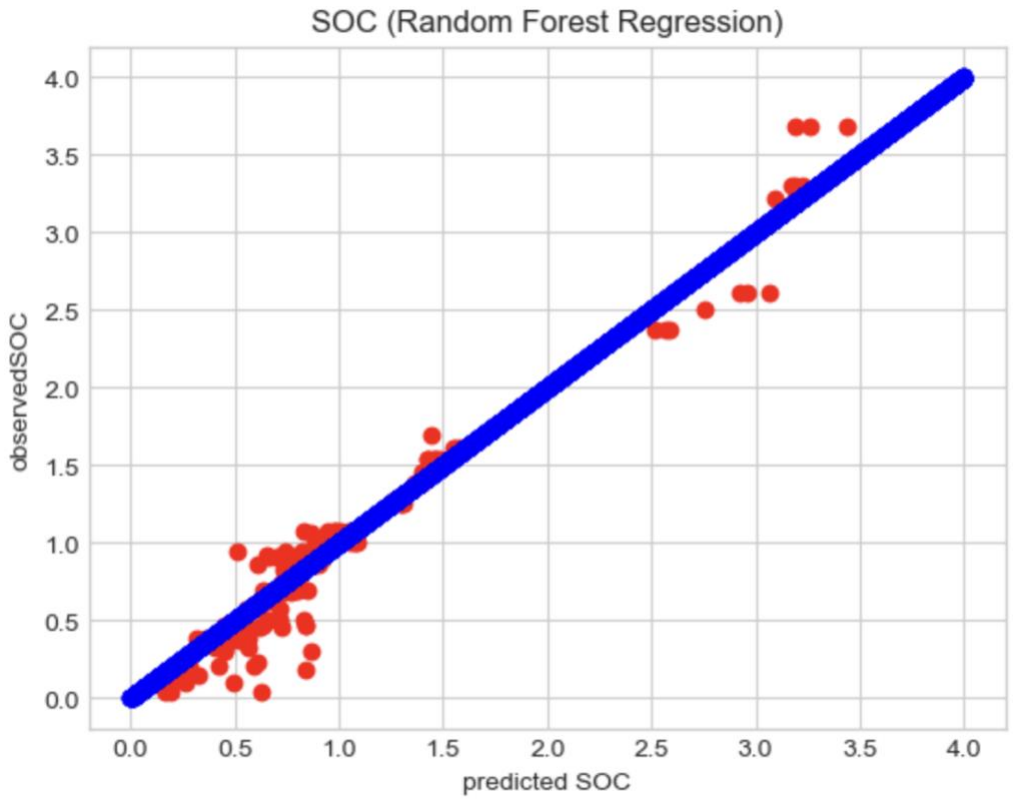
1. RF 20m – S2A, SWHC, ST

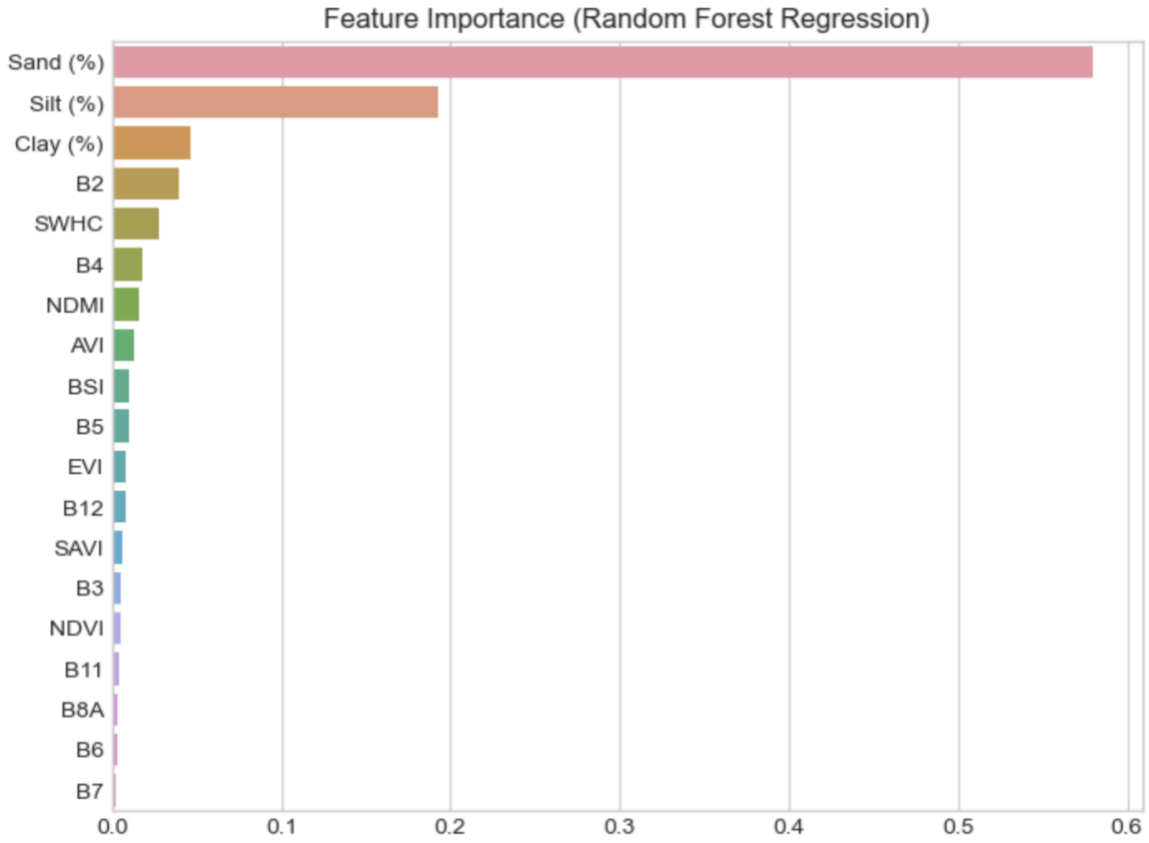


In [16]:
```
##########모델 검증
# R-squared

print(model.score(x_train, y_train)) #

print(model.score(x_test, y_test)) #
```

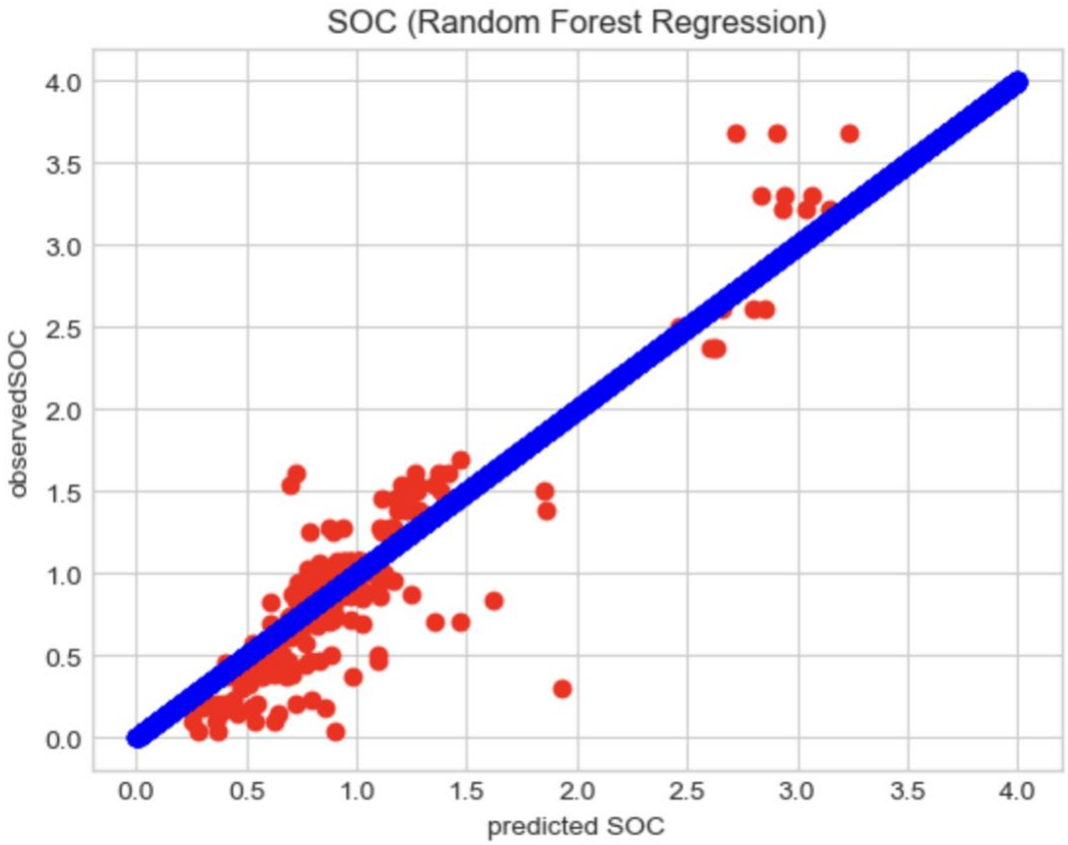
```
0.9840598360435518
0.8792249798724048
```

In [17]:
```
# R-squared

model.score(x.values,y.values)
```

Out[17]:
```
0.9555695460570376
```


Feature Importance (Random Forest Regression)
Sand (%)
Silt (%)
Clay (%)
B2
SWHC
B4
NDMI
AVI
BSI
B5
EVI
B12
SAVI
B3
NDVI
B11
B8A
B6
B7
0.0
0.1
0.2
0.3
0.4
0.5
0.6

2. RF 20m – S2A

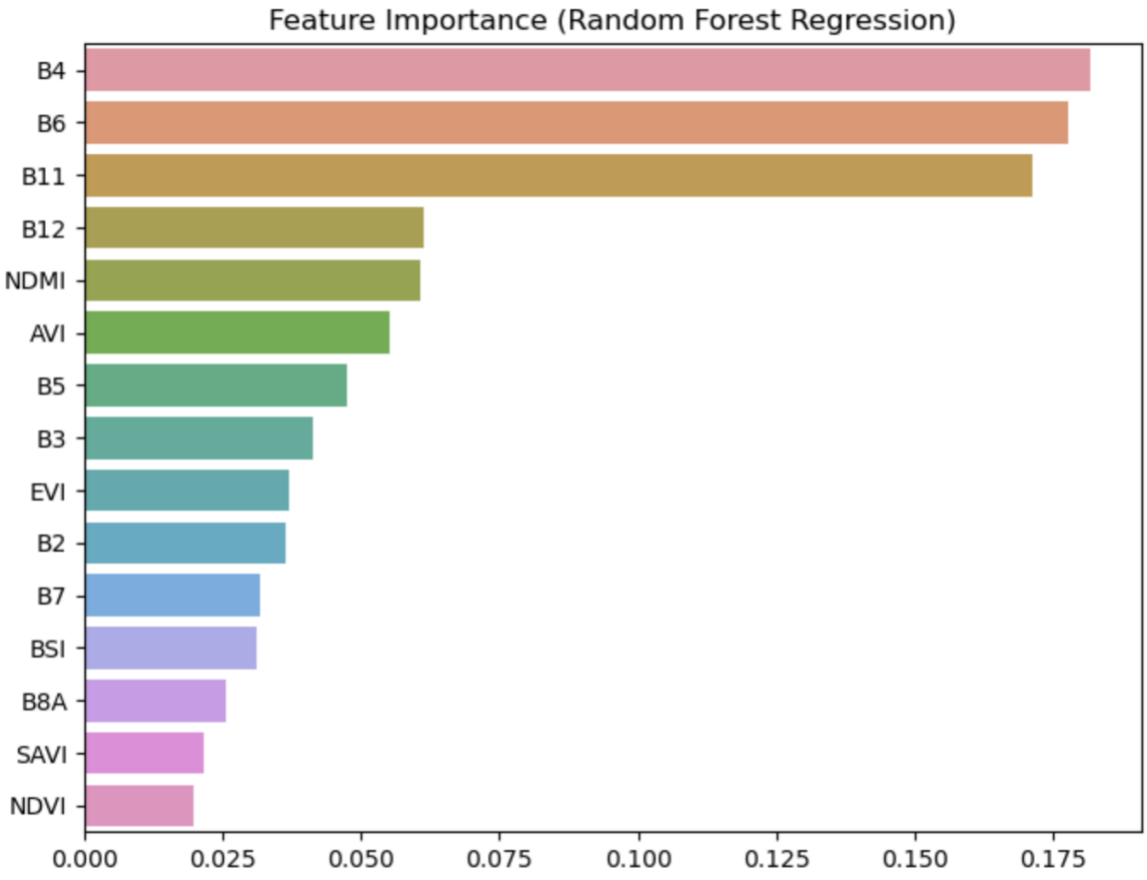


In [52]:
```
##########모델 검증
# R-squared

print(model.score(x_train, y_train)) #

print(model.score(x_test, y_test)) #
```

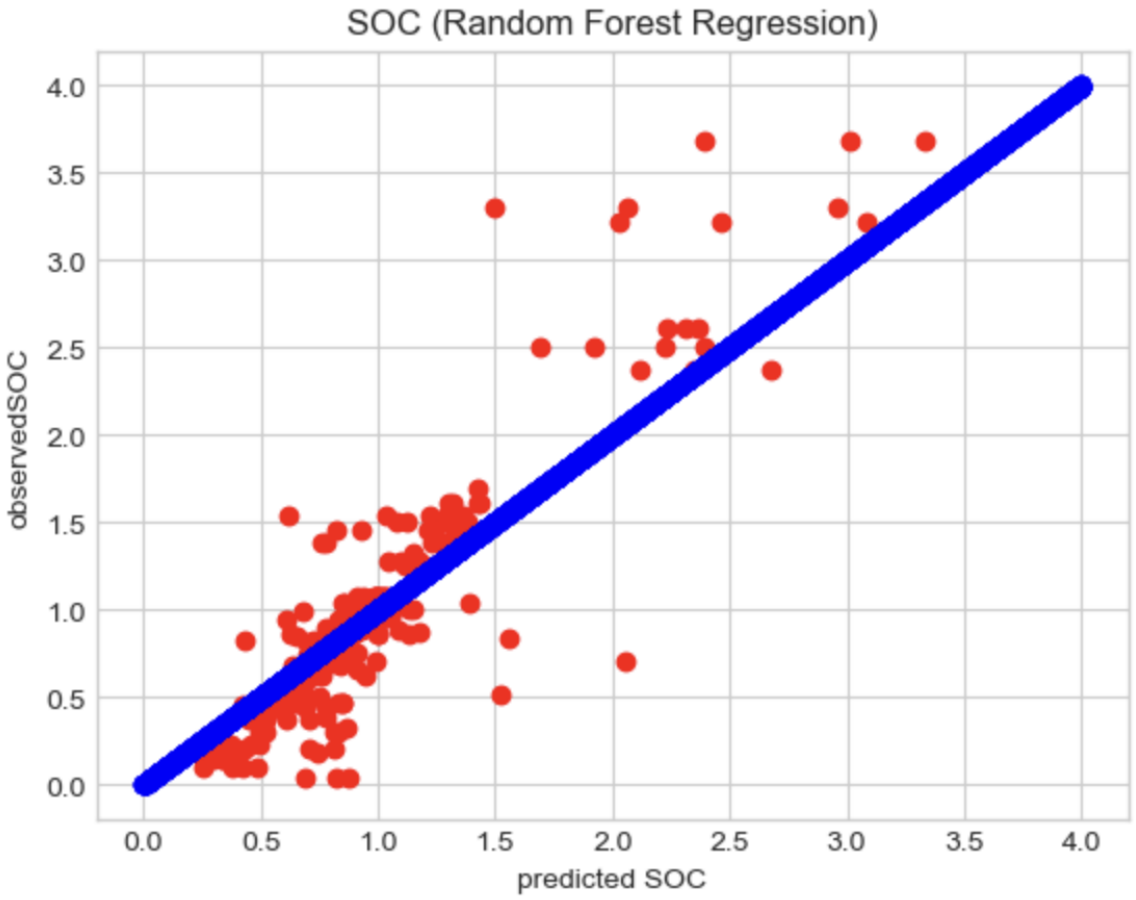
```
0.9242057852726683
0.6057669014964413
```

In [53]:
```
# R-squared

model.score(x.values,y.values)
```

Out[53]:
```
0.8376773911089964
```


Feature Importance (Random Forest Regression)
B4
B6
B11
B12
NDMI
AVI
B5
B3
EVI
B2
B7
BSI
B8A
SAVI
NDVI
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175

3. RF 10m – S2A

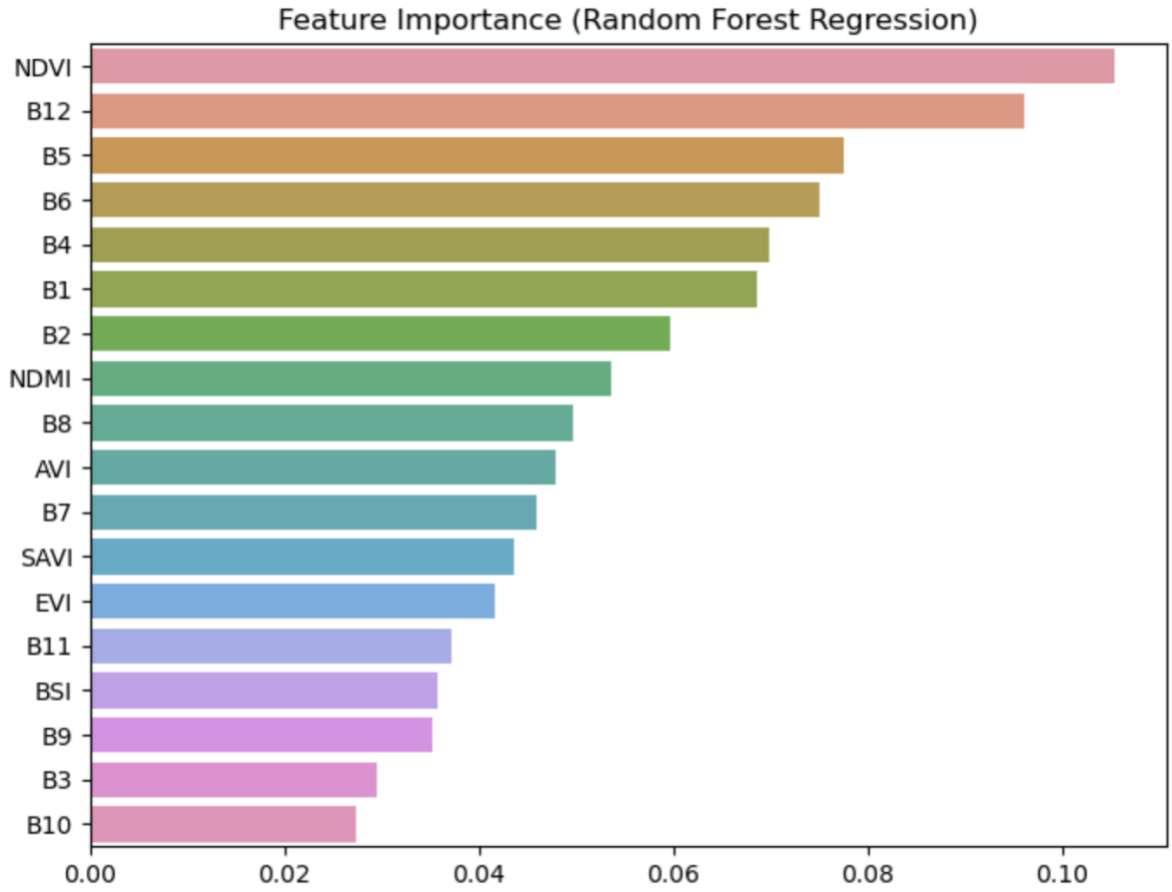


In [17]:
```
##########모델 검증
# R-squared

print(model.score(x_train, y_train)) #

print(model.score(x_test, y_test)) #
```

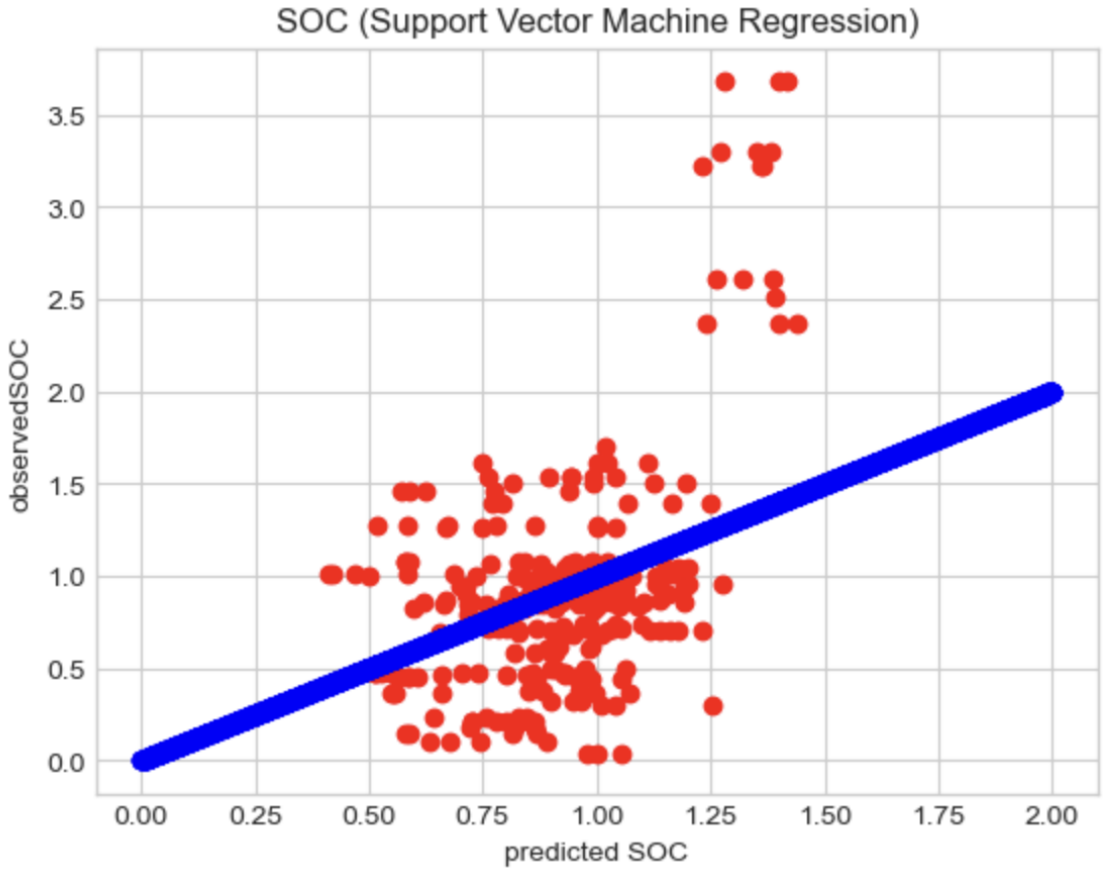
```
0.9170525397681053
0.5685443701601358
```

In [18]:
```
# R-squared

model.score(x.values,y.values)
```

Out[18]:
```
0.7914683627769743
```


Feature Importance (Random Forest Regression)
NDVI
B12
B5
B6
B4
B1
B2
NDMI
B8
AVI
B7
SAVI
EVI
B11
BSI
B9
B3
B10
0.00
0.02
0.04
0.06
0.08
0.10

4. SVM 20m – S2A

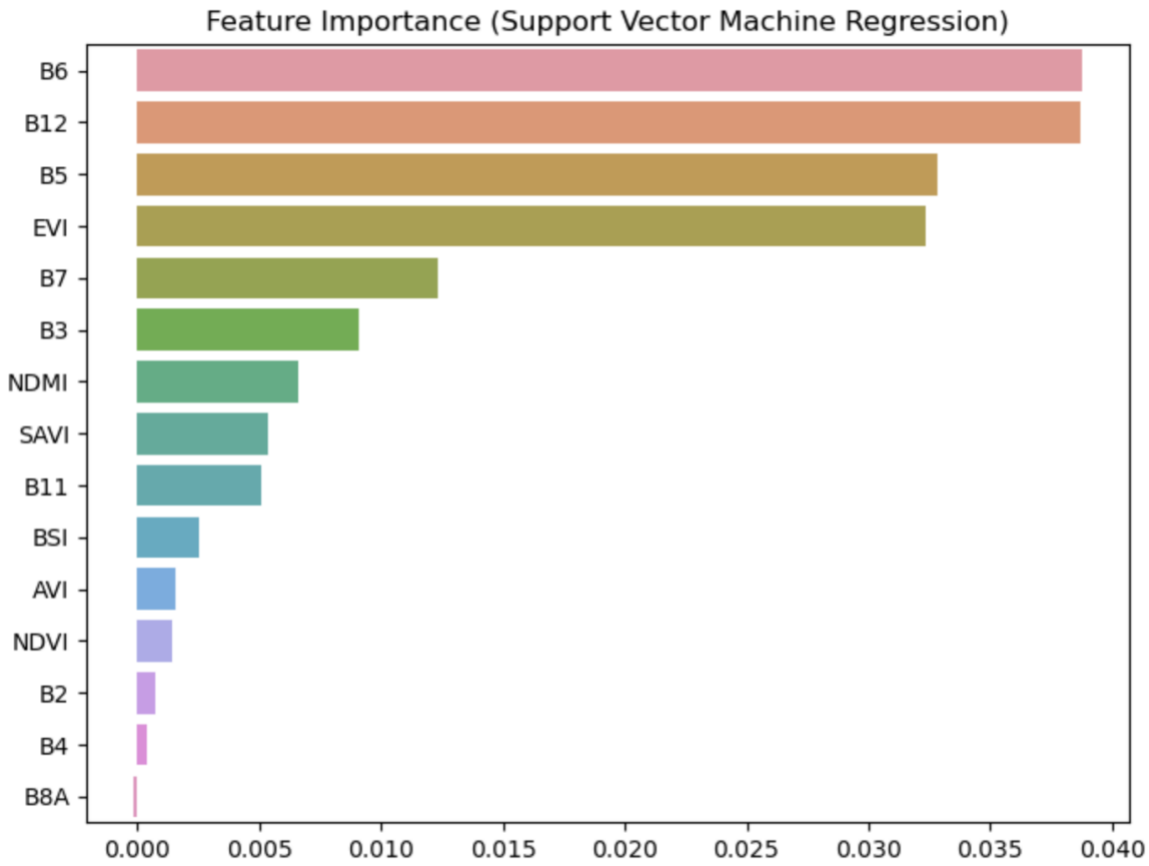



```
In [28]: ##########모델 검증

         print(model.score(x_train, y_train)) #

         print(model.score(x_test, y_test)) #
```

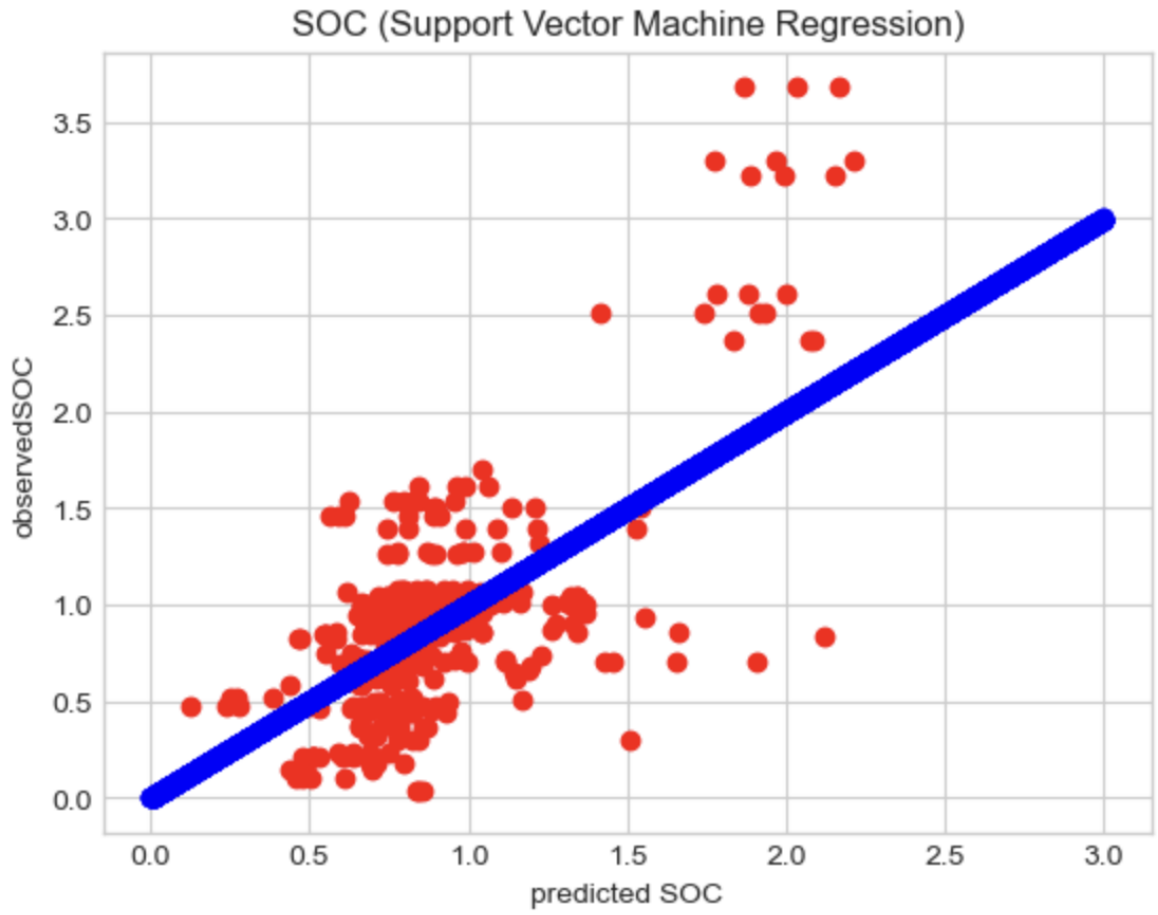
```
0.21828143851075743
0.21173982749234543
```

```
In [29]: # R-squared

         model.score(x.values,y.values)
```

```
Out[29]: 0.2181592290334443
```


Feature Importance (Support Vector Machine Regression)
B6
B12
B5
EVI
B7
B3
NDMI
SAVI
B11
BSI
AVI
NDVI
B2
B4
B8A
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040

5. SVM 10m – S2A

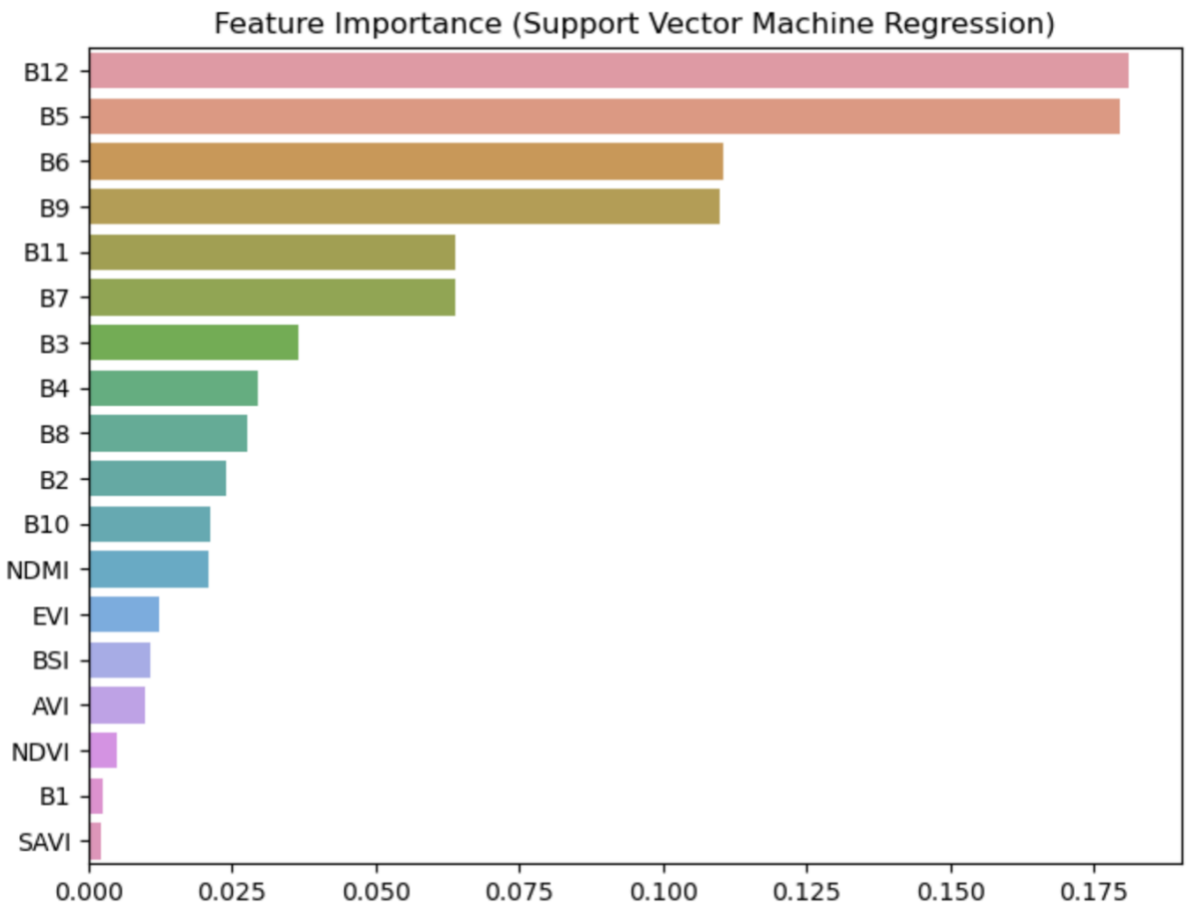


In [21]:
```
##########모델 검증

print(model.score(x_train, y_train)) #

print(model.score(x_test, y_test)) #
```

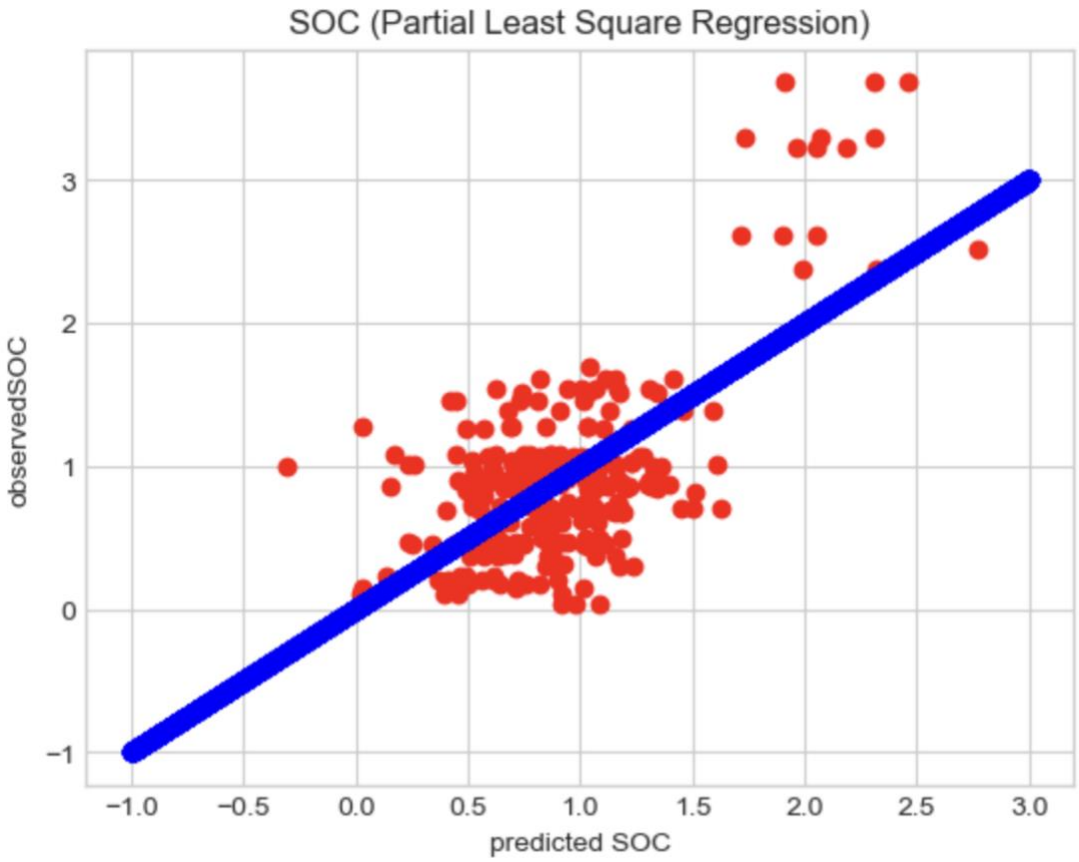
```
0.5012814508773755
0.477561219778905
```

In [22]:
```
# R-squared

model.score(x.values,y.values)
```

Out[22]:
```
0.4943502560590146
```


Feature Importance (Support Vector Machine Regression)
B12
B5
B6
B9
B11
B7
B3
B4
B8
B2
B10
NDMI
EVI
BSI
AVI
NDVI
B1
SAVI
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175

6. PLSR 20m – S2A

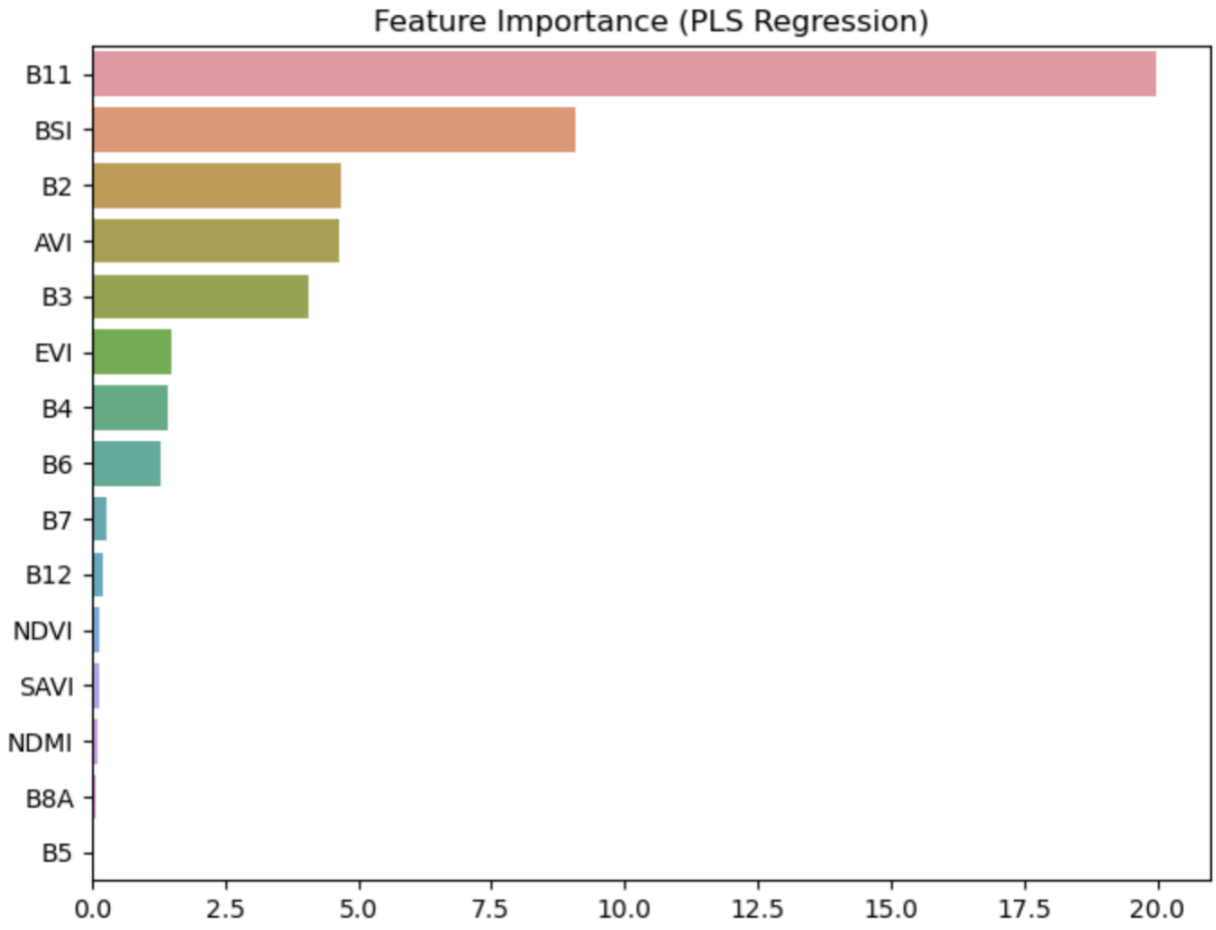


In [27]:
```
##########모델 검증

print(model.score(x_train, y_train)) #

print(model.score(x_test, y_test)) #
```

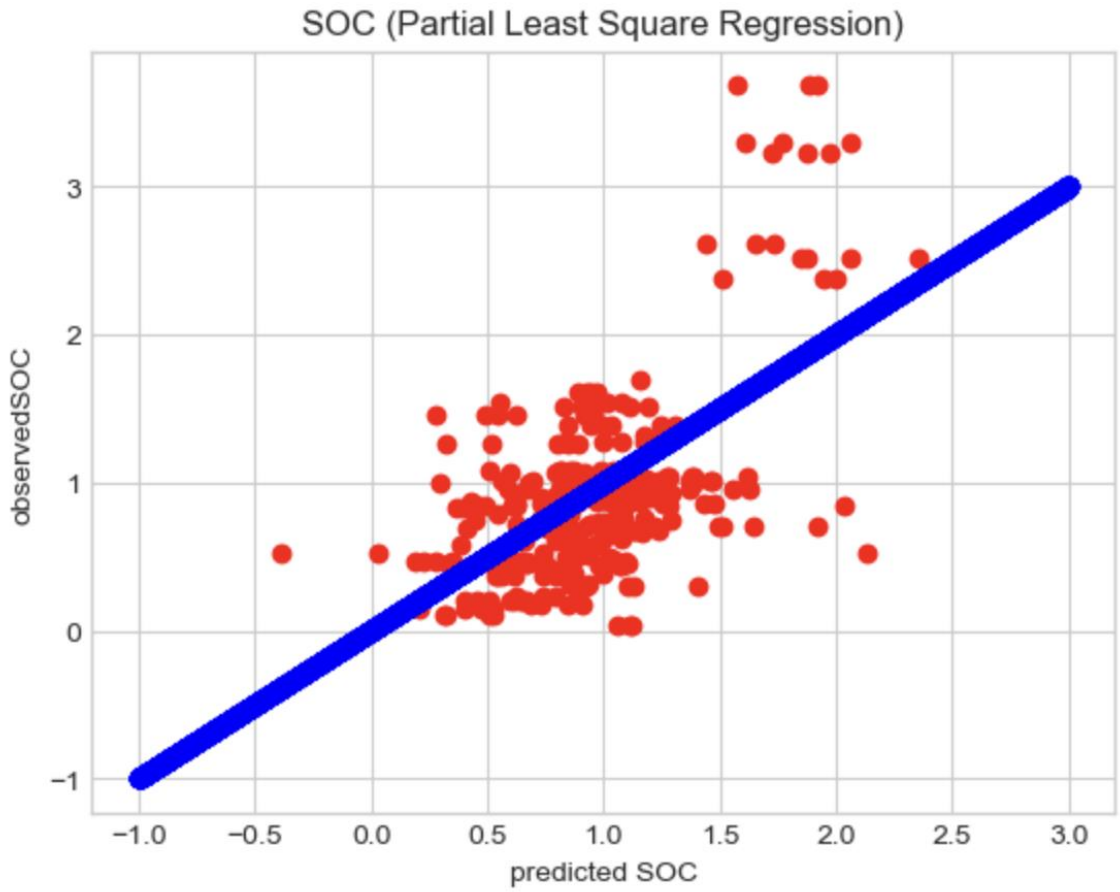
```
0.4773710491083185
0.2965150117922244
```

In [28]:
```
# R-squared

model.score(x.values,y.values)
```

Out[28]:
```
0.43852002723321537
```


Feature Importance (PLS Regression)
B11
BSI
B2
AVI
B3
EVI
B4
B6
B7
B12
NDVI
SAVI
NDMI
B8A
B5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

7. PLSR 10m – S2A

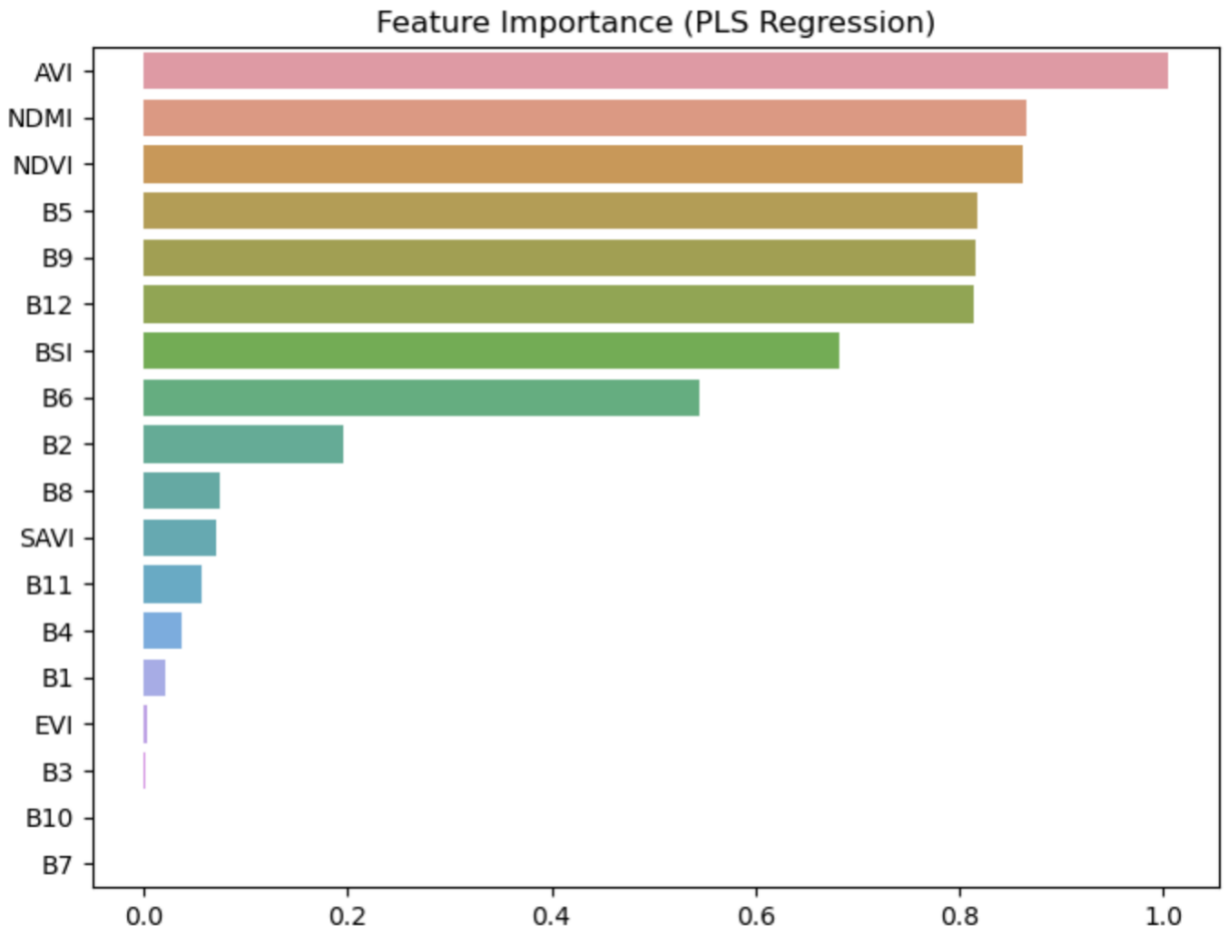


In [45]:
```
##########모델 검증

print(model.score(x_train, y_train)) #

print(model.score(x_test, y_test)) #
```

```
0.3771753932542794
0.25154658044795175
```

In [47]:
```
# R-squared

model.score(x.values,y.values)
```

Out[47]:
```
0.35297288763133783
```


Feature Importance (PLS Regression)
AVI
NDMI
NDVI
B5
B9
B12
BSI
B6
B2
B8
SAVI
B11
B4
B1
EVI
B3
B10
B7
0.0
0.2
0.4
0.6
0.8
1.0

8. ANN 20m – S2A

In [20]:
```
# 테스트셋을 통한 모델 평가
model.evaluate(normed_test_data, pd.DataFrame(y_test), verbose=2)
# verbose : 결과 출력의 단계 설정
# auto - 대부분 1로 지정됨
# 0 - 출력 없음
# 1 - 진행 상황 출력(프로그레스바 포함)
# 2 - 진행 상황 출력(프로그레스바 제외, 1에 비해 간소화)
```

```
3/3 - 0s - loss: 0.2684 - accuracy: 0.0714 - 160ms/epoch - 53ms/ste
```

```
2022-11-08 19:13:42.692184: I tensorflow/core/grappler/optimizers/d
mizer for device_type GPU is enabled.
```

Out[20]:
```
[0.26841333508491516, 0.0714285746216774]
```