```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.preprocessing import StandardScaler  , LabelEncoder
from sklearn.model_selection import train_test_split , cross_val_score, GridSearchCV
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
from sklearn.metrics import r2_score
```

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

```
data.describe()
```

```
data.dtypes
```

```
data.head()
```

```
X = data.drop(columns = ['Date' , 'City'  , 'Type'])
y = data['Type']
```

```
fig , ax = plt.subplots(5,2 , figsize=(12,8))
axes_ = [axes_row for  axes in  ax  for axes_row in axes]
for i,j in enumerate(X.columns):
  g = sns.boxplot(x = X[j] , ax = axes_[i])
  g.set_title(j)
  plt.tight_layout()
```

```
plt.figure(figsize = (20, 15))
```

```
plt.figure(figsize = (20, 15))
plotnumber = 1

for column in X:
    if plotnumber <= 6:
        ax = plt.subplot(5, 6, plotnumber)
        sns.displot(X[column])
        plt.xlabel(column)

    plotnumber += 1


plt.show()
```

```
y_le = LabelEncoder()
y = y_le.fit_transform(y)
```

```
std = StandardScaler()
X = std.fit_transform(X)
```

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

```
base = DecisionTreeClassifier()
```

```
model = XGBClassifier(use_label_encoder=False , eval_matric='logloss')
param_grid = {
    'n_estimators': [50, 100, 180, 200],
    'learning_rate': [0.01, 0.1, 0.5, 1.0],
    'max_depth': [3, 4, 5]
}

grid_search = GridSearchCV(
    estimator=model,
    param_grid=param_grid,
    cv=5,
    scoring='accuracy',
```

```
    scoring='accuracy',
    n_jobs=-1
)
```

```
grid_search.fit(X,y)
```

```
best_params= grid_search.best_params_
```

```
print(best_params)
```

```
xgb = XGBClassifier(**best_params ,use_label_encoder=False , eval_matric='logloss')
```

```
xgb.fit(X_train , y_train)
```

```
y_pred = xgb.predict(X_test)
```

```
print(r2_score(y_test , y_pred))
```

```
```