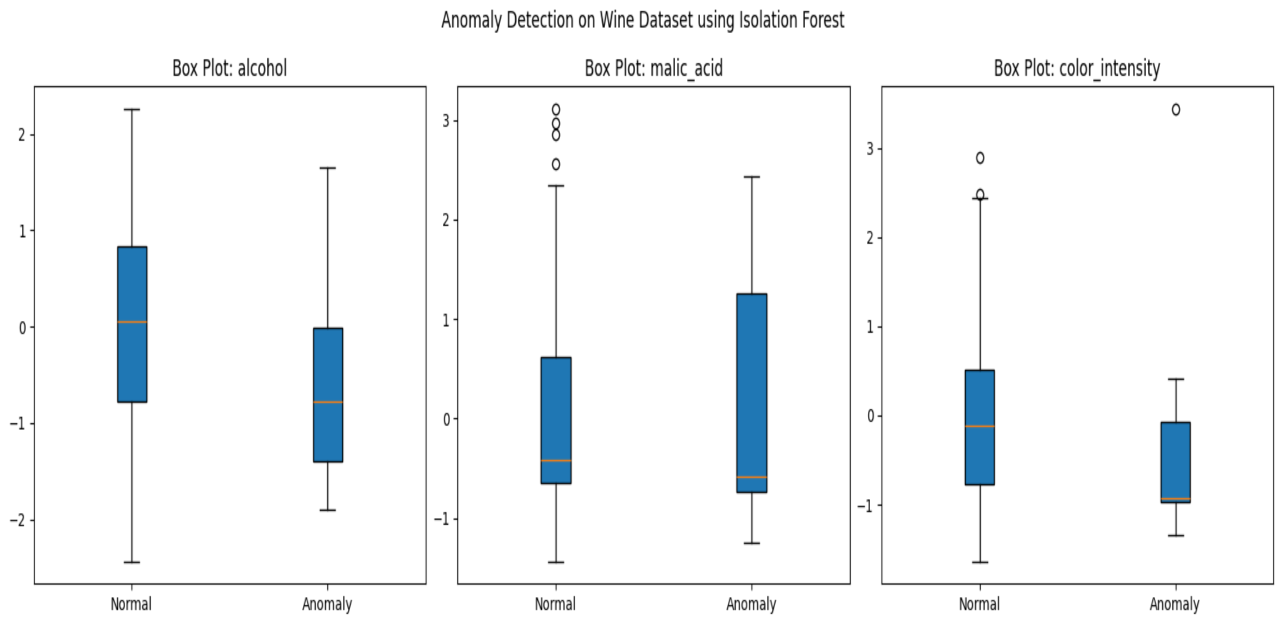
APPLIED ARTIFICIAL INTELLIGENCE

EXPERIMENT – 05

```
# Unsupervised Anomaly Detection using Isolation Forest on Wine Dataset
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
from sklearn.ensemble import IsolationForest
from sklearn.preprocessing import StandardScaler

# Load the Wine dataset
data = load_wine()
X = pd.DataFrame(data.data, columns=data.feature_names)

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_scaled = pd.DataFrame(X_scaled, columns=X.columns)

# Apply Isolation Forest for anomaly detection
iso_forest = IsolationForest(contamination=0.05, random_state=42)
X_scaled['Anomaly'] = iso_forest.fit_predict(X_scaled)

# Convert anomaly labels: -1 = Anomaly, 1 = Normal
X_scaled['Anomaly'] = X_scaled['Anomaly'].map({1: 'Normal', -1: 'Anomaly'})

# Box plot of selected features
selected_features = ['alcohol', 'malic_acid', 'color_intensity']
fig, axes = plt.subplots(1, 3, figsize=(18, 5))
for i, feature in enumerate(selected_features):
    axes[i].set_title(f"Box Plot: {feature}")
    axes[i].boxplot(
        [X_scaled[X_scaled['Anomaly'] == 'Normal'][feature],
         X_scaled[X_scaled['Anomaly'] == 'Anomaly'][feature]],
        tick_labels=['Normal', 'Anomaly'],  # ✅ updated parameter
        patch_artist=True
    )
plt.suptitle("Anomaly Detection on Wine Dataset using Isolation Forest")
plt.tight_layout()
plt.show()
```

output:


Anomaly Detection on Wine Dataset using Isolation Forest
Box Plot: alcohol
Normal
Anomaly
Box Plot: malic_acid
Normal
Anomaly
Box Plot: color_intensity
Normal
Anomaly