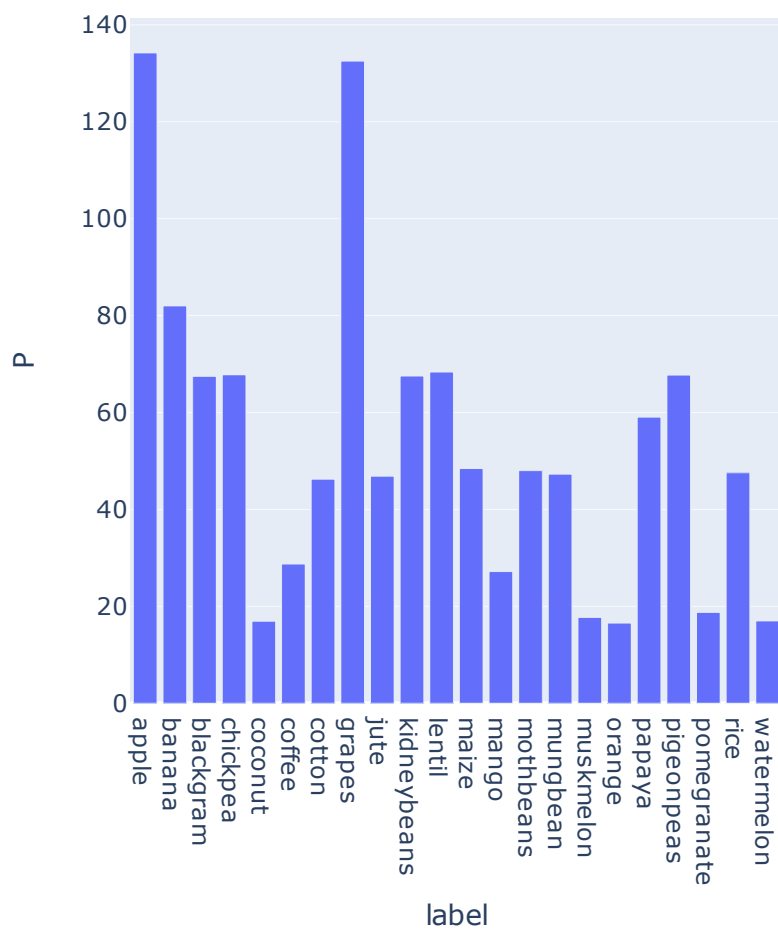
In [38]:

```python
plt.figure(figsize=(15,6))
sns.barplot(y = 'K', x = 'label', data=crop_summary_new, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```
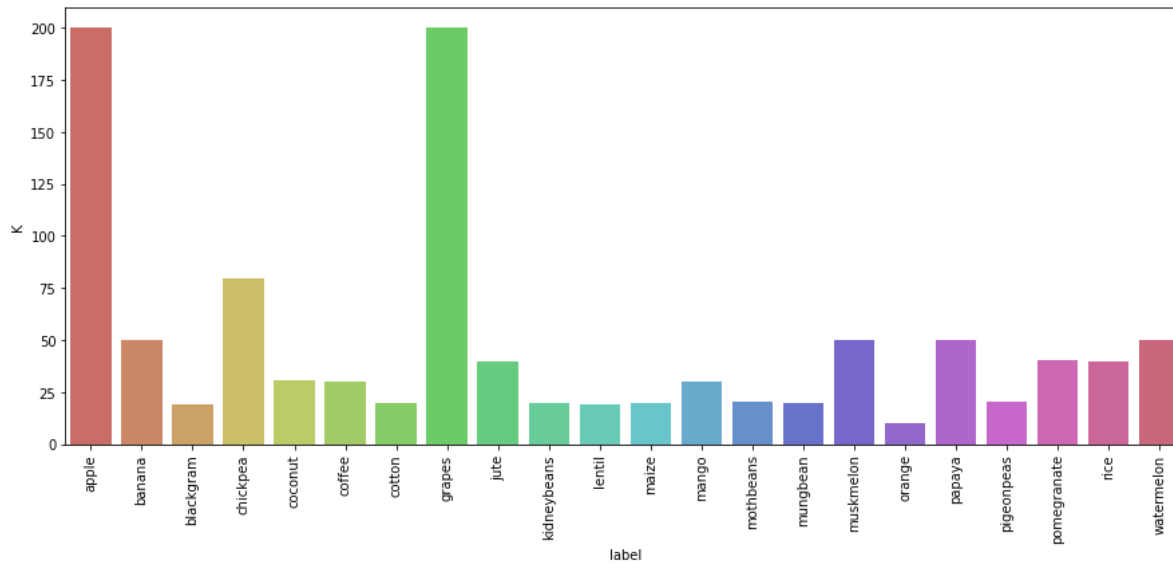
In [39]:

```python
fig3 = px.bar(crop_summary_new, x='label', y='K')
fig3.show()
```
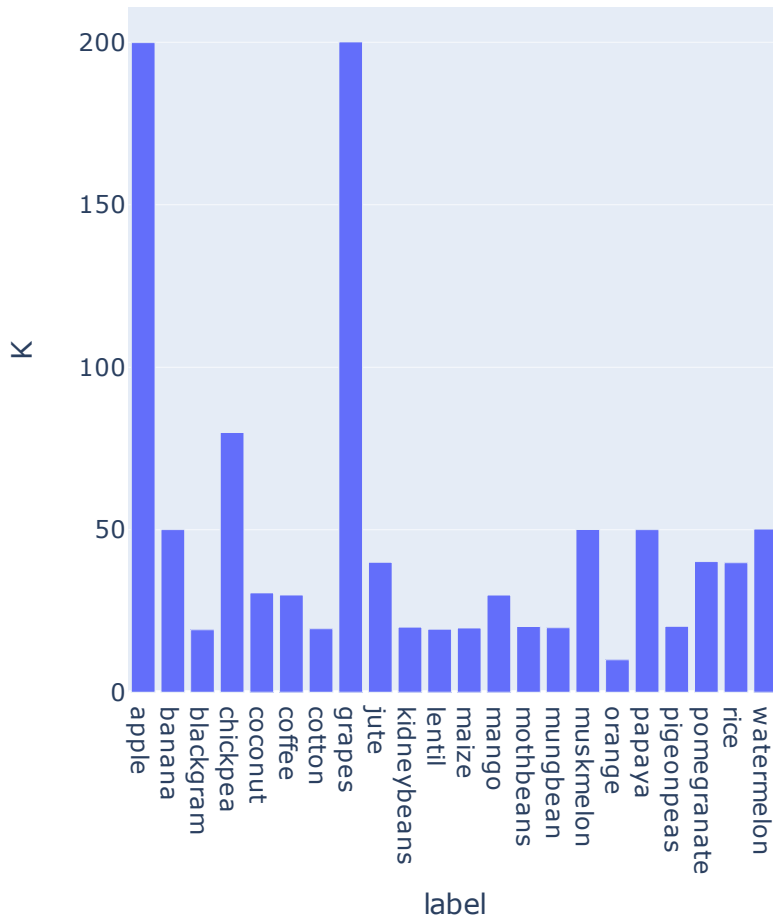

In [40]:

```python
colorarr = ['#0592D0','#Cd7f32', '#E97451', '#Bdb76b', '#954535', '#C2b280', '#8080
           '#32cd32','#39ff14','#00ff7f', '#008080', '#36454f', '#F88379', '#Ff4500
           '#Faf0e6', '#8c92ac', '#Dbd7d2','#A7a6ba', '#B38b6d']
```

In [41]:

```python
import random
```

In [42]:

```python
crop_summary_N = crop_summary.sort_values(by='N',
                                          ascending=False)

fig = make_subplots(rows=1, cols=2)

top = {
    'y' : crop_summary_N['N'][0:10].sort_values().index,
    'x' : crop_summary_N['N'][0:10].sort_values()
}

last = {
    'y' : crop_summary_N['N'][-10:].index,
    'x' : crop_summary_N['N'][-10:]
}

fig.add_trace(
    go.Bar(top,
           name="Most nitrogen required",
           marker_color=random.choice(colorarr),
           orientation='h',
          text=top['x']),

    row=1, col=1
)

fig.add_trace(
    go.Bar(last,
           name="Least nitrogen required",
           marker_color=random.choice(colorarr),
           orientation='h',
          text=last['x']),
    row=1, col=2
)
fig.update_traces(texttemplate='%{text}', textposition='inside')
fig.update_layout(title_text="Nitrogen (N)",
                  plot_bgcolor='white',
                  font_size=7,
                  font_color='black',
                 height=500)

fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```
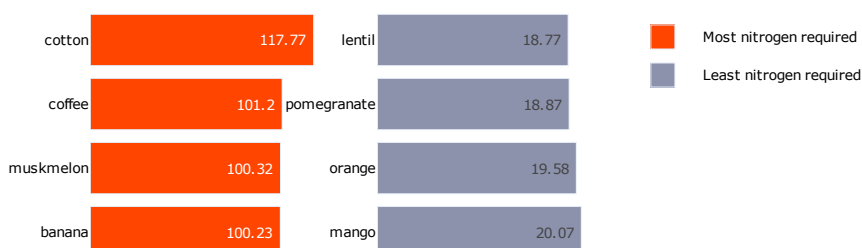
Nitrogen (N)

cotton	117.77		lentil	18.77	■ Most nitrogen required
coffee	101.2		pomegranate	18.87	■ Least nitrogen required
muskmelon	100.32		orange	19.58	
banana	100.23		mango	20.07	

watermelon	99.42	pigeonpeas	20.73
rice	79.89	kidneybeans	20.75
jute	78.4	apple	20.8
maize	77.76	mungbean	20.99
papaya	49.88	mothbeans	21.44
chickpea	40.09	coconut	21.98

In [43]:

```python
crop_summary_P = crop_summary.sort_values(by='P', ascending=False)

fig = make_subplots(rows=1, cols=2)

top = {
    'y' : crop_summary_P['P'][0:10].sort_values().index,
    'x' : crop_summary_P['P'][0:10].sort_values()
}

last = {
    'y' : crop_summary_P['P'][-10:].index,
    'x' : crop_summary_P['P'][-10:]
}

fig.add_trace(
    go.Bar(top,
            name="Most phosphorus required",
            marker_color=random.choice(colorarr),
            orientation='h',
        text=top['x']),

    row=1, col=1
)

fig.add_trace(
    go.Bar(last,
            name="Least phosphorus required",
            marker_color=random.choice(colorarr),
            orientation='h',
        text=last['x']),
    row=1, col=2
)
fig.update_traces(texttemplate='%{text}', textposition='inside')
fig.update_layout(title_text="Phosphorus (P)",
                    plot_bgcolor='white',
                    font_size=7,
                    font_color='black',
                  height=500)

fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```
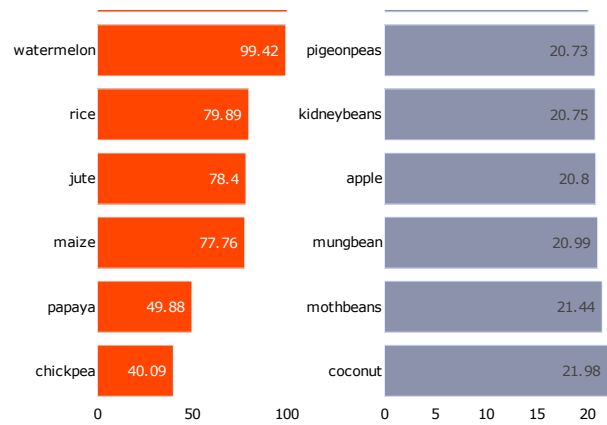
Phosphorus (P)

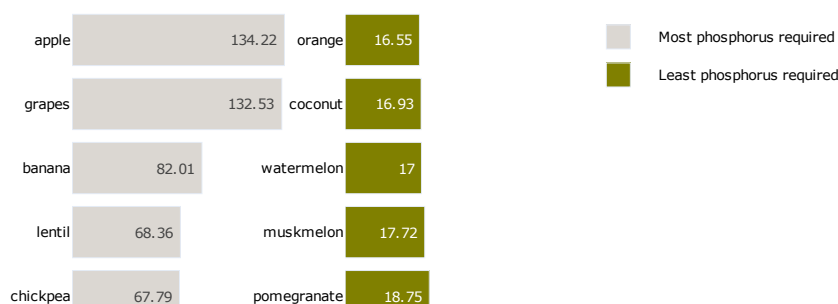| Crop	Value		Crop	Value
pigeonpeas	67.73		mango	27.18
kidneybeans	67.54		coffee	28.74
blackgram	67.47		cotton	46.24
papaya	59.05		jute	46.86
maize	48.44		mungbean	47.28

In [44]:

```python
crop_summary_K = crop_summary.sort_values(by='K', ascending=False)

fig = make_subplots(rows=1, cols=2)

top = {
    'y' : crop_summary_K['K'][0:10].sort_values().index,
    'x' : crop_summary_K['K'][0:10].sort_values()
}

last = {
    'y' : crop_summary_K['K'][-10:].index,
    'x' : crop_summary_K['K'][-10:]
}

fig.add_trace(
    go.Bar(top,
            name="Most potassium required",
            marker_color=random.choice(colorarr),
            orientation='h',
          text=top['x']),

    row=1, col=1
)

fig.add_trace(
    go.Bar(last,
            name="Least potassium required",
            marker_color=random.choice(colorarr),
            orientation='h',
          text=last['x']),
    row=1, col=2
)
fig.update_traces(texttemplate='%{text}', textposition='inside')
fig.update_layout(title_text="Potassium (K)",
                  plot_bgcolor='white',
                  font_size=7,
                  font_color='black',
                  height=500)

fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```
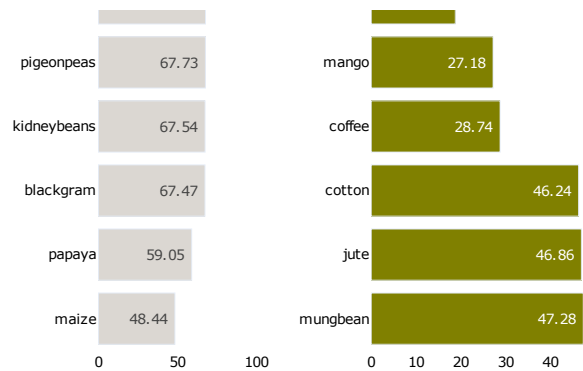
Potassium (K)
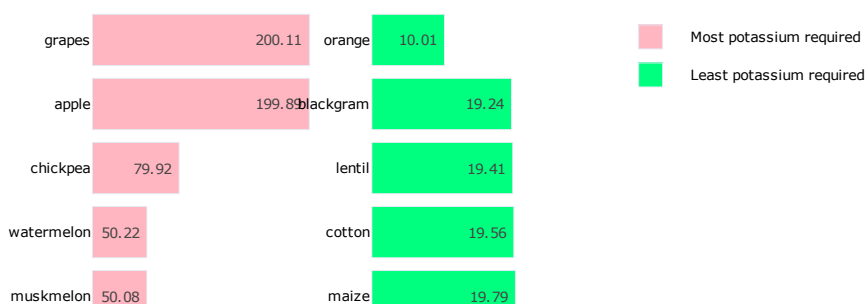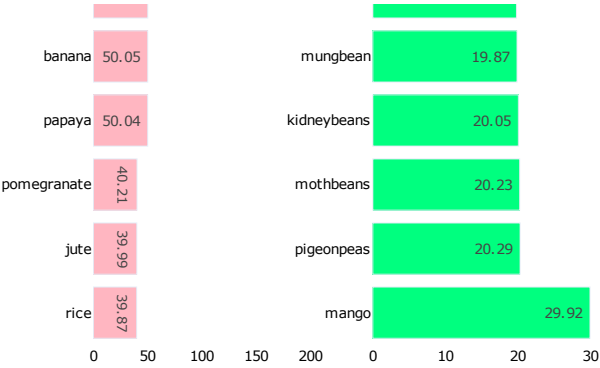
Crop	K (Most)	Crop	K (Least)
grapes	200.11	orange	10.01
apple	199.89	blackgram	19.24
chickpea	79.92	lentil	19.41
watermelon	50.22	cotton	19.56
muskmelon	50.08	maize	19.79

Most potassium required

Least potassium required

In [45]:

```python
fig = go.Figure()
fig.add_trace(go.Bar(
    x=crop_summary.index,
    y=crop_summary['N'],
    name='Nitrogen',
    marker_color='indianred'
))
fig.add_trace(go.Bar(
    x=crop_summary.index,
    y=crop_summary['P'],
    name='Phosphorous',
    marker_color='lightsalmon'
))
fig.add_trace(go.Bar(
    x=crop_summary.index,
    y=crop_summary['K'],
    name='Potash',
    marker_color='crimson'
))

fig.update_layout(title="N, P, K values comparision between crops",
                  plot_bgcolor='white',
                  barmode='group',
                  xaxis_tickangle=-45)

fig.show()
```

N, P, K values comparision between crops
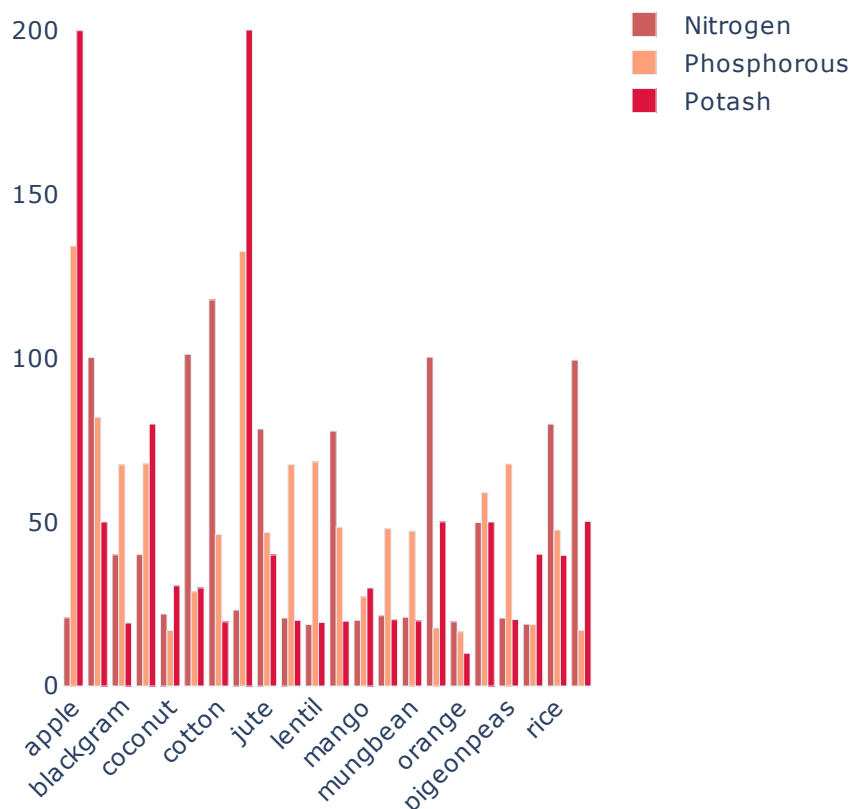
In [46]:

```python
labels = ['Nitrogen(N)','Phosphorous(P)','Potash(K)']
fig = make_subplots(rows=1, cols=5, specs=[[{'type':'domain'}, {'type':'domain'},
                                          {'type':'domain'}, {'type':'domain'},
                                          {'type':'domain'}]])

rice_npk = crop_summary[crop_summary.index=='rice']
values = [rice_npk['N'][0], rice_npk['P'][0], rice_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Rice"),1, 1)

cotton_npk = crop_summary[crop_summary.index=='cotton']
values = [cotton_npk['N'][0], cotton_npk['P'][0], cotton_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Cotton"),1, 2)

jute_npk = crop_summary[crop_summary.index=='jute']
values = [jute_npk['N'][0], jute_npk['P'][0], jute_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Jute"),1, 3)

maize_npk = crop_summary[crop_summary.index=='maize']
values = [maize_npk['N'][0], maize_npk['P'][0], maize_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Maize"),1, 4)

lentil_npk = crop_summary[crop_summary.index=='lentil']
values = [lentil_npk['N'][0], lentil_npk['P'][0], lentil_npk['K'][0]]
fig.add_trace(go.Pie(labels=labels, values=values,name="Lentil"),1, 5)

fig.update_traces(hole=.4, hoverinfo="label+percent+name")
fig.update_layout(
    title_text="NPK ratio for rice, cotton, jute, maize, lentil",
    annotations=[dict(text='Rice',x=0.06,y=0.8, font_size=7, showarrow=False),
                 dict(text='Cotton',x=0.26,y=0.8, font_size=7, showarrow=False),
                 dict(text='Jute',x=0.50,y=0.8, font_size=7, showarrow=False),
                 dict(text='Maize',x=0.74,y=0.8, font_size=7, showarrow=False),
                 dict(text='Lentil',x=0.94,y=0.8, font_size=7, showarrow=False)])
fig.show()
```

NPK ratio for rice, cotton, jute, maize, lentil
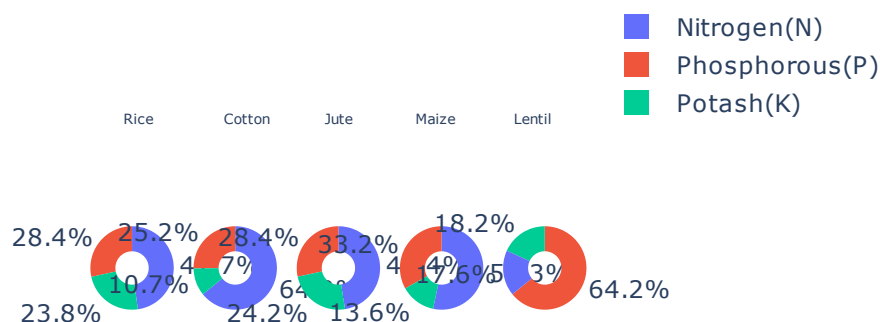
In [48]:

```python
Nitrogen(N)','Phosphorous(P)','Potash(K)']
'type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'doma
type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'domai
_subplots(rows=2, cols=5, specs=specs)
= ['rgb(255, 128, 0)', 'rgb(0, 153, 204)', 'rgb(173, 173, 133)']

crop_summary[crop_summary.index=='apple']
apple_npk['N'][0], apple_npk['P'][0], apple_npk['K'][0]]
ce(go.Pie(labels=labels, values=values,name="Apple", marker_colors=cafe_colors),1, 1)

= crop_summary[crop_summary.index=='banana']
banana_npk['N'][0], banana_npk['P'][0], banana_npk['K'][0]]
ce(go.Pie(labels=labels, values=values,name="Banana", marker_colors=cafe_colors),1, 2)

= crop_summary[crop_summary.index=='grapes']
grapes_npk['N'][0], grapes_npk['P'][0], grapes_npk['K'][0]]
ce(go.Pie(labels=labels, values=values,name="Grapes", marker_colors=cafe_colors),1, 3)

= crop_summary[crop_summary.index=='orange']
orange_npk['N'][0], orange_npk['P'][0], orange_npk['K'][0]]
ce(go.Pie(labels=labels, values=values,name="Orange", marker_colors=cafe_colors),1, 4)

crop_summary[crop_summary.index=='mango']
mango_npk['N'][0], mango_npk['P'][0], mango_npk['K'][0]]
ce(go.Pie(labels=labels, values=values,name="Mango", marker_colors=cafe_colors),1, 5)

= crop_summary[crop_summary.index=='coconut']
coconut_npk['N'][0], coconut_npk['P'][0], coconut_npk['K'][0]]
ce(go.Pie(labels=labels, values=values,name="Coconut", marker_colors=cafe_colors),2, 1)

= crop_summary[crop_summary.index=='papaya']
papaya_npk['N'][0], papaya_npk['P'][0], papaya_npk['K'][0]]
ce(go.Pie(labels=labels, values=values,name="Papaya", marker_colors=cafe_colors),2, 2)

_npk = crop_summary[crop_summary.index=='pomegranate']
pomegranate_npk['N'][0], pomegranate_npk['P'][0], pomegranate_npk['K'][0]]
ce(go.Pie(labels=labels, values=values,name="Pomegranate", marker_colors=cafe_colors),2,

_npk = crop_summary[crop_summary.index=='watermelon']
watermelon_npk['N'][0], watermelon_npk['P'][0], watermelon_npk['K'][0]]
ce(go.Pie(labels=labels, values=values,name="Watermelon", marker_colors=cafe_colors),2,

npk = crop_summary[crop_summary.index=='muskmelon']
muskmelon_npk['N'][0], muskmelon_npk['P'][0], muskmelon_npk['K'][0]]
ce(go.Pie(labels=labels, values=values,name="Muskmelon", marker_colors=cafe_colors),2, 5

layout(
ext="NPK ratio for fruits",
ions=[dict(text='Apple',x=0.06,y=1.08, font_size=7, showarrow=False),
      dict(text='Banana',x=0.26,y=1.08, font_size=7, showarrow=False),
      dict(text='Grapes',x=0.50,y=1.08, font_size=7, showarrow=False),
      dict(text='Orange',x=0.74,y=1.08, font_size=7, showarrow=False),
      dict(text='Mango',x=0.94,y=1.08, font_size=7, showarrow=False),
      dict(text='Coconut',x=0.06,y=0.46, font_size=7, showarrow=False),
      dict(text='Papaya',x=0.26,y=0.46, font_size=7, showarrow=False),
      dict(text='Pomegranate',x=0.50,y=0.46, font_size=7, showarrow=False),
      dict(text='Watermelon',x=0.74,y=0.46, font_size=7, showarrow=False),
      dict(text='Muskmelon',x=0.94,y=0.46, font_size=7, showarrow=False)])
```

NPK ratio for fruits

Apple   Banana   Grapes   Orange   Mango

- Potash(K)
- Phosphorous(P)
- Nitrogen(N)

37.8%   35.3%   37.2%   35.9%   35.2%   58.8%
5.86%   21.5%   6.51%   21.7%   26%

Coconut   Papaya   Pomegranate   Watermelon   Muskmelon

31.6%   31.5%   24.2%   30.1%   29.8%
24.4%   31.4%   24.1%   10.2%   10.5%   59.7%
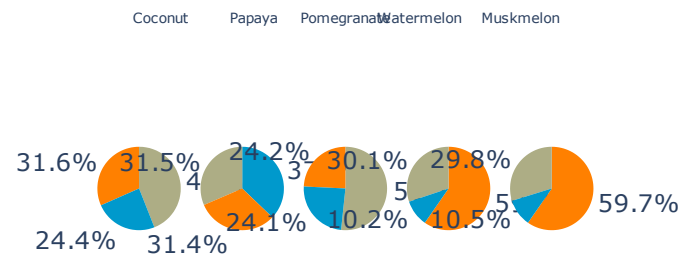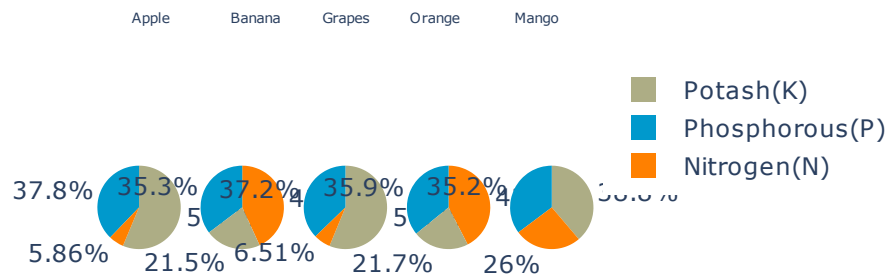
In [51]:

```python
crop_scatter = data[(data['label']=='rice') |
                    (data['label']=='jute') |
                    (data['label']=='cotton') |
                    (data['label']=='maize') |
                    (data['label']=='lentil')]

fig = px.scatter(crop_scatter, x="temperature", y="humidity", color="label", symbol
fig.update_layout(plot_bgcolor='yellow')
fig.update_xaxes(showgrid=True)
fig.update_yaxes(showgrid=True)

fig.show()
```
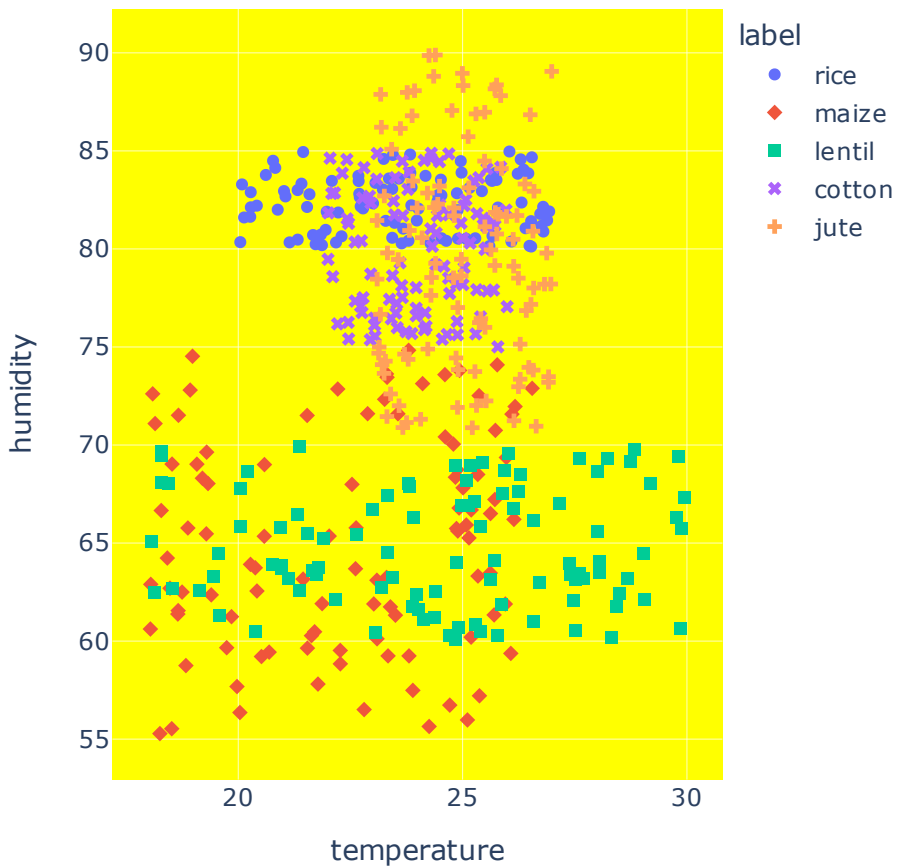
In [52]:

```python
fig = px.bar(crop_summary, x=crop_summary.index, y=["rainfall", "temperature", "hum
fig.update_layout(title_text="Comparision between rainfall, temerature and humidity
                  plot_bgcolor='white',
                  height=500)

fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```

Comparision between rainfall, temerature and humidi
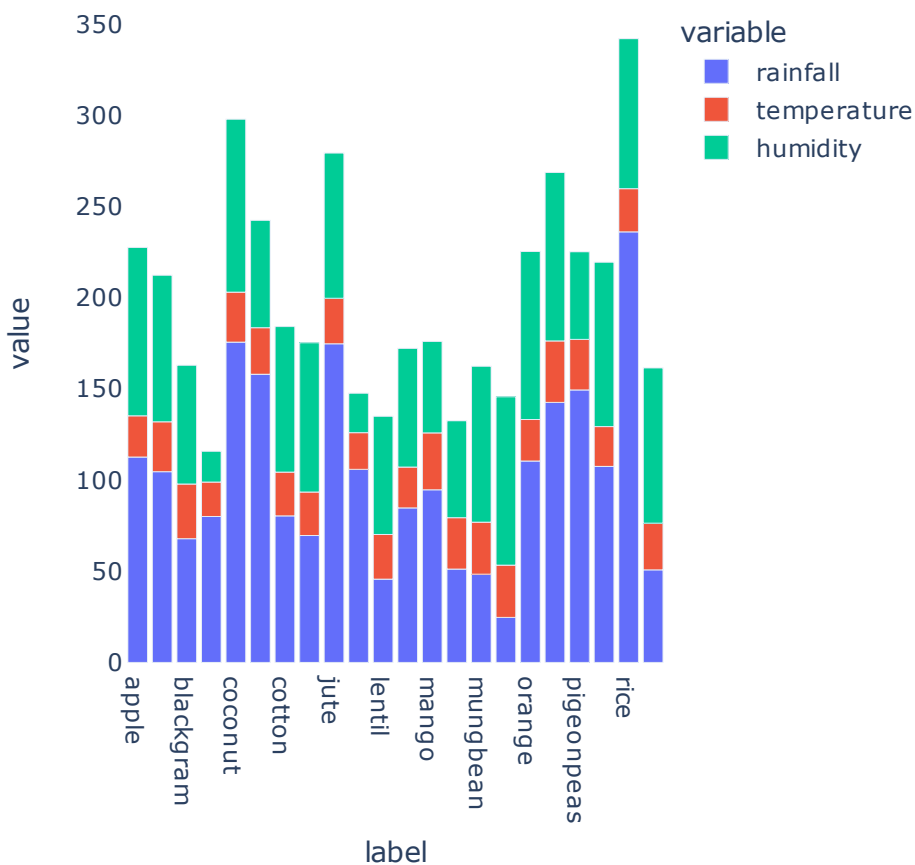
In [53]:

```
1 data.corr()
```

Out[53]:

	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

In [54]:

```
1 fig, ax = plt.subplots(1, 1, figsize=(15, 9))
2 sns.heatmap(data.corr(), annot=True,cmap='Wistia')
3 ax.set(xlabel='features')
4 ax.set(ylabel='features')
5
6 plt.title('Correlation between different features', fontsize = 15, c='black')
7 plt.show()
```


Correlation between different features
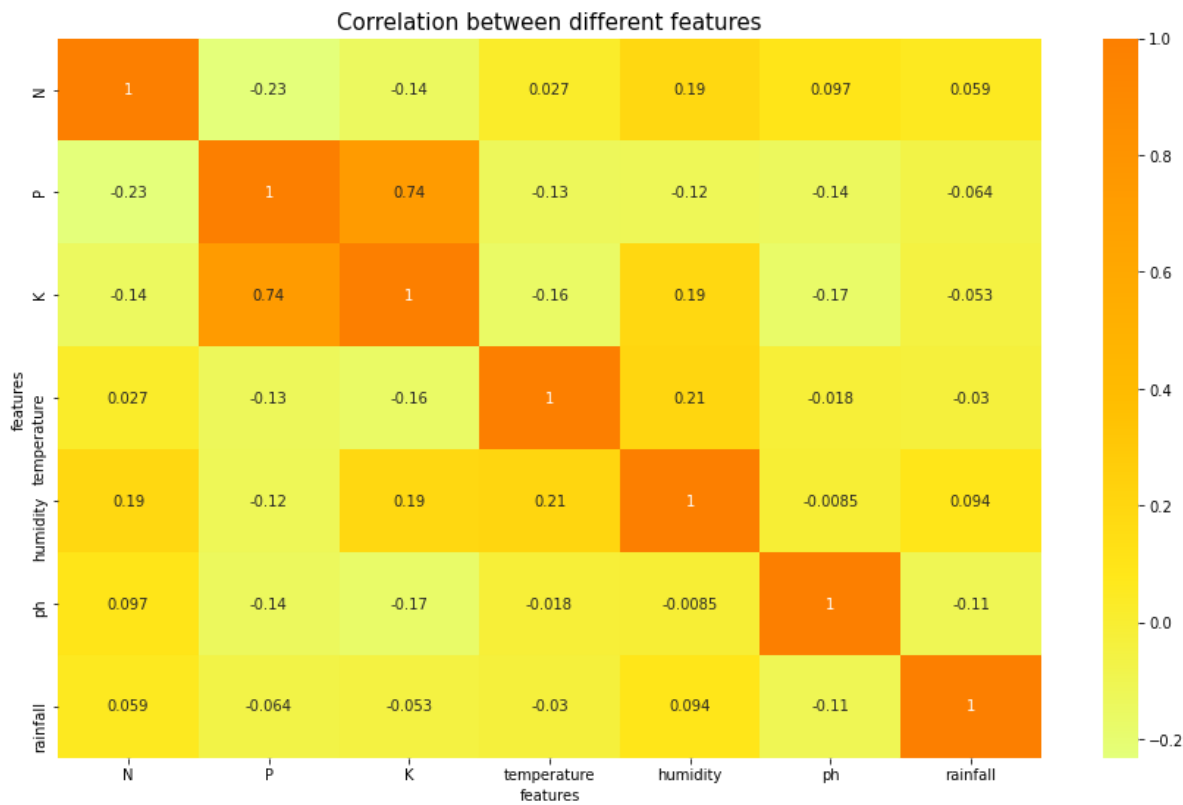
In [55]:

```python
X = data.drop('label', axis=1)
y = data['label']
```

In [56]:

```python
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33,
                                                    shuffle = True, random_state =
```

In [57]:

```python
import lightgbm as lgb
model = lgb.LGBMClassifier()
model.fit(X_train, y_train)
```

Out[57]:

```
LGBMClassifier()
```

In [58]:

```python
y_pred=model.predict(X_test)
```

In [59]:

```python
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_pred, y_test)
print('LightGBM Model accuracy score: {0:0.4f}'.format(accuracy_score(y_test, y_pre
```

```
LightGBM Model accuracy score: 0.9890
```

In [60]:

```python
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(15,15))
sns.heatmap(cm, annot=True, fmt=".0f", linewidths=.5, square = True, cmap = 'Blues'
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
all_sample_title = 'Confusion Matrix - score:'+str(accuracy_score(y_test,y_pred))
plt.title(all_sample_title, size = 15);
plt.show()
```


Confusion Matrix - score:0.9889807162534435

In [61]:

```python
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	31
banana	1.00	1.00	1.00	32
blackgram	0.97	0.97	0.97	35
chickpea	1.00	1.00	1.00	39
coconut	1.00	1.00	1.00	30
coffee	1.00	0.97	0.98	32
cotton	0.97	1.00	0.98	28
grapes	1.00	1.00	1.00	33
jute	0.97	0.94	0.95	31
kidneybeans	1.00	1.00	1.00	30
lentil	1.00	0.96	0.98	26
maize	0.93	0.97	0.95	29
mango	1.00	1.00	1.00	29
mothbeans	0.97	1.00	0.99	39
mungbean	1.00	1.00	1.00	31
muskmelon	1.00	1.00	1.00	31
orange	1.00	1.00	1.00	40
papaya	1.00	1.00	1.00	43
pigeonpeas	1.00	0.97	0.98	32
pomegranate	1.00	1.00	1.00	38
rice	0.95	0.97	0.96	40
watermelon	1.00	1.00	1.00	27
accuracy			0.99	726
macro avg	0.99	0.99	0.99	726
weighted avg	0.99	0.99	0.99	726

In [80]:

```python
from sklearn.tree import DecisionTreeClassifier
classifier= DecisionTreeClassifier(criterion='entropy', random_state=0)
classifier.fit(X_train, y_train)
```

Out[80]:

```
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

In [81]:

```python
y_pred=classifier.predict(X_test)
```

In [82]:

```python
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_pred, y_test)
print('Decision Tree Model accuracy score: {0:0.4f}'.format(accuracy_score(y_test,
```
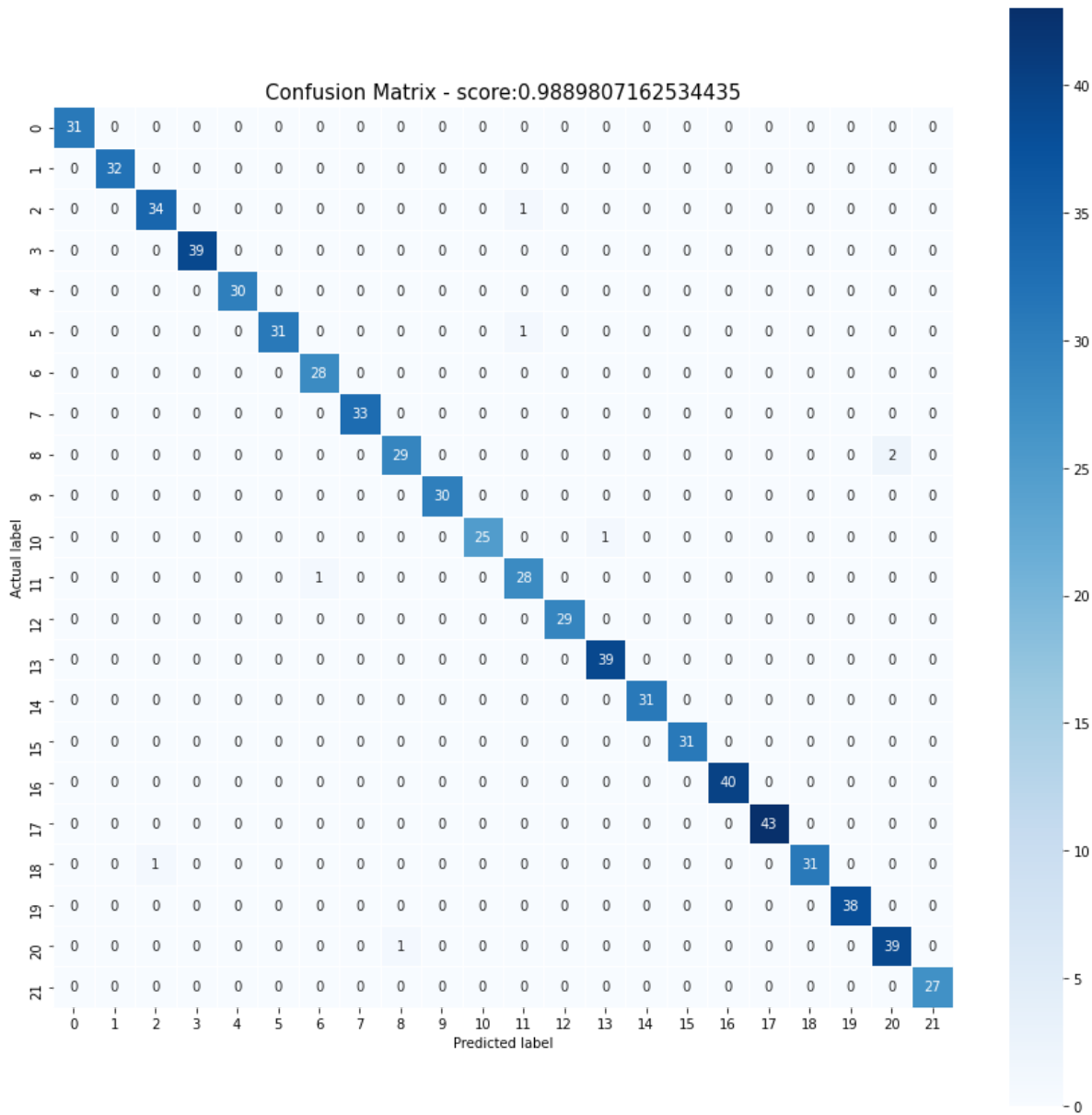
Decision Tree Model accuracy score: 0.9848

In [83]:

```python
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(15,15))
sns.heatmap(cm, annot=True, fmt=".0f", linewidths=.5, square = True, cmap = 'Blues'
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
all_sample_title = 'Confusion Matrix - score:'+str(accuracy_score(y_test,y_pred))
plt.title(all_sample_title, size = 15);
plt.show()
```


Confusion Matrix - score:0.9848484848484849

In [84]:

```python
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	31
banana	1.00	1.00	1.00	32
blackgram	0.97	0.94	0.96	35
chickpea	1.00	1.00	1.00	39
coconut	1.00	1.00	1.00	30
coffee	1.00	1.00	1.00	32
cotton	1.00	1.00	1.00	28
grapes	1.00	1.00	1.00	33
jute	0.93	0.87	0.90	31
kidneybeans	1.00	1.00	1.00	30
lentil	1.00	0.92	0.96	26
maize	0.97	1.00	0.98	29
mango	1.00	1.00	1.00	29
mothbeans	0.93	0.97	0.95	39
mungbean	1.00	1.00	1.00	31
muskmelon	1.00	1.00	1.00	31
orange	1.00	1.00	1.00	40
papaya	0.96	1.00	0.98	43
pigeonpeas	1.00	1.00	1.00	32
pomegranate	1.00	1.00	1.00	38
rice	0.95	0.95	0.95	40
watermelon	1.00	1.00	1.00	27
accuracy			0.98	726
macro avg	0.99	0.98	0.99	726
weighted avg	0.98	0.98	0.98	726

In [85]:

```python
from sklearn.ensemble import RandomForestClassifier
classifier_rf= RandomForestClassifier(n_estimators= 10, criterion="entropy")
classifier_rf.fit(X_train, y_train)
```

Out[85]:

RandomForestClassifier(criterion='entropy', n_estimators=10)

In [86]:

```python
y_pred= classifier_rf.predict(X_test)
```

In [87]:

```python
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_pred, y_test)
print('Random Forest Model accuracy score: {0:0.4f}'.format(accuracy_score(y_test,
```

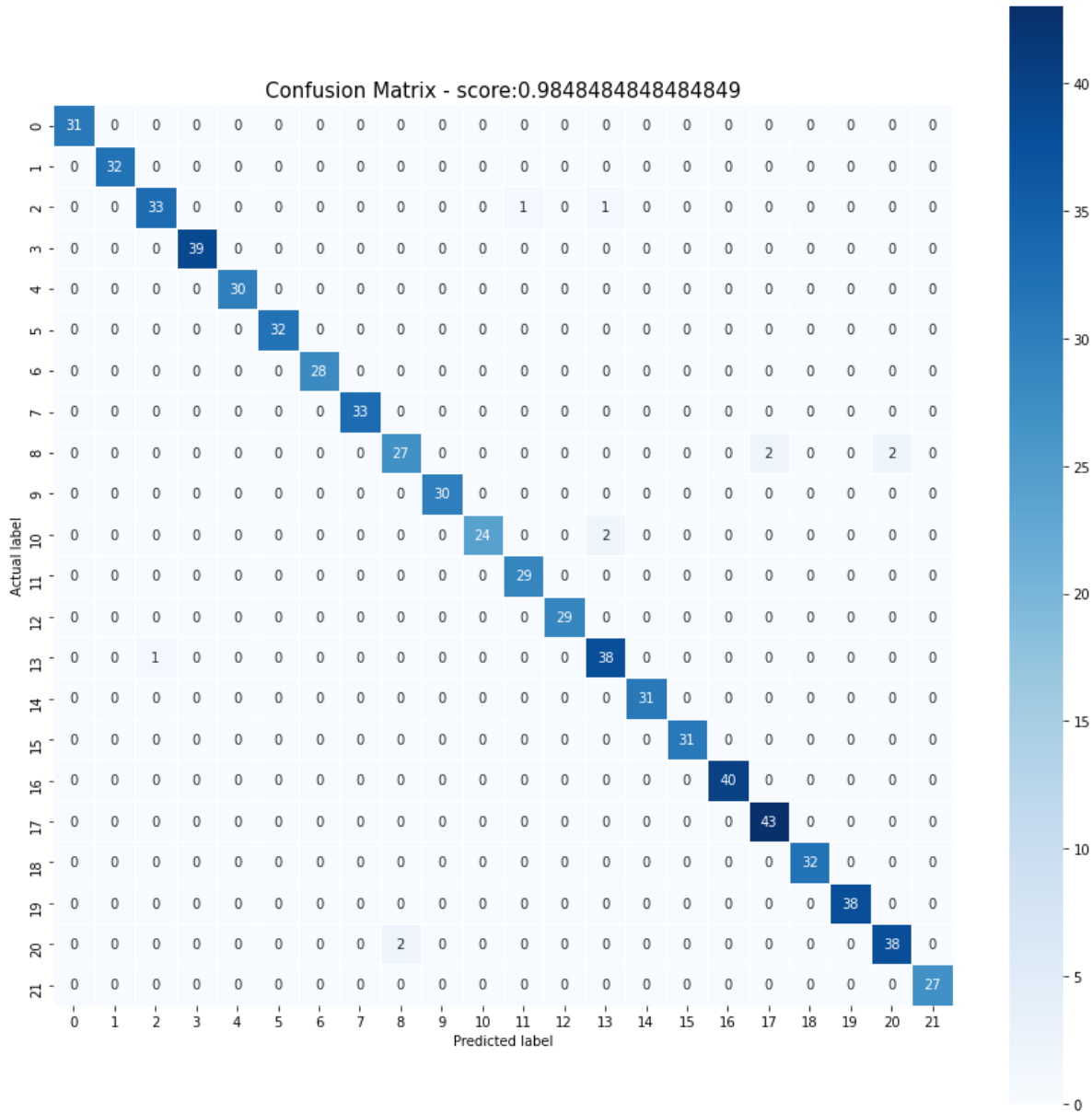Random Forest Model accuracy score: 0.9904

In [88]:

```python
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(15,15))
sns.heatmap(cm, annot=True, fmt=".0f", linewidths=.5, square = True, cmap = 'Blues'
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
all_sample_title = 'Confusion Matrix - score:'+str(accuracy_score(y_test,y_pred))
plt.title(all_sample_title, size = 15);
plt.show()
```

In [89]:

```python
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	31
banana	1.00	1.00	1.00	32
blackgram	0.97	0.97	0.97	35
chickpea	1.00	1.00	1.00	39
coconut	1.00	1.00	1.00	30
coffee	1.00	1.00	1.00	32
cotton	1.00	1.00	1.00	28
grapes	1.00	1.00	1.00	33
jute	0.94	0.94	0.94	31
kidneybeans	1.00	1.00	1.00	30
lentil	0.96	1.00	0.98	26
maize	0.97	1.00	0.98	29
mango	1.00	1.00	1.00	29
mothbeans	1.00	0.95	0.97	39
mungbean	1.00	1.00	1.00	31
muskmelon	1.00	1.00	1.00	31
orange	1.00	1.00	1.00	40
papaya	1.00	0.98	0.99	43
pigeonpeas	1.00	1.00	1.00	32
pomegranate	1.00	1.00	1.00	38
rice	0.95	0.97	0.96	40
watermelon	1.00	1.00	1.00	27
accuracy			0.99	726
macro avg	0.99	0.99	0.99	726
weighted avg	0.99	0.99	0.99	726

In [79]:

```python
from sklearn.linear_model import LogisticRegression
classifier_lr = LogisticRegression(random_state = 0)
classifier_lr.fit(X_train, y_train)
```

Out[79]:

```
LogisticRegression(random_state=0)
```

In [75]:

```python
y_pred = classifier_lr.predict(X_test)
```

In [76]:

```python
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_pred, y_test)
print('Logistic Regression Model accuracy score: {0:0.4f}'.format(accuracy_score(y_
```
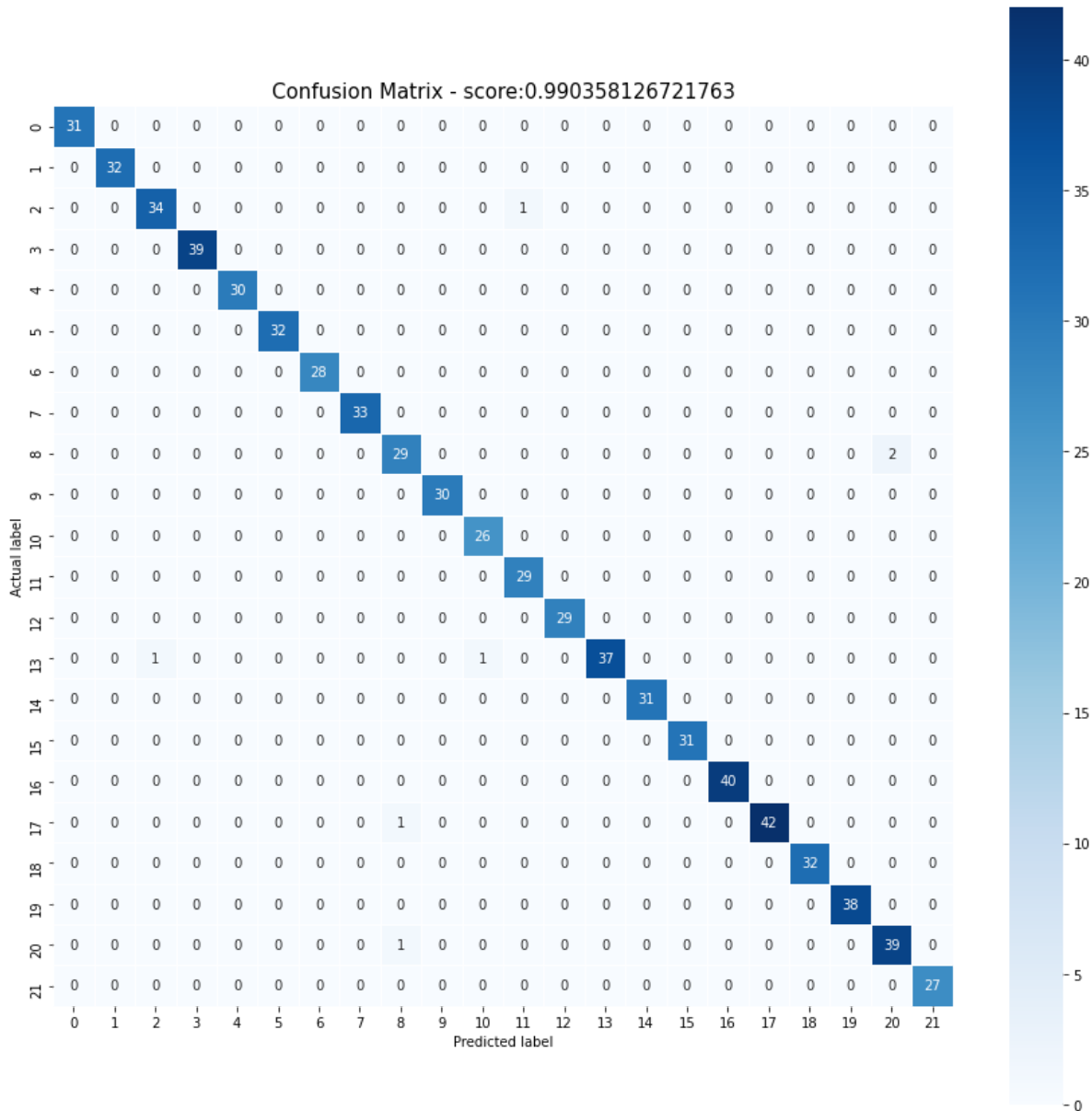
```
Logistic Regression Model accuracy score: 0.9435
```

In [78]:

```python
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(15,15))
sns.heatmap(cm, annot=True, fmt=".0f", linewidths=.5, square = True, cmap = 'Blues'
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
all_sample_title = 'Confusion Matrix - score:'+str(accuracy_score(y_test,y_pred))
plt.title(all_sample_title, size = 15);
plt.show()
```


Confusion Matrix - score:0.9435261707988981

In [77]:

```python
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```
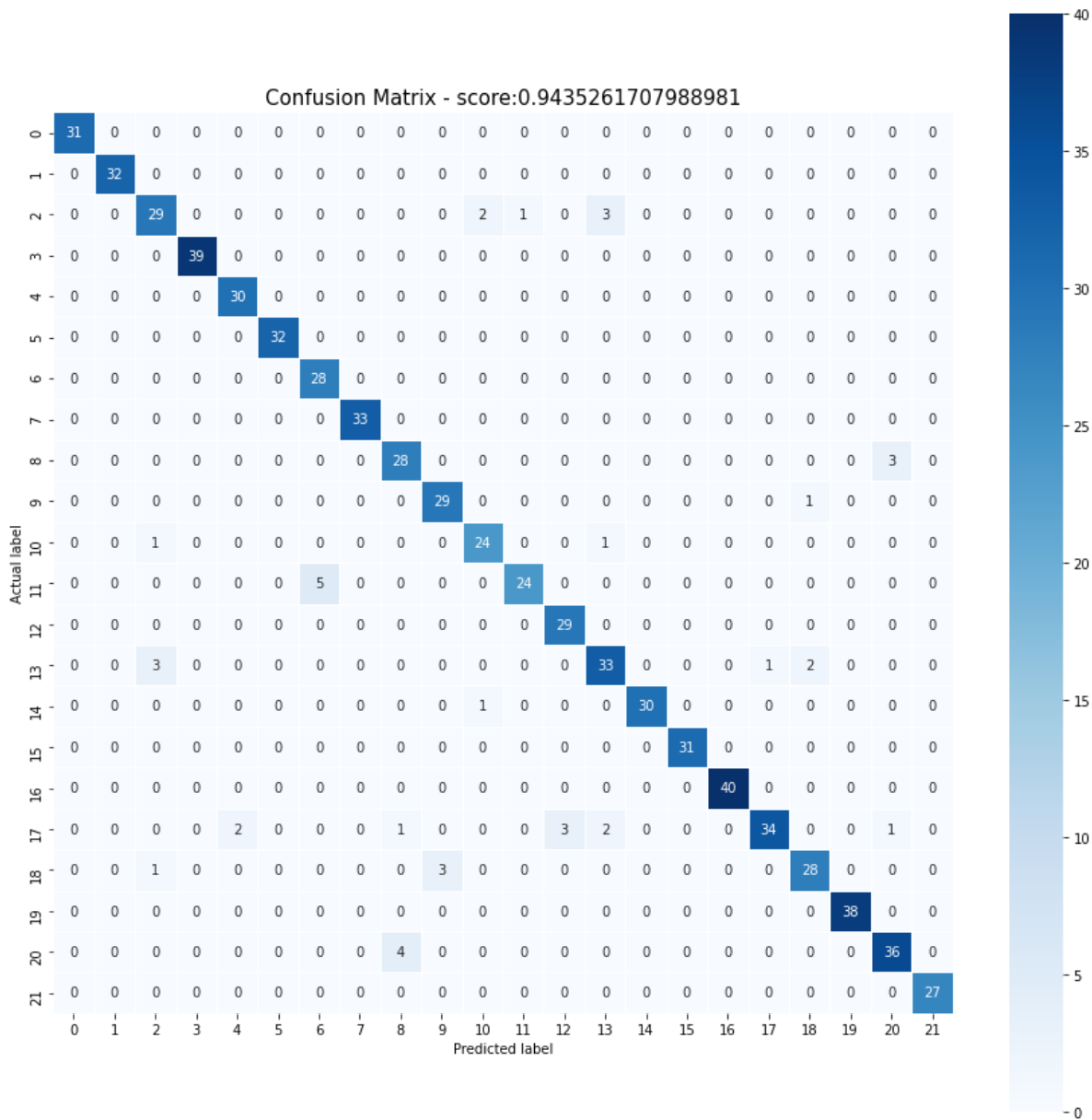
```
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        31
      banana       1.00      1.00      1.00        32
   blackgram       0.85      0.83      0.84        35
    chickpea       1.00      1.00      1.00        39
     coconut       0.94      1.00      0.97        30
      coffee       1.00      1.00      1.00        32
      cotton       0.85      1.00      0.92        28
      grapes       1.00      1.00      1.00        33
        jute       0.85      0.90      0.88        31
  kidneybeans       0.91      0.97      0.94        30
      lentil       0.89      0.92      0.91        26
       maize       0.96      0.83      0.89        29
       mango       0.91      1.00      0.95        29
    mothbeans       0.85      0.85      0.85        39
    mungbean       1.00      0.97      0.98        31
   muskmelon       1.00      1.00      1.00        31
      orange       1.00      1.00      1.00        40
      papaya       0.97      0.79      0.87        43
  pigeonpeas       0.90      0.88      0.89        32
 pomegranate       1.00      1.00      1.00        38
        rice       0.90      0.90      0.90        40
  watermelon       1.00      1.00      1.00        27

    accuracy                           0.94       726
   macro avg       0.94      0.95      0.94       726
weighted avg       0.95      0.94      0.94       726
```