Exploratory Data Analysis (EDA)

```
import pandas as pd
df = pd.read_csv("/content/Fertilizer Prediction.csv")
```

```
print(df.head())
```

```
   Temparature  Humidity  Moisture Soil Type    Crop Type  Nitrogen  \
0           32        51        41       Red  Ground Nuts         7
1           35        58        35     Black       Cotton         4
2           27        55        43     Sandy    Sugarcane        28
3           33        56        56     Loamy  Ground Nuts        37
4           32        70        60       Red  Ground Nuts         4

   Potassium  Phosphorous Fertilizer Name
0          3           19        14-35-14
1         14           16            Urea
2          0           17           20-20
3          5           24           28-28
4          6            9        14-35-14
```

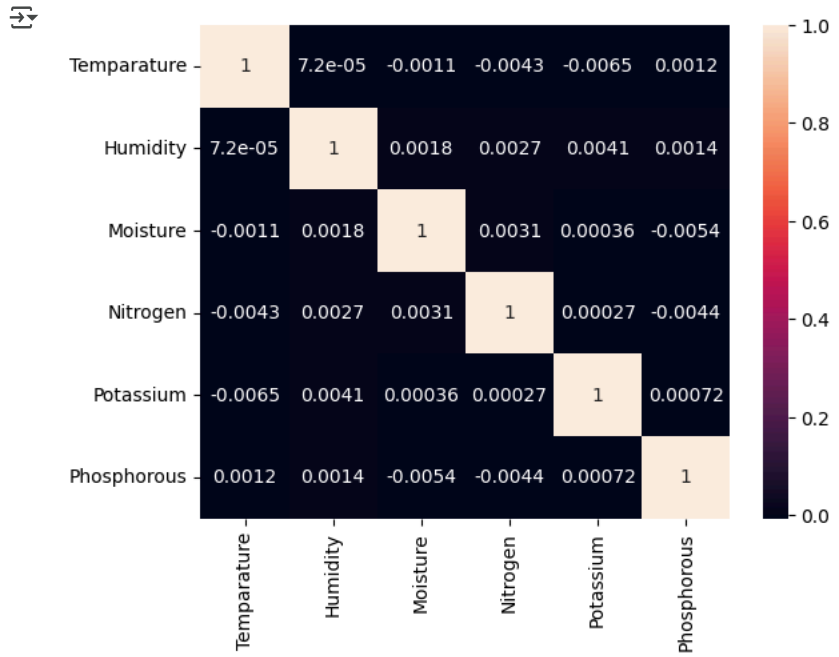
```
print(df.info())
print(df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   Temparature      100000 non-null  int64
 1   Humidity         100000 non-null  int64
 2   Moisture         100000 non-null  int64
 3   Soil Type        100000 non-null  object
 4   Crop Type        100000 non-null  object
 5   Nitrogen         100000 non-null  int64
 6   Potassium        100000 non-null  int64
 7   Phosphorous      100000 non-null  int64
 8   Fertilizer Name  100000 non-null  object
dtypes: int64(6), object(3)
memory usage: 6.9+ MB
None
         Temparature       Humidity      Moisture       Nitrogen  \
count  100000.000000  100000.000000  100000.00000  100000.000000
mean       31.503300      60.985810      45.00344      22.986770
std         4.019942       6.651393      11.83871      11.247289
min        25.000000      50.000000      25.00000       4.000000
25%        28.000000      55.000000      35.00000      13.000000
50%        32.000000      61.000000      45.00000      23.000000
75%        35.000000      67.000000      55.00000      33.000000
max        38.000000      72.000000      65.00000      42.000000

           Potassium   Phosphorous
count  100000.000000  100000.00000
mean        9.472220      21.01348
std         5.768565      12.39118
min         0.000000       0.00000
25%         4.000000      10.00000
50%         9.000000      21.00000
75%        14.000000      32.00000
max        19.000000      42.00000
```

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(df.corr(numeric_only=True), annot=True)
plt.show()
```



Data Transformation

```
df = df.dropna()
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Soil Type'] = le.fit_transform(df['Soil Type'])
df['Crop Type'] = le.fit_transform(df['Crop Type'])
df['Fertilizer Name'] = le.fit_transform(df['Fertilizer Name'])
```

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
df[df.select_dtypes(include='number').columns] = scaler.fit_transform(df.select_dtypes(include='number'))
```

Feature Selection

```
print(df.columns)
```

```
Index(['Temparature', 'Humidity', 'Moisture', 'Soil Type', 'Crop Type',
       'Nitrogen', 'Potassium', 'Phosphorous', 'Fertilizer Name'],
      dtype='object')
```

```
print(df.columns.tolist())
```

```
['Temparature', 'Humidity', 'Moisture', 'Soil Type', 'Crop Type', 'Nitrogen', 'Potassium', 'Phosphorous', 'Fertilizer Name']
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder

df_encoded = df.copy()

df_encoded['Fertilizer Name'] = df_encoded['Fertilizer Name'].astype(str)

for col in df_encoded.select_dtypes(include=['object']).columns:
    if col != 'Fertilizer Name':
        df_encoded[col] = LabelEncoder().fit_transform(df_encoded[col])

X = df_encoded.drop('Fertilizer Name', axis=1)
y = df_encoded['Fertilizer Name']


model = RandomForestClassifier()
model.fit(X, y)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report


data = pd.read_csv("/content/Fertilizer Prediction.csv")
print("Shape:", data.shape)
print("Columns:", data.columns)


label_enc = LabelEncoder()
for col in ["Soil Type", "Crop Type", "Fertilizer Name"]:
    data[col] = label_enc.fit_transform(data[col])

X = data.drop("Fertilizer Name", axis=1)
y = data["Fertilizer Name"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)


model = RandomForestClassifier(n_estimators=200, random_state=42)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("✅ Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Shape: (100000, 9)
Columns: Index(['Temparature', 'Humidity', 'Moisture', 'Soil Type', 'Crop Type',
       'Nitrogen', 'Potassium', 'Phosphorous', 'Fertilizer Name'],
      dtype='object')
✅ Accuracy: 0.14545

Classification Report:
                precision    recall  f1-score   support

           0       0.14      0.15      0.14      2876
           1       0.16      0.18      0.17      2898
           2       0.15      0.15      0.15      2834
           3       0.13      0.13      0.13      2836
           4       0.15      0.14      0.15      2847
           5       0.15      0.14      0.14      2844
           6       0.14      0.14      0.14      2865

    accuracy                           0.15     20000
   macro avg       0.15      0.15      0.15     20000
weighted avg       0.15      0.15      0.15     20000
```