```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
import tensorflow as tf
```

```
import sklearn
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

```
df=pd.read_csv('agriculture_dataset.csv')
```

```
df
```

	High_Resolution_RGB	Multispectral_Images	Thermal_Images	Temporal_Images	Spatial_Resolution	GPS_Coordi
0	0	0	0	0	0.667324	2
1	1	1	0	0	1.459000	2
2	0	0	0	0	0.500442	8
3	0	0	0	0	1.865161	6
4	0	1	1	1	1.392331	8
...	...	...	...	...	...	
212014	0	0	0	0	1.613700	2
212015	0	0	0	0	2.167963	8
212016	0	0	0	0	1.553286	1
212017	0	0	0	0	1.550687	8
212018	0	0	0	0	1.409110	9

212019 rows × 32 columns

```
df.head()
```

	High_Resolution_RGB	Multispectral_Images	Thermal_Images	Temporal_Images	Spatial_Resolution	GPS_Coordinates
0	0	0	0	0	0.667324	201538
1	1	1	0	0	1.459000	215854
2	0	0	0	0	0.500442	890802
3	0	0	0	0	1.865161	605584
4	0	1	1	1	1.392331	871732

5 rows × 32 columns

```
df.tail()
```

	High_Resolution_RGB	Multispectral_Images	Thermal_Images	Temporal_Images	Spatial_Resolution	GPS_Coordi
212014	0	0	0	0	1.613700	2
212015	0	0	0	0	2.167963	8
212016	0	0	0	0	1.553286	1
212017	0	0	0	0	1.550687	8
212018	0	0	0	0	1.409110	9

5 rows × 32 columns

```
df.isnull().sum()
```

	0
High_Resolution_RGB	0
Multispectral_Images	0
Thermal_Images	0
Temporal_Images	0
Spatial_Resolution	0
GPS_Coordinates	0
Field_Boundaries	0
Elevation_Data	0
Canopy_Coverage	0
NDVI	0
SAVI	0
Chlorophyll_Content	0
Leaf_Area_Index	0
Crop_Stress_Indicator	0
Temperature	0
Humidity	0
Rainfall	0
Wind_Speed	0
Soil_Moisture	0
Soil_pH	0
Organic_Matter	0
Pest_Hotspots	0
Weed_Coverage	0
Pest_Damage	0
Crop_Growth_Stage	0
Expected_Yield	0
Crop_Type	0
Ground_Truth_Segmentation	0
Bounding_Boxes	0
Water_Flow	0
Drainage_Features	0
Crop_Health_Label	0

dtype: int64

```
df.describe()
```

	High_Resolution_RGB	Multispectral_Images	Thermal_Images	Temporal_Images	Spatial_Resolution	GPS_Coordina
count	212019.000000	212019.000000	212019.000000	212019.000000	212019.000000	212019.00
mean	0.200232	0.299714	0.399492	0.100430	1.200244	550433.74
std	0.400175	0.458134	0.489795	0.300573	0.499427	260335.47
min	0.000000	0.000000	0.000000	0.000000	-0.959732	100003.00
25%	0.000000	0.000000	0.000000	0.000000	0.862755	324279.00
50%	0.000000	0.000000	0.000000	0.000000	1.199967	550668.00

```
df.max()
```

8 rows × 31 columns

	0
High_Resolution_RGB	1
Multispectral_Images	1
Thermal_Images	1
Temporal_Images	1
Spatial_Resolution	3.344946
GPS_Coordinates	999986
Field_Boundaries	3
Elevation_Data	99.999944
Canopy_Coverage	582.456962
NDVI	1.15804
SAVI	0.979315
Chlorophyll_Content	7.83929
Leaf_Area_Index	5.708551
Crop_Stress_Indicator	99
Temperature	46.78311
Humidity	89.999833
Rainfall	299.939332
Wind_Speed	15.104049
Soil_Moisture	34.9999
Soil_pH	8.627347
Organic_Matter	27.134563
Pest_Hotspots	1
Weed_Coverage	9.344244
Pest_Damage	99
Crop_Growth_Stage	4
Expected_Yield	6684.320978
Crop_Type	Wheat
Ground_Truth_Segmentation	1
Bounding_Boxes	9
Water_Flow	49.999447
Drainage_Features	1
Crop_Health_Label	1

dtype: object

```
df.min()
```

	0
High_Resolution_RGB	0
Multispectral_Images	0
Thermal_Images	0
Temporal_Images	0
Spatial_Resolution	-0.959732
GPS_Coordinates	100003
Field_Boundaries	1
Elevation_Data	10.000034
Canopy_Coverage	0.000237
NDVI	-0.165259
SAVI	-0.088355
Chlorophyll_Content	0.001995
Leaf_Area_Index	0.000702
Crop_Stress_Indicator	0
Temperature	2.683095
Humidity	30.00007
Rainfall	0.000108
Wind_Speed	0.002633
Soil_Moisture	5.000124
Soil_pH	4.266919
Organic_Matter	0.000001
Pest_Hotspots	0
Weed_Coverage	0.00194
Pest_Damage	0
Crop_Growth_Stage	1
Expected_Yield	-468.653721
Crop_Type	Maize
Ground_Truth_Segmentation	0
Bounding_Boxes	0
Water_Flow	0.000051
Drainage_Features	0
Crop_Health_Label	0

dtype: object

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 212019 entries, 0 to 212018
Data columns (total 32 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   High_Resolution_RGB       212019 non-null  int64
 1   Multispectral_Images      212019 non-null  int64
 2   Thermal_Images            212019 non-null  int64
 3   Temporal_Images           212019 non-null  int64
 4   Spatial_Resolution        212019 non-null  float64
 5   GPS_Coordinates           212019 non-null  int64
 6   Field_Boundaries          212019 non-null  int64
 7   Elevation_Data            212019 non-null  float64
 8   Canopy_Coverage           212019 non-null  float64
 9   NDVI                      212019 non-null  float64
 10  SAVI                      212019 non-null  float64
 11  Chlorophyll_Content       212019 non-null  float64
 12  Leaf_Area_Index           212019 non-null  float64
 13  Crop_Stress_Indicator     212019 non-null  int64
 14  Temperature               212019 non-null  float64
 15  Humidity                  212019 non-null  float64
 16  Rainfall                  212019 non-null  float64
 17  Wind_Speed                212019 non-null  float64
 18  Soil_Moisture             212019 non-null  float64
```

```
 19  Soil_pH                   212019 non-null  float64
 20  Organic_Matter            212019 non-null  float64
 21  Pest_Hotspots             212019 non-null  int64
 22  Weed_Coverage             212019 non-null  float64
 23  Pest_Damage               212019 non-null  int64
 24  Crop_Growth_Stage         212019 non-null  int64
 25  Expected_Yield            212019 non-null  float64
 26  Crop_Type                 212019 non-null  object
 27  Ground_Truth_Segmentation  212019 non-null  int64
 28  Bounding_Boxes            212019 non-null  int64
 29  Water_Flow                212019 non-null  float64
 30  Drainage_Features         212019 non-null  int64
 31  Crop_Health_Label         212019 non-null  int64
dtypes: float64(17), int64(14), object(1)
memory usage: 51.8+ MB
```

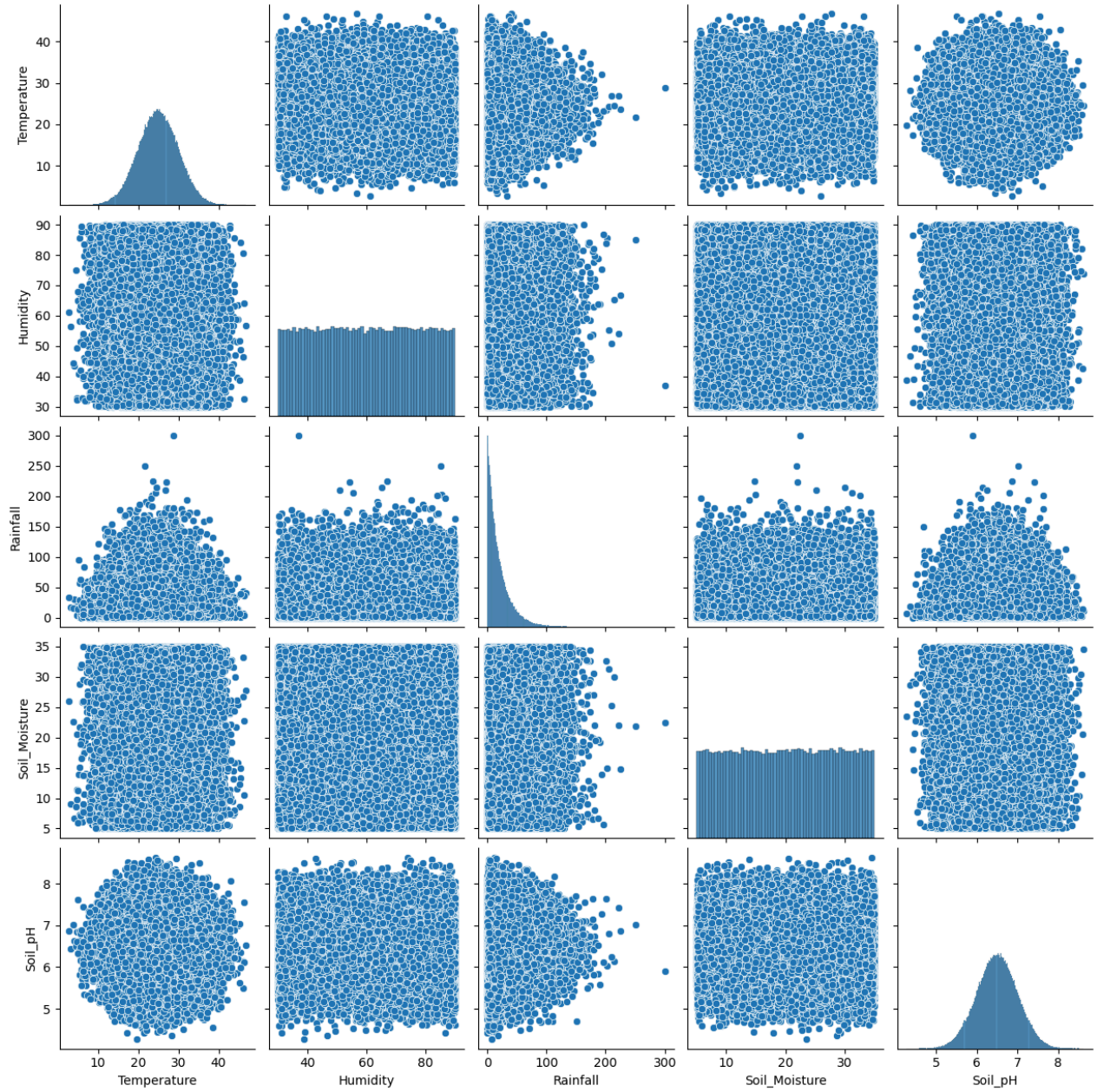
```
x=df[['Temperature', 'Humidity', 'Rainfall',  'Soil_Moisture','Soil_pH']]
```

```
len(x)
```

```
212019
```

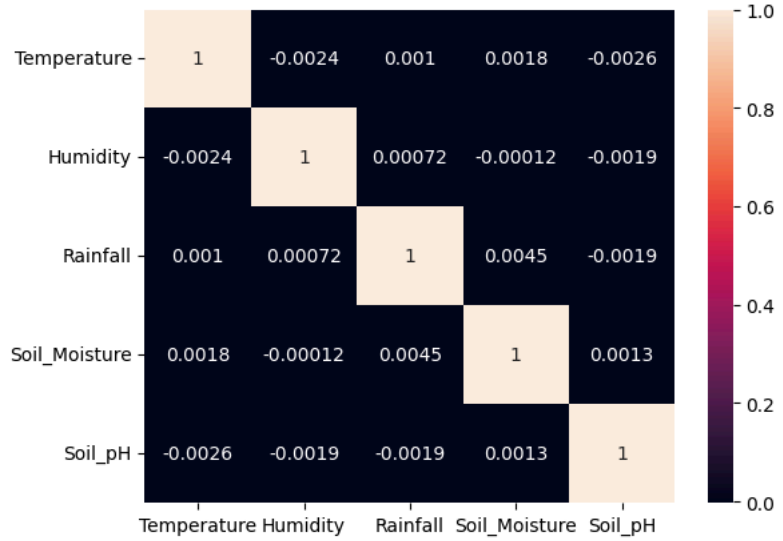
```
sns.pairplot(x)
```

```
<seaborn.axisgrid.PairGrid at 0x7f32d56a6360>
```




```
sns.heatmap(x.corr(),annot=True)
```

```
<Axes: >
```




```
y=df['Crop_Health_Label']
```

```
len(y)
```

```
212019
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

```
model=StandardScaler()
x_train=model.fit_transform(x_train)
x_test=model.transform(x_test)
```

```
model = tf.keras.Sequential([
    tf.keras.layers.Dense(16, activation='relu', input_shape=(5,)),
    tf.keras.layers.Dense(8, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)

model.summary()
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/core/dense.py:93: UserWarning: Do not pass an `input_sha
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Model: "sequential"
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	96
dense_1 (Dense)	(None, 8)	136
dense_2 (Dense)	(None, 1)	9

```
 Total params: 241 (964.00 B)
 Trainable params: 241 (964.00 B)
 Non-trainable params: 0 (0.00 B)
```

```
history = model.fit(
    x_train,
    y_train,
    epochs=10,
    batch_size=8,
    validation_split=0.2
)
```

```
Epoch 1/10
16962/16962 ━━━━━━━━━━━━━━━━━━━━ 40s 2ms/step - accuracy: 0.6976 - loss: 0.6163 - val_accuracy: 0.6965 - val_loss
Epoch 2/10
16962/16962 ━━━━━━━━━━━━━━━━━━━━ 40s 2ms/step - accuracy: 0.6995 - loss: 0.6119 - val_accuracy: 0.6965 - val_loss
```

```
Epoch 3/10
16962/16962 ━━━━━━━━━━━━━━━━━━━━ 51s 3ms/step - accuracy: 0.6982 - loss: 0.6127 - val_accuracy: 0.6965 - val_loss
Epoch 4/10
16962/16962 ━━━━━━━━━━━━━━━━━━━━ 40s 2ms/step - accuracy: 0.7005 - loss: 0.6107 - val_accuracy: 0.6965 - val_loss
Epoch 5/10
16962/16962 ━━━━━━━━━━━━━━━━━━━━ 40s 2ms/step - accuracy: 0.7001 - loss: 0.6110 - val_accuracy: 0.6965 - val_loss
Epoch 6/10
16962/16962 ━━━━━━━━━━━━━━━━━━━━ 40s 2ms/step - accuracy: 0.6971 - loss: 0.6134 - val_accuracy: 0.6965 - val_loss
Epoch 7/10
16962/16962 ━━━━━━━━━━━━━━━━━━━━ 39s 2ms/step - accuracy: 0.6995 - loss: 0.6113 - val_accuracy: 0.6965 - val_loss
Epoch 8/10
16962/16962 ━━━━━━━━━━━━━━━━━━━━ 39s 2ms/step - accuracy: 0.6987 - loss: 0.6120 - val_accuracy: 0.6965 - val_loss
Epoch 9/10
16962/16962 ━━━━━━━━━━━━━━━━━━━━ 40s 2ms/step - accuracy: 0.6989 - loss: 0.6118 - val_accuracy: 0.6965 - val_loss
Epoch 10/10
16962/16962 ━━━━━━━━━━━━━━━━━━━━ 42s 2ms/step - accuracy: 0.6985 - loss: 0.6122 - val_accuracy: 0.6965 - val_loss
```

```
loss, accuracy = model.evaluate(x_test, y_test)
print("Test Loss:", loss)
print("Test Accuracy:", accuracy)
```

```
1326/1326 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - accuracy: 0.7048 - loss: 0.6068
Test Loss: 0.6100826859474182
Test Accuracy: 0.701018750667572
```

```
# Predictions
y_pred_prob = model.predict(x_test)
y_pred = (y_pred_prob > 0.5).astype(int)
```