```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import  make_regression
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC, SVR
from sklearn.metrics import accuracy_score, precision_score,
confusion_matrix, classification_report, mean_squared_error

from sklearn.datasets import load_wine

wine = load_wine()
X_class = wine.data
y_class = wine.target

# Split the data into training and test sets
X_train_c, X_test_c, y_train_c, y_test_c = train_test_split(
    X_class, y_class, test_size=0.3, random_state=42
)

kernels = ['linear', 'poly', 'rbf']

for kernel in kernels:
    print(f"\nKernel: {kernel}")
    clf = SVC(kernel=kernel)
    clf.fit(X_train_c, y_train_c)
    y_pred_c = clf.predict(X_test_c)

    acc = accuracy_score(y_test_c, y_pred_c)
    prec = precision_score(y_test_c, y_pred_c, average='macro')
    report = classification_report(y_test_c, y_pred_c)

    print("Accuracy:", acc)
    print("Precision (macro avg):", prec)
    print("Classification Report:\n", report)
```

```
=== Support Vector Classification ===

Kernel: linear
Accuracy: 0.9814814814814815
Precision (macro avg): 0.9777777777777779
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       1.00      0.95      0.98        21
           2       0.93      1.00      0.97        14

    accuracy                           0.98        54
   macro avg       0.98      0.98      0.98        54
weighted avg       0.98      0.98      0.98        54
```

```
Kernel: poly
Accuracy: 0.7592592592592593
Precision (macro avg): 0.7539426523297491
Classification Report:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        19
           1       0.65      0.95      0.77        21
           2       0.67      0.14      0.24        14

    accuracy                           0.76        54
   macro avg       0.75      0.70      0.66        54
weighted avg       0.76      0.76      0.70        54


Kernel: rbf
Accuracy: 0.7592592592592593
Precision (macro avg): 0.7444444444444445
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       0.63      0.90      0.75        21
           2       0.60      0.21      0.32        14

    accuracy                           0.76        54
   macro avg       0.74      0.71      0.69        54
weighted avg       0.75      0.76      0.72        54
```

```python
# Generate synthetic regression data
X_reg, y_reg = make_regression(n_samples=100, n_features=1, noise=10,
random_state=42)

X_train_r, X_test_r, y_train_r, y_test_r = train_test_split(X_reg,
y_reg, test_size=0.3, random_state=42)

print("\n=== Support Vector Regression ===")
for kernel in kernels:
    print(f"\nKernel: {kernel}")
    svr = SVR(kernel=kernel)
    svr.fit(X_train_r, y_train_r)
    y_pred_r = svr.predict(X_test_r)

    rmse = np.sqrt(mean_squared_error(y_test_r, y_pred_r))
    print("Root Mean Squared Error (RMSE):", rmse)
```

```
=== Support Vector Regression ===
```

```
Kernel: linear
Root Mean Squared Error (RMSE): 14.563268157855786

Kernel: poly
Root Mean Squared Error (RMSE): 26.128547435628096

Kernel: rbf
Root Mean Squared Error (RMSE): 32.98371077931491
```

```
plt.scatter(X_test_r, y_test_r, color='blue', label='Actual')
plt.scatter(X_test_r, y_pred_r, color='red', label='Predicted')
plt.title(f'SVR with {kernel} kernel')
plt.legend()
plt.xlabel('Feature')
plt.ylabel('Target')
plt.show()
```