Code:

```python
from sklearn import datasets

from sklearn.model_selection import train_test_split

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report


# Load dataset
data = datasets.load_breast_cancer()


X = data.data

y = data.target


# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)


# Define kernels
kernels = ['linear', 'poly', 'rbf', 'sigmoid']


# Iterate over each kernel
for kernel in kernels:

    print("Kernel:", kernel)


    # SVM model with current kernel
    svm_model = SVC(kernel=kernel)

    svm_model.fit(X_train, y_train)

    y_pred = svm_model.predict(X_test)


    # Accuracy
    accuracy = accuracy_score(y_test, y_pred)

    print("Accuracy:", accuracy)
```

```python
    # Confusion Matrix
    confusion_mat = confusion_matrix(y_test, y_pred)
    print("Confusion Matrix:")
    print(confusion_mat)


    # Classification Report
    class_report = classification_report(y_test, y_pred)
    print("Classification Report:")
    print(class_report)


    print("------------------------\n")
```

Output:

Kernel: linear

Accuracy: 0.9649122807017544

Confusion Matrix:

[[59   4]

 [2 106]]

Classification Report:

	precision	recall	f1-score	support
0	0.97	0.94	0.95	63
1	0.96	0.98	0.97	108
accuracy			0.96	171
macro avg	0.97	0.96	0.96	171
weighted avg	0.96	0.96	0.96	171

Kernel: poly

Accuracy: 0.9415204678362573

```
Confusion Matrix:

[[ 54    9]

 [  1 107]]

Classification Report:

              precision    recall  f1-score   support

           0       0.98      0.86      0.92        63

           1       0.92      0.99      0.96       108


    accuracy                           0.94       171

   macro avg       0.95      0.92      0.94       171

weighted avg       0.94      0.94      0.94       171



------------------------


Kernel: rbf

Accuracy: 0.935672514619883

Confusion Matrix:

[[ 52   11]

 [  0 108]]

Classification Report:

              precision    recall  f1-score   support

           0       1.00      0.83      0.90        63

           1       0.91      1.00      0.95       108


    accuracy                           0.94       171

   macro avg       0.95      0.91      0.93       171

weighted avg       0.94      0.94      0.93       171



------------------------


Kernel: sigmoid
```

```
Accuracy: 0.4678362573099415
Confusion Matrix:
[[ 8 55]
 [36 72]]
Classification Report:
              precision    recall  f1-score   support

           0       0.18      0.13      0.15        63
           1       0.57      0.67      0.61       108

    accuracy                           0.47       171
   macro avg       0.37      0.40      0.38       171
weighted avg       0.43      0.47      0.44       171
```

```python
CODE:
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

# Load the dataset
data = pd.read_csv("../content/UniversalBank (1).csv")

# Drop irrelevant columns if necessary and split features and target
variable
X = data.drop(['ID', 'ZIP Code', 'Personal Loan'], axis=1)
y = data['Personal Loan']

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Define kernels
kernels = ['linear', 'poly', 'rbf', 'sigmoid']

# Iterate over each kernel
for kernel in kernels:

    print("Kernel:", kernel)
```

```python
    # SVM model with current kernel
    svm_model = SVC(kernel=kernel)
    svm_model.fit(X_train, y_train)

    y_pred = svm_model.predict(X_test)

    # Accuracy
    accuracy = accuracy_score(y_test, y_pred)
    print("Accuracy:", accuracy)

    # Confusion Matrix
    confusion_mat = confusion_matrix(y_test, y_pred)
    print("Confusion Matrix:")
    print(confusion_mat)

    # Classification Report
    class_report = classification_report(y_test, y_pred)
    print("Classification Report:")
    print(class_report)
    print("------------------------\n")
```

```
OUTPUT:
Kernel: linear
Accuracy: 0.9473333333333334
Confusion Matrix:
[[1334    9]
 [  70   87]]
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.99      0.97      1343
           1       0.91      0.55      0.69       157

    accuracy                           0.95      1500
   macro avg       0.93      0.77      0.83      1500
weighted avg       0.95      0.95      0.94      1500


------------------------

Kernel: poly
Accuracy: 0.904
Confusion Matrix:
[[1343    0]
 [ 144   13]]
Classification Report:
              precision    recall  f1-score   support

           0       0.90      1.00      0.95      1343
           1       1.00      0.08      0.15       157
```

```
      accuracy                              0.90      1500
     macro avg       0.95      0.54        0.55      1500
  weighted avg       0.91      0.90        0.87      1500


------------------------

Kernel: rbf
Accuracy: 0.8993333333333333
Confusion Matrix:
[[1336    7]
 [ 144   13]]
Classification Report:
               precision    recall  f1-score   support

           0       0.90      0.99      0.95      1343
           1       0.65      0.08      0.15       157

    accuracy                           0.90      1500
   macro avg       0.78      0.54      0.55      1500
weighted avg       0.88      0.90      0.86      1500


------------------------

Kernel: sigmoid
Accuracy: 0.8626666666666667
Confusion Matrix:
[[1244   99]
 [ 107   50]]
Classification Report:
               precision    recall  f1-score   support

           0       0.92      0.93      0.92      1343
           1       0.34      0.32      0.33       157

    accuracy                           0.86      1500
   macro avg       0.63      0.62      0.63      1500
weighted avg       0.86      0.86      0.86      1500


------------------------
```