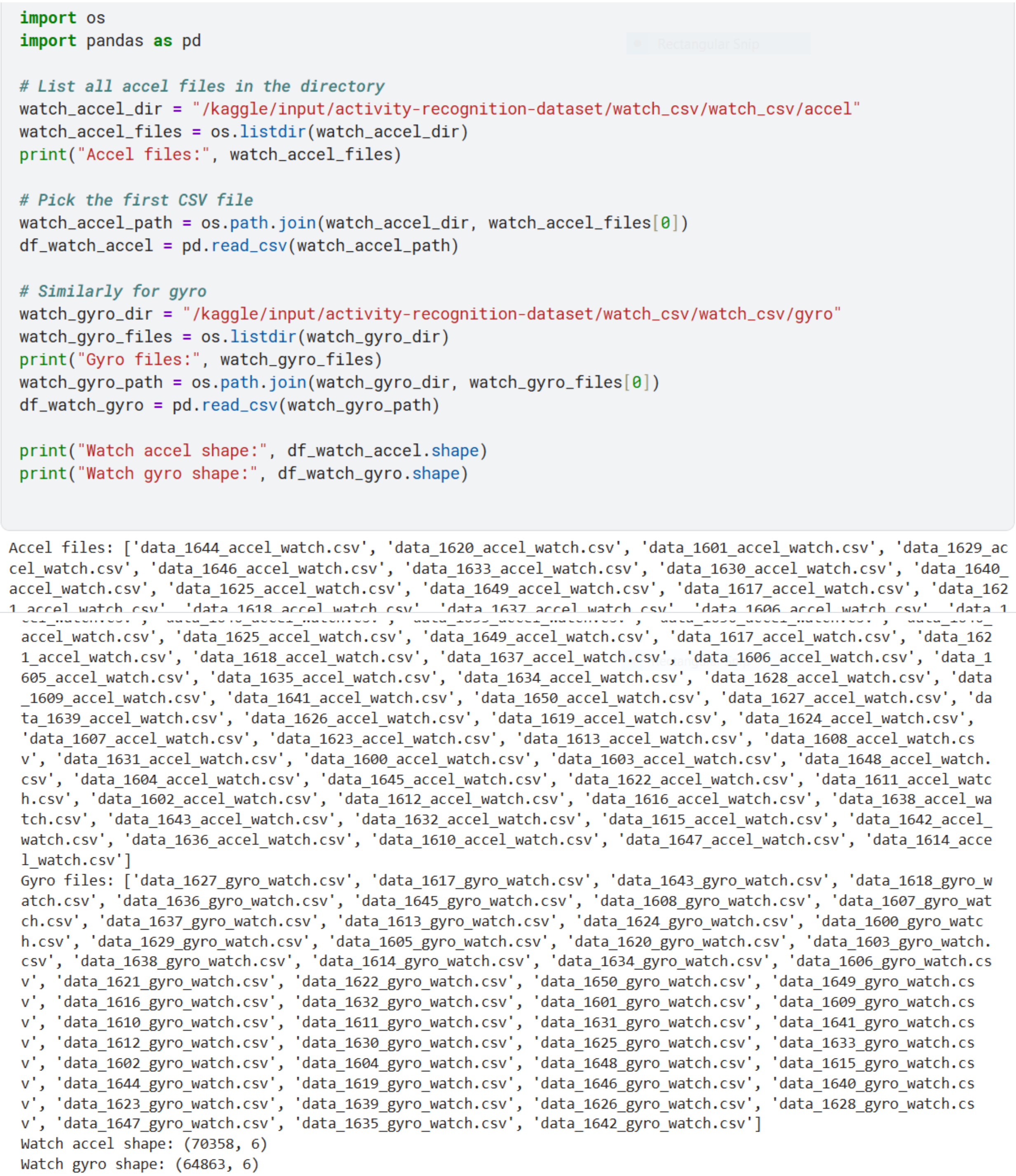
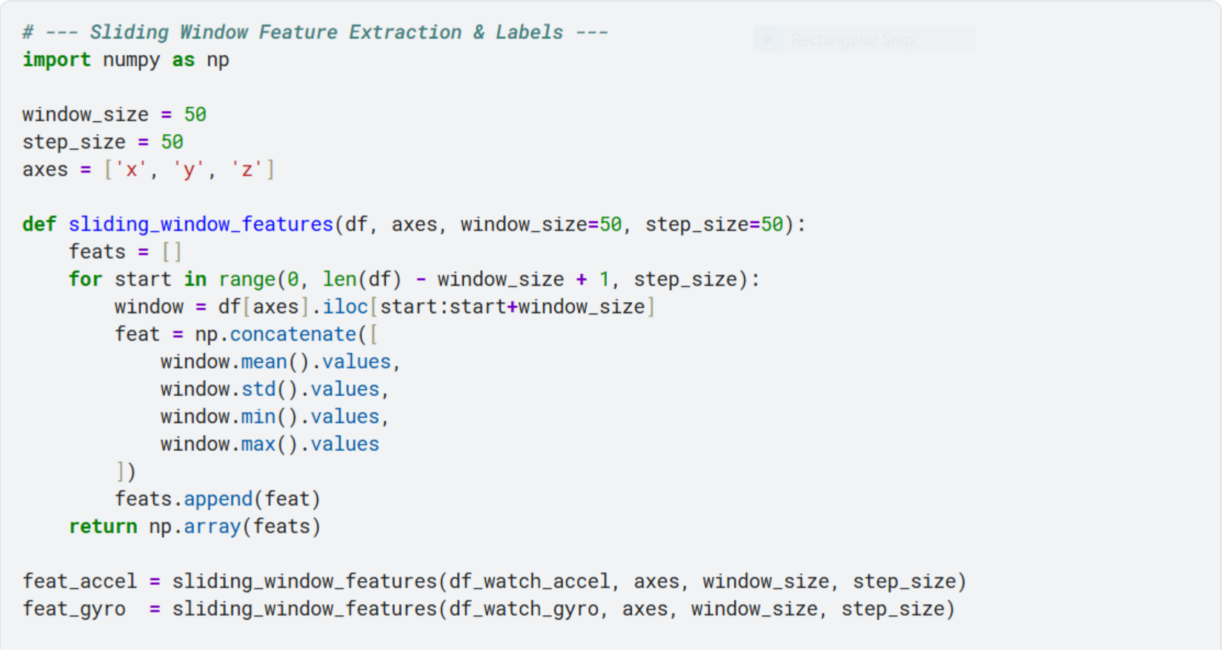


```python
import os
import pandas as pd

# List all accel files in the directory
watch_accel_dir = "/kaggle/input/activity-recognition-dataset/watch_csv/watch_csv/accel"
watch_accel_files = os.listdir(watch_accel_dir)
print("Accel files:", watch_accel_files)

# Pick the first CSV file
watch_accel_path = os.path.join(watch_accel_dir, watch_accel_files[0])
df_watch_accel = pd.read_csv(watch_accel_path)

# Similarly for gyro
watch_gyro_dir = "/kaggle/input/activity-recognition-dataset/watch_csv/watch_csv/gyro"
watch_gyro_files = os.listdir(watch_gyro_dir)
print("Gyro files:", watch_gyro_files)
watch_gyro_path = os.path.join(watch_gyro_dir, watch_gyro_files[0])
df_watch_gyro = pd.read_csv(watch_gyro_path)

print("Watch accel shape:", df_watch_accel.shape)
print("Watch gyro shape:", df_watch_gyro.shape)
```

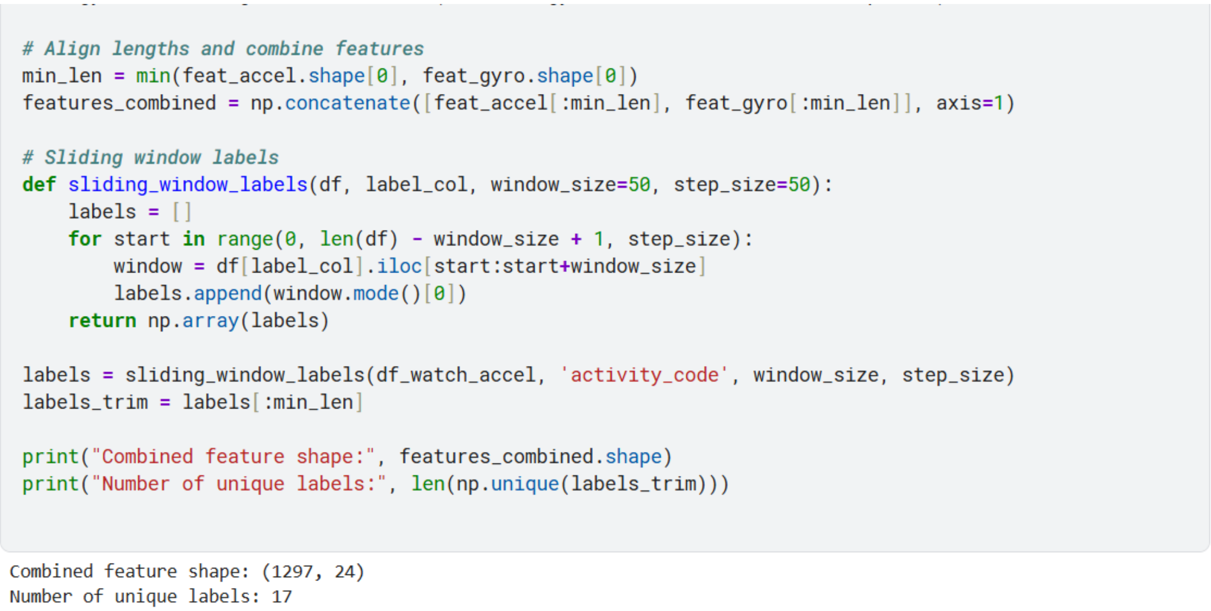
```
Accel files: ['data_1644_accel_watch.csv', 'data_1620_accel_watch.csv', 'data_1601_accel_watch.csv', 'data_1629_ac
cel_watch.csv', 'data_1646_accel_watch.csv', 'data_1633_accel_watch.csv', 'data_1630_accel_watch.csv', 'data_1640_
accel_watch.csv', 'data_1625_accel_watch.csv', 'data_1649_accel_watch.csv', 'data_1617_accel_watch.csv', 'data_162
```

```
accel_watch.csv', 'data_1625_accel_watch.csv', 'data_1649_accel_watch.csv', 'data_1617_accel_watch.csv', 'data_162
```

```
1_accel_watch.csv', 'data_1618_accel_watch.csv', 'data_1637_accel_watch.csv', 'data_1606_accel_watch.csv', 'data_1
605_accel_watch.csv', 'data_1635_accel_watch.csv', 'data_1634_accel_watch.csv', 'data_1628_accel_watch.csv', 'data
_1609_accel_watch.csv', 'data_1641_accel_watch.csv', 'data_1650_accel_watch.csv', 'data_1627_accel_watch.csv', 'da
ta_1639_accel_watch.csv', 'data_1626_accel_watch.csv', 'data_1619_accel_watch.csv', 'data_1624_accel_watch.csv',
'data_1607_accel_watch.csv', 'data_1623_accel_watch.csv', 'data_1613_accel_watch.csv', 'data_1608_accel_watch.cs
v', 'data_1631_accel_watch.csv', 'data_1600_accel_watch.csv', 'data_1603_accel_watch.csv', 'data_1648_accel_watch.
csv', 'data_1604_accel_watch.csv', 'data_1645_accel_watch.csv', 'data_1622_accel_watch.csv', 'data_1611_accel_watc
h.csv', 'data_1602_accel_watch.csv', 'data_1612_accel_watch.csv', 'data_1616_accel_watch.csv', 'data_1638_accel_wa
tch.csv', 'data_1643_accel_watch.csv', 'data_1632_accel_watch.csv', 'data_1615_accel_watch.csv', 'data_1642_accel_
watch.csv', 'data_1636_accel_watch.csv', 'data_1610_accel_watch.csv', 'data_1647_accel_watch.csv', 'data_1614_acce
l_watch.csv']
Gyro files: ['data_1627_gyro_watch.csv', 'data_1617_gyro_watch.csv', 'data_1643_gyro_watch.csv', 'data_1618_gyro_w
atch.csv', 'data_1636_gyro_watch.csv', 'data_1645_gyro_watch.csv', 'data_1608_gyro_watch.csv', 'data_1607_gyro_wat
ch.csv', 'data_1637_gyro_watch.csv', 'data_1613_gyro_watch.csv', 'data_1624_gyro_watch.csv', 'data_1600_gyro_watc
h.csv', 'data_1629_gyro_watch.csv', 'data_1605_gyro_watch.csv', 'data_1620_gyro_watch.csv', 'data_1603_gyro_watch.
csv', 'data_1638_gyro_watch.csv', 'data_1614_gyro_watch.csv', 'data_1634_gyro_watch.csv', 'data_1606_gyro_watch.cs
v', 'data_1621_gyro_watch.csv', 'data_1622_gyro_watch.csv', 'data_1650_gyro_watch.csv', 'data_1649_gyro_watch.cs
v', 'data_1616_gyro_watch.csv', 'data_1632_gyro_watch.csv', 'data_1601_gyro_watch.csv', 'data_1609_gyro_watch.cs
v', 'data_1610_gyro_watch.csv', 'data_1611_gyro_watch.csv', 'data_1631_gyro_watch.csv', 'data_1641_gyro_watch.cs
v', 'data_1612_gyro_watch.csv', 'data_1630_gyro_watch.csv', 'data_1625_gyro_watch.csv', 'data_1633_gyro_watch.cs
v', 'data_1602_gyro_watch.csv', 'data_1604_gyro_watch.csv', 'data_1648_gyro_watch.csv', 'data_1615_gyro_watch.cs
v', 'data_1644_gyro_watch.csv', 'data_1619_gyro_watch.csv', 'data_1646_gyro_watch.csv', 'data_1640_gyro_watch.cs
v', 'data_1623_gyro_watch.csv', 'data_1639_gyro_watch.csv', 'data_1626_gyro_watch.csv', 'data_1628_gyro_watch.cs
v', 'data_1647_gyro_watch.csv', 'data_1635_gyro_watch.csv', 'data_1642_gyro_watch.csv']
Watch accel shape: (70358, 6)
Watch gyro shape: (64863, 6)
```

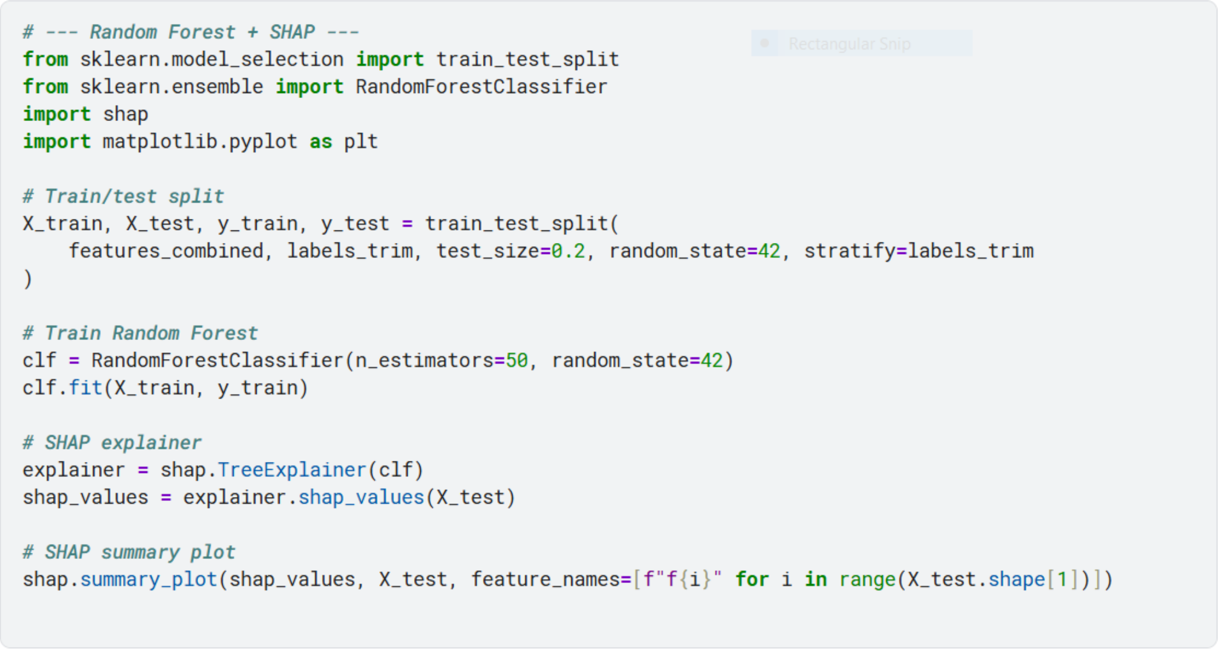


```python
# --- Sliding Window Feature Extraction & Labels ---
import numpy as np

window_size = 50
step_size = 50
axes = ['x', 'y', 'z']

def sliding_window_features(df, axes, window_size=50, step_size=50):
    feats = []
    for start in range(0, len(df) - window_size + 1, step_size):
        window = df[axes].iloc[start:start+window_size]
        feat = np.concatenate([
            window.mean().values,
            window.std().values,
            window.min().values,
            window.max().values
        ])
        feats.append(feat)
    return np.array(feats)

feat_accel = sliding_window_features(df_watch_accel, axes, window_size, step_size)
feat_gyro  = sliding_window_features(df_watch_gyro, axes, window_size, step_size)
```

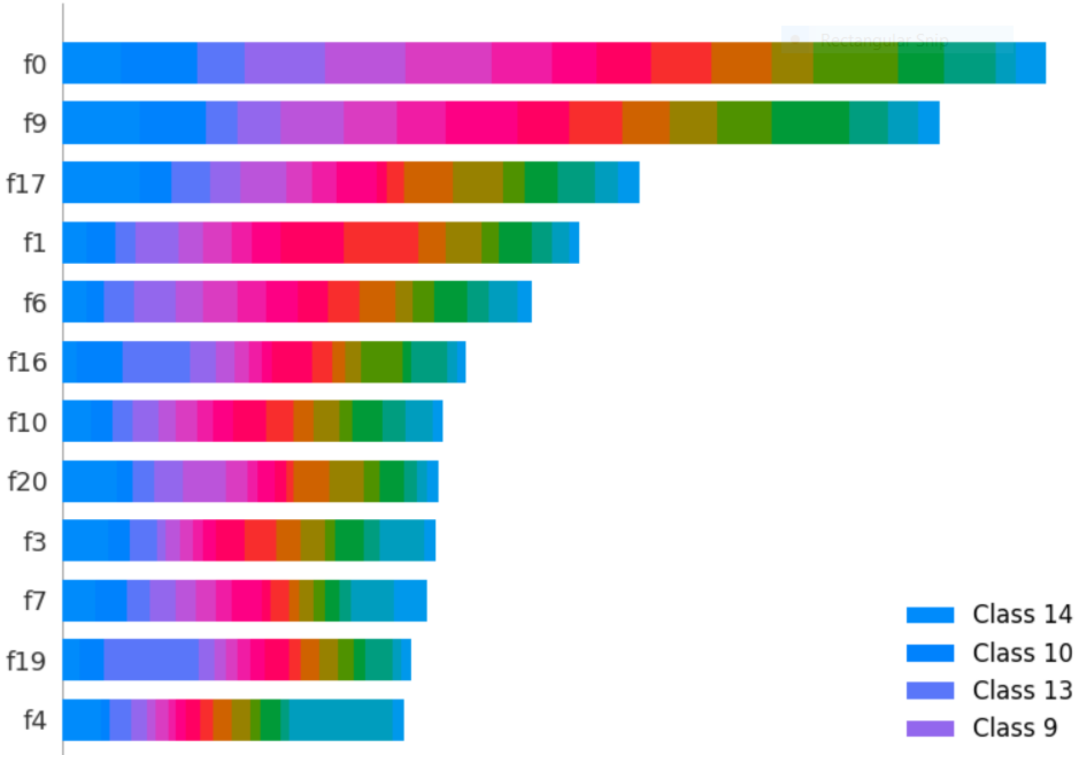


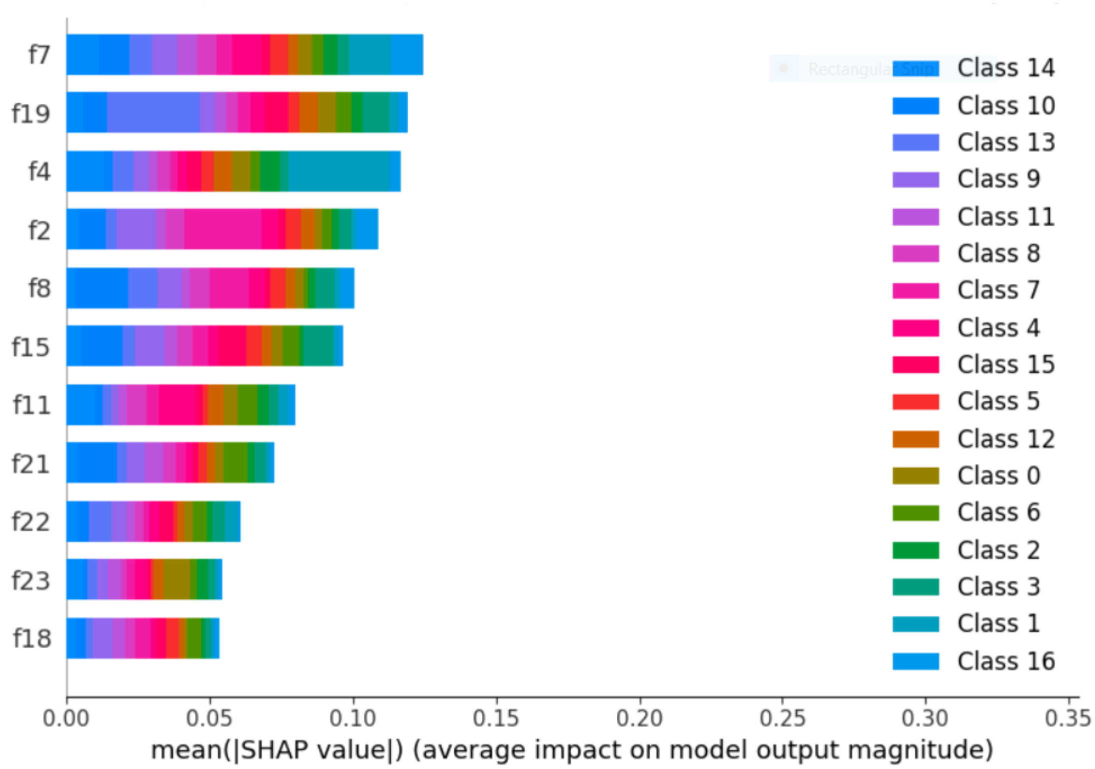
```python
# Align lengths and combine features
min_len = min(feat_accel.shape[0], feat_gyro.shape[0])
features_combined = np.concatenate([feat_accel[:min_len], feat_gyro[:min_len]], axis=1)

# Sliding window labels
def sliding_window_labels(df, label_col, window_size=50, step_size=50):
    labels = []
    for start in range(0, len(df) - window_size + 1, step_size):
        window = df[label_col].iloc[start:start+window_size]
        labels.append(window.mode()[0])
    return np.array(labels)

labels = sliding_window_labels(df_watch_accel, 'activity_code', window_size, step_size)
labels_trim = labels[:min_len]

print("Combined feature shape:", features_combined.shape)
print("Number of unique labels:", len(np.unique(labels_trim)))
```

```
Combined feature shape: (1297, 24)
Number of unique labels: 17
```

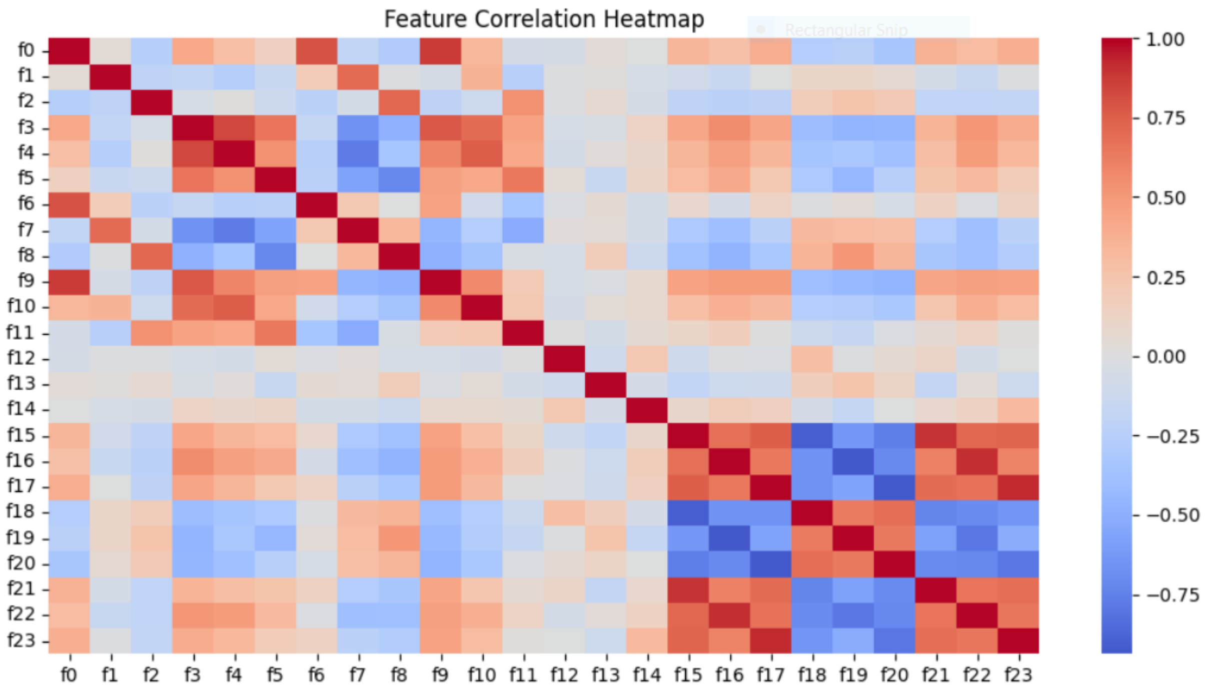


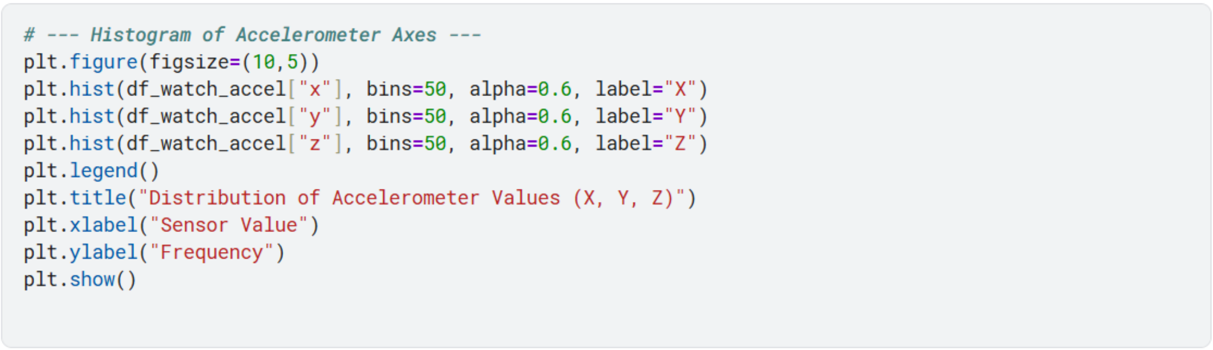
```python
# --- Random Forest + SHAP ---
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import shap
import matplotlib.pyplot as plt

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(
    features_combined, labels_trim, test_size=0.2, random_state=42, stratify=labels_trim
)

# Train Random Forest
clf = RandomForestClassifier(n_estimators=50, random_state=42)
clf.fit(X_train, y_train)

# SHAP explainer
explainer = shap.TreeExplainer(clf)
shap_values = explainer.shap_values(X_test)

# SHAP summary plot
shap.summary_plot(shap_values, X_test, feature_names=[f"f{i}" for i in range(X_test.shape[1])])
```



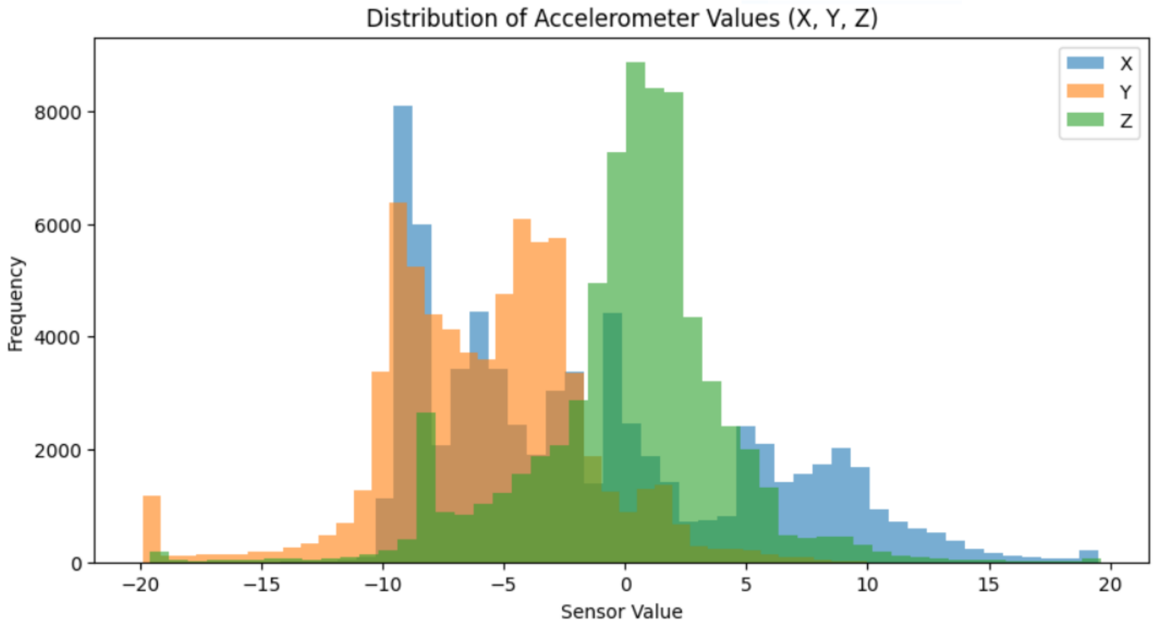


```
# --- Feature Correlation Heatmap ---
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

df_features = pd.DataFrame(features_combined, columns=[f"f{i}" for i in range(features_combined.shape[1]
plt.figure(figsize=(12,6))
sns.heatmap(df_features.corr(), cmap="coolwarm", center=0)
plt.title("Feature Correlation Heatmap")
plt.show()
```

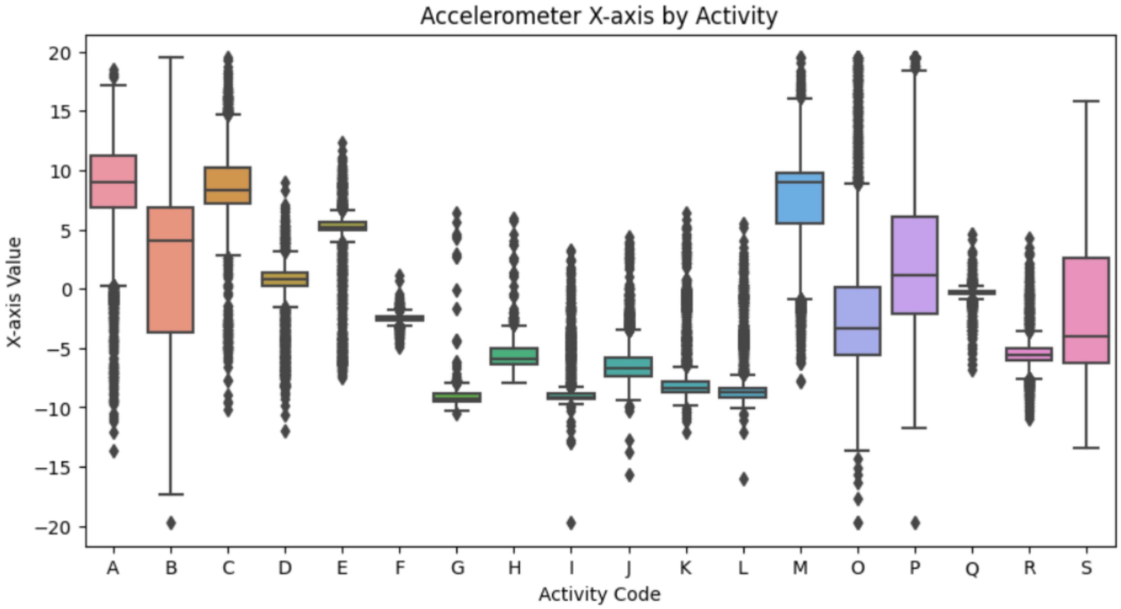




```
# --- Histogram of Accelerometer Axes ---
plt.figure(figsize=(10,5))
plt.hist(df_watch_accel["x"], bins=50, alpha=0.6, label="X")
plt.hist(df_watch_accel["y"], bins=50, alpha=0.6, label="Y")
plt.hist(df_watch_accel["z"], bins=50, alpha=0.6, label="Z")
plt.legend()
plt.title("Distribution of Accelerometer Values (X, Y, Z)")
plt.xlabel("Sensor Value")
plt.ylabel("Frequency")
plt.show()
```



```
# --- Boxplot: X-axis by Activity ---
plt.figure(figsize=(10,5))
sns.boxplot(x="activity_code", y="x", data=df_watch_accel)
plt.title("Accelerometer X-axis by Activity")
plt.xlabel("Activity Code")
plt.ylabel("X-axis Value")
plt.show()
```


Accelerometer X-axis by Activity
X-axis Value
A
B
C
D
E
F
G
H
I
J
K
L
M
O
P
Q
R
S
Activity Code