Support Vector Machine Demo

Shubhan Tamhane

Table of contents

Importing necessary libraries

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score, precision_score, confusion_mat
import matplotlib.pyplot as plt
from sklearn import datasets
```

Importing libraries for accessing MNIST data

```
import ssl
import certifi

ssl._create_default_https_context = lambda: ssl.create_default_context(cafile=certifi.where())
```

```
from sklearn.datasets import fetch_openml
mnist = fetch_openml('mnist_784', version=1, as_frame=False)
```

Predicting whether a number is a 4 or a 9. Set test size to 20%.

```
X, y = mnist['data'], mnist['target'].astype(int)

mask = np.isin(y, [4,9])
X, y = X[mask], y[mask]

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=3255, test_size=0.2)
```

Fit Support Vector Classifier model on data, using a Radius Basis Function Kernel, gamma set to scale, and C set to 1.

```
m1 = SVC(kernel = 'rbf', gamma='scale', C =1)
m1.fit(X_test, y_test)
y_pred = m1.predict(X_test)
```

```
accuracy_score = accuracy_score(y_test, y_pred)
precision_score = precision_score(y_test, y_pred, pos_label=9)
f1_score = f1_score(y_test, y_pred, pos_label=9)
confusion_matrix = confusion_matrix(y_test, y_pred)
```

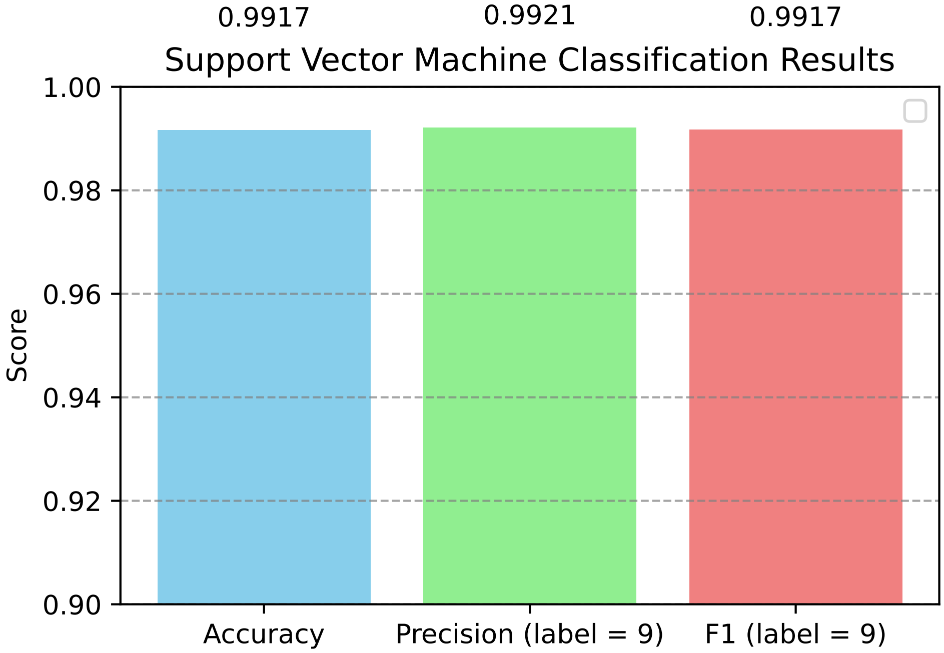
Model evaluation.

```
print("Accuracy Score:", accuracy_score)
print("Precision Score:", precision_score)
print("F1 Score:", f1_score)
print("Confusion Matrix:", confusion_matrix)
```

```
Accuracy Score: 0.9916575988393181
Precision Score: 0.9921033740129217
F1 Score: 0.9917473986365267
Confusion Matrix: [[1352   11]
 [  12 1382]]
```

```
metrics = ['Accuracy', 'Precision (label = 9)', 'F1 (label = 9)']
scores = [accuracy_score, precision_score, f1_score]

colors = ['skyblue', 'lightgreen', 'lightcoral']

plt.bar(metrics, scores, color = colors)
plt.title("Support Vector Machine Classification Results")
plt.ylabel('Score')
plt.ylim(0.9,1)
plt.grid(axis = 'y', color = 'grey', linestyle = '--', alpha = 0.7)

for i, score in enumerate(scores):
  plt.text(i, score+.02, f'{score:.4f}', ha='center')

plt.legend()
plt.show()
```

```
/var/folders/c3/2w85_lw50qdgkzcnmbnz057m0000gn/T/ipykernel_73155/2674982142.py:15: UserWarning:

No artists with labels found to put in legend.  Note that artists whose label start with an under
```


0.9917
0.9921
0.9917
Support Vector Machine Classification Results
Score
1.00
0.98
0.96
0.94
0.92
0.90
Accuracy
Precision (label = 9)
F1 (label = 9)