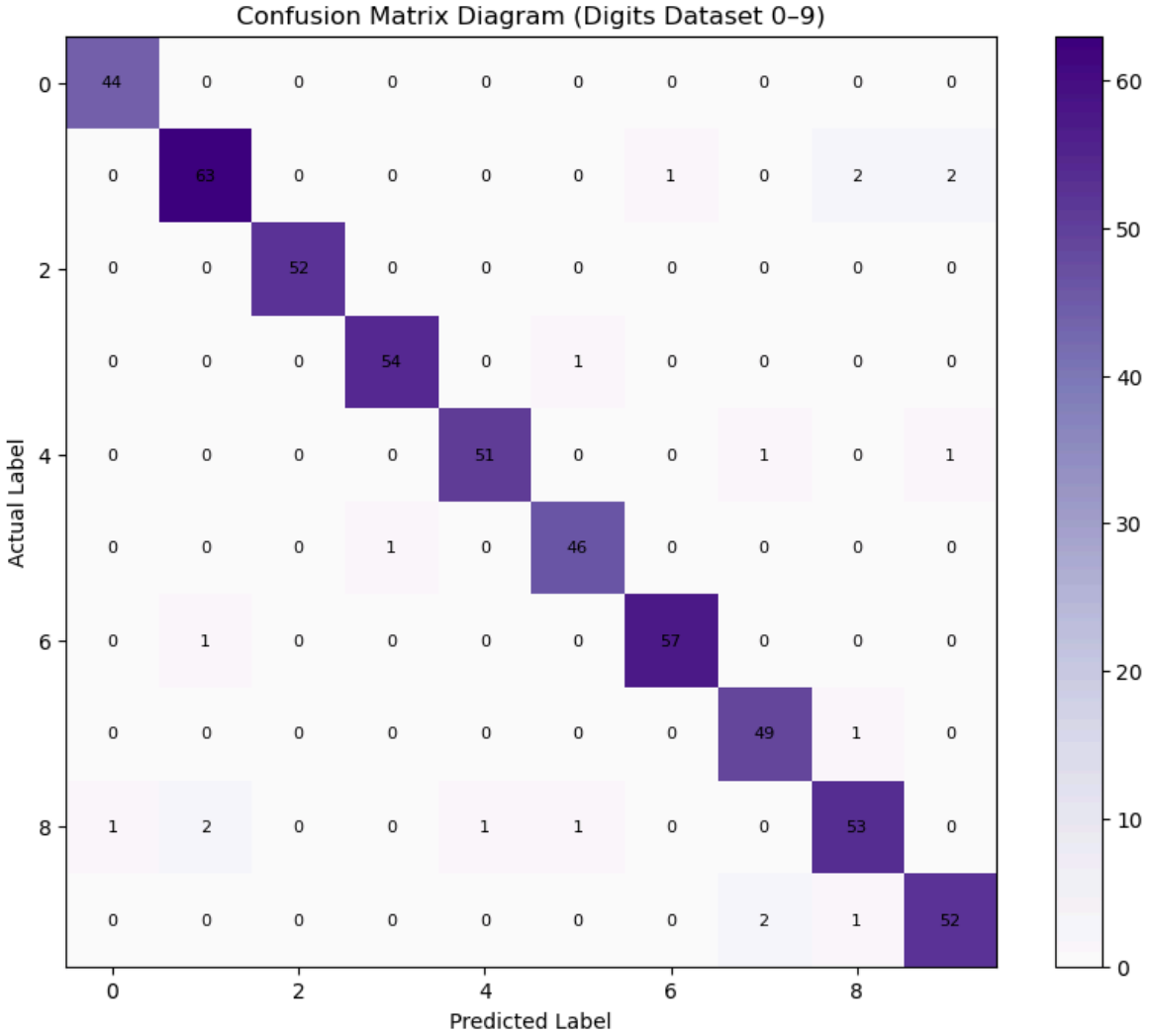
In [6]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score, precision_s
from sklearn.preprocessing import StandardScaler
digits = load_digits()
X = digits.data
y = digits.target
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=40
)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
model = LogisticRegression(max_iter=3000)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix (10-Class):")
print(cm)
plt.figure(figsize=(10,8))
plt.imshow(cm, cmap="Purples")
plt.title("Confusion Matrix Diagram (Digits Dataset 0–9)")
plt.xlabel("Predicted Label")
plt.ylabel("Actual Label")
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        plt.text(j, i, cm[i, j],
                 ha='center', va='center',
                 fontsize=8, color='black')
plt.colorbar()
plt.show()
accuracy  = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='macro')
recall    = recall_score(y_test, y_pred, average='macro')
f1        = f1_score(y_test, y_pred, average='macro')
print("\nAccuracy :", accuracy)
print("Precision:", precision)
print("Recall   :", recall)
print("F1 Score :", f1)
```

```
Confusion Matrix (10-Class):
[[44  0  0  0  0  0  0  0  0  0]
 [ 0 63  0  0  0  0  1  0  2  2]
 [ 0  0 52  0  0  0  0  0  0  0]
 [ 0  0  0 54  0  1  0  0  0  0]
 [ 0  0  0  0 51  0  0  1  0  1]
 [ 0  0  0  1  0 46  0  0  0  0]
 [ 0  1  0  0  0  0 57  0  0  0]
 [ 0  0  0  0  0  0  0 49  1  0]
 [ 1  2  0  0  1  1  0  0 53  0]
 [ 0  0  0  0  0  0  0  2  1 52]]
```




```
Accuracy : 0.9648148148148148
Precision: 0.9653589398099379
Recall   : 0.9671282594844668
F1 Score : 0.9661466968589313
```

In [ ]: