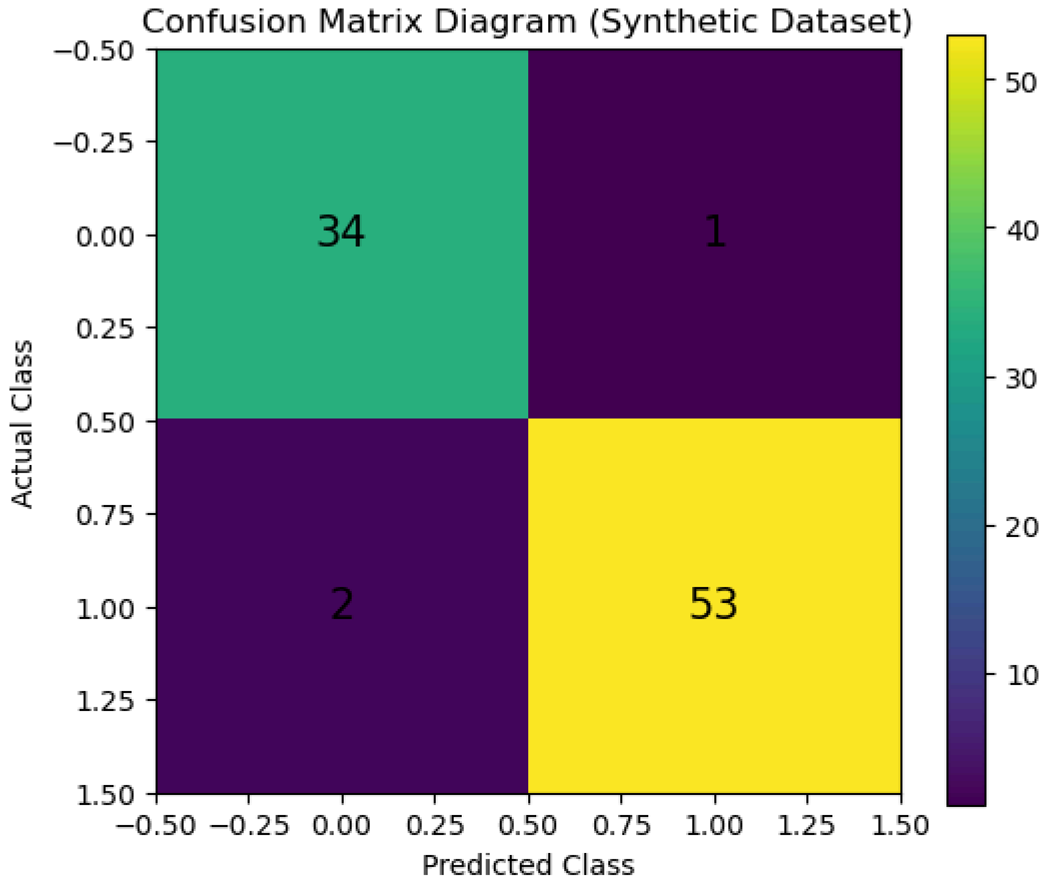
In [5]:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
X, y = make_classification(
    n_samples=300,
    n_features=2,
    n_informative=2,
    n_redundant=0,
    n_clusters_per_class=1,
    random_state=42,
    class_sep=1.5   # Increase class separation
)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=20
)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix:\n", cm)
plt.figure(figsize=(6,5))
plt.imshow(cm)
plt.title("Confusion Matrix Diagram (Synthetic Dataset)")
plt.xlabel("Predicted Class")
plt.ylabel("Actual Class")
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=15)
plt.colorbar()
plt.show()
print("\nAccuracy:", accuracy_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred))
print("Recall:", recall_score(y_test, y_pred))
print("F1 Score:", f1_score(y_test, y_pred))
```

```
Confusion Matrix:
 [[34  1]
 [ 2 53]]
```



```
Accuracy: 0.9666666666666667
Precision: 0.9814814814814815
Recall: 0.9636363636363636
F1 Score: 0.9724770642201835
```

```
In [ ]:
```