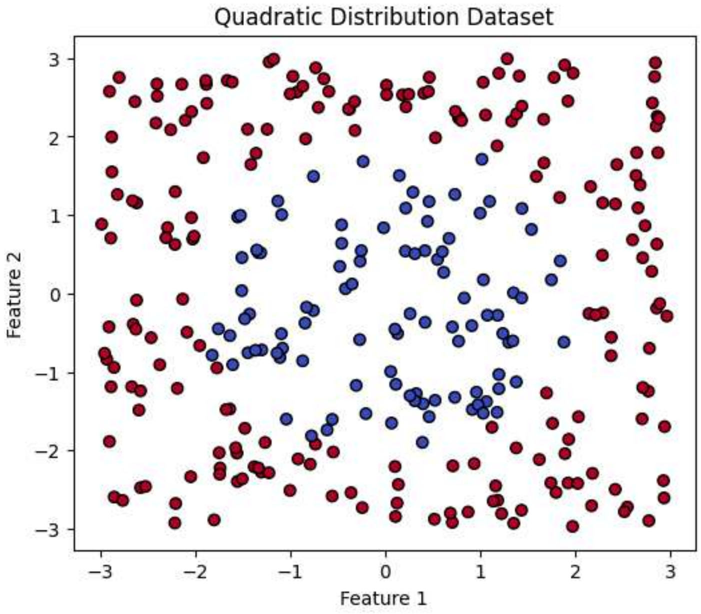
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
```

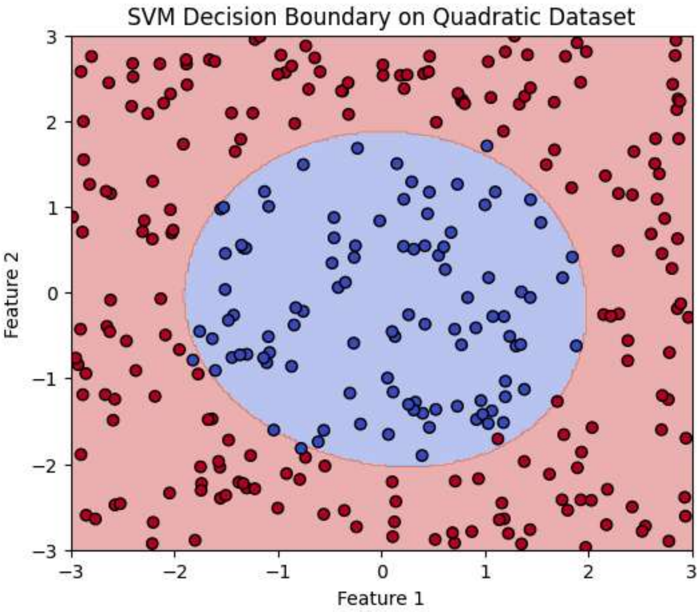
```
# Step 2: Create dataset points
np.random.seed(0)
n_samples = 300
X_quad = np.random.uniform(-3, 3, (n_samples, 2))


# Step 3: Create labels based on quadratic boundary (circle radius 2)
y_quad = (X_quad[:, 0]**2 + X_quad[:, 1]**2 > 4).astype(int)


# Step 4: Plot the quadratic dataset
plt.figure(figsize=(6,5))
plt.scatter(X_quad[:, 0], X_quad[:, 1], c=y_quad, cmap='coolwarm', edgecolor='k')
plt.title("Quadratic Distribution Dataset")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.show()
```



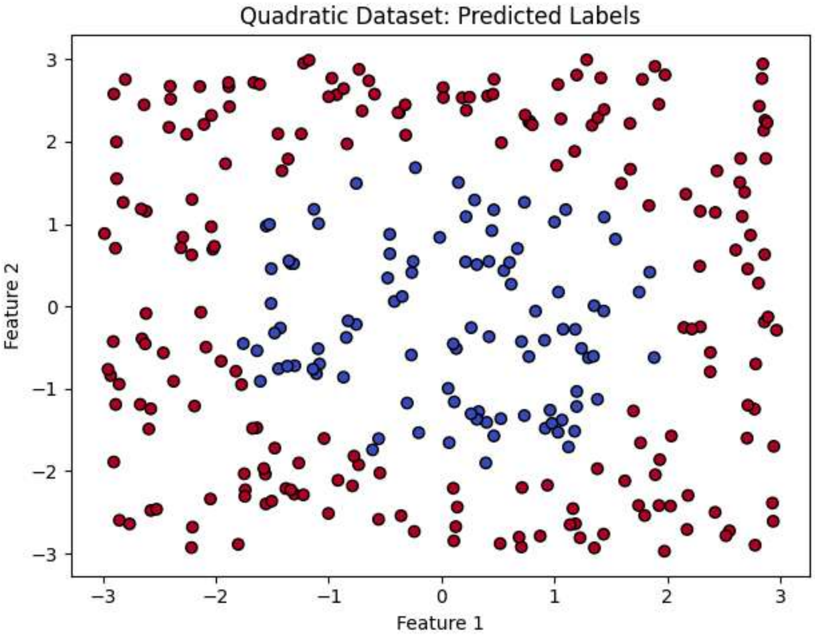
```
# Step 5: Train SVM with RBF kernel on quadratic data
svm_quad = SVC(kernel='rbf', gamma='scale')
svm_quad.fit(X_quad, y_quad)
```

```
▾ SVC
SVC()
```

```
# Step 6: Plot decision boundary
xx, yy = np.meshgrid(np.linspace(-3, 3, 300), np.linspace(-3, 3, 300))
Z = svm_quad.predict(np.c_[xx.ravel(), yy.ravel()]).reshape(xx.shape)



plt.figure(figsize=(6,5))
plt.contourf(xx, yy, Z, alpha=0.4, cmap='coolwarm')
plt.scatter(X_quad[:, 0], X_quad[:, 1], c=y_quad, cmap='coolwarm', edgecolor='k')
plt.title("SVM Decision Boundary on Quadratic Dataset")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.show()
```



```
y_quad_pred = svm_quad.predict(X_quad)



plt.figure(figsize=(12,5))




plt.subplot(1,2,2)
plt.scatter(X_quad[:, 0], X_quad[:, 1], c=y_quad_pred, cmap='coolwarm', edgecolor='k')
plt.title("Quadratic Dataset: Predicted Labels")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")

plt.tight_layout()
plt.show()
```



Start coding or generate with AI.