Francesco Michael Kusuma

13522038

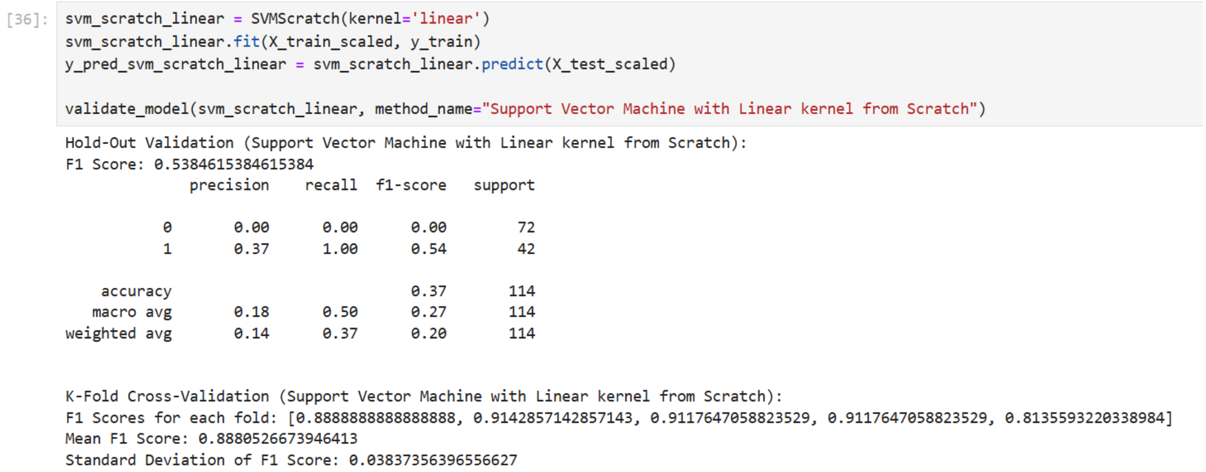
Support Vector Machine

Cara Kerja:

1. Pilih kernel untuk memetakan data ke dimensi yang lebih tinggi, seperti linear, polynomial, RBF, sigmoid, dan sebagainya.
2. Cari hyperplane yang memaksimalkan margin antara dua kelas.
3. Klasifikasikan data baru berdasarkan posisi relatif terhadap hyperplane.

Perbandingan:

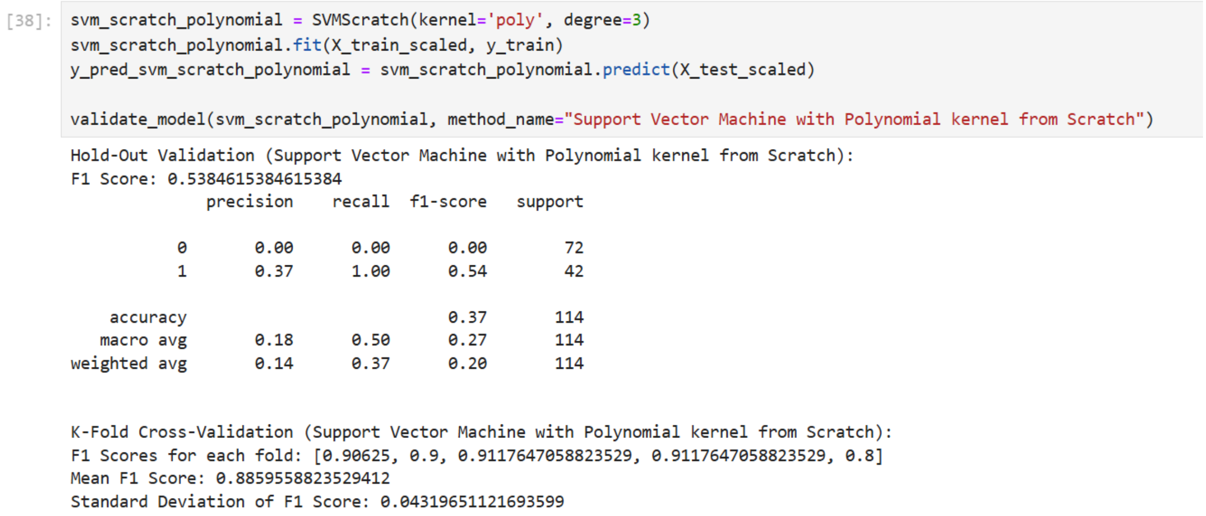
- *Scratch*
  - *Linear Kernel*



```
[36]: svm_scratch_linear = SVMScratch(kernel='linear')
      svm_scratch_linear.fit(X_train_scaled, y_train)
      y_pred_svm_scratch_linear = svm_scratch_linear.predict(X_test_scaled)

      validate_model(svm_scratch_linear, method_name="Support Vector Machine with Linear kernel from Scratch")
```

```
Hold-Out Validation (Support Vector Machine with Linear kernel from Scratch):
F1 Score: 0.5384615384615384
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        72
           1       0.37      1.00      0.54        42

    accuracy                           0.37       114
   macro avg       0.18      0.50      0.27       114
weighted avg       0.14      0.37      0.20       114


K-Fold Cross-Validation (Support Vector Machine with Linear kernel from Scratch):
F1 Scores for each fold: [0.8888888888888888, 0.9142857142857143, 0.9117647058823529, 0.9117647058823529, 0.8135593220338984]
Mean F1 Score: 0.8880526673946413
Standard Deviation of F1 Score: 0.03837356396556627
```

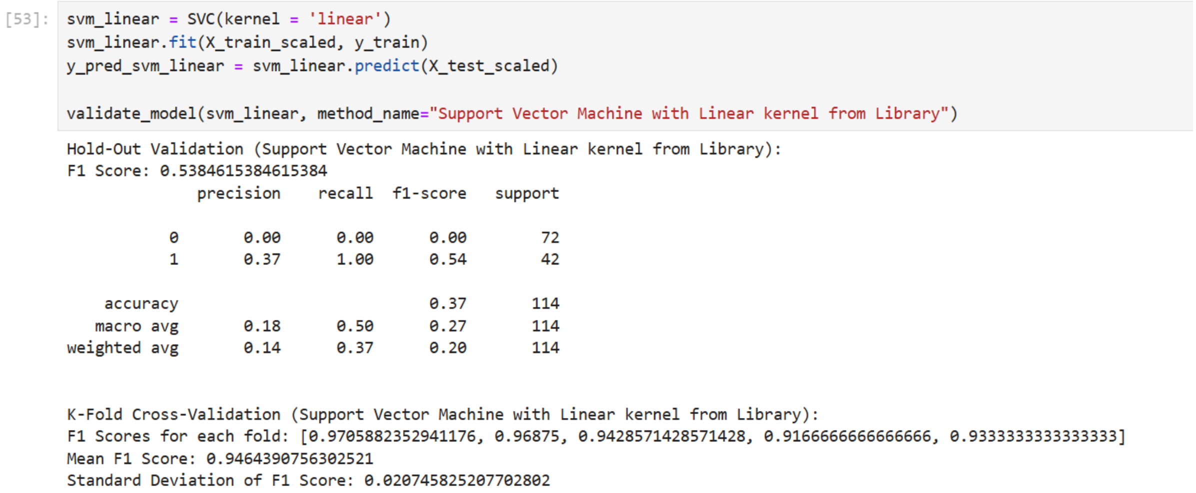
  - *Polynomial Kernel*



```
[38]: svm_scratch_polynomial = SVMScratch(kernel='poly', degree=3)
      svm_scratch_polynomial.fit(X_train_scaled, y_train)
      y_pred_svm_scratch_polynomial = svm_scratch_polynomial.predict(X_test_scaled)

      validate_model(svm_scratch_polynomial, method_name="Support Vector Machine with Polynomial kernel from Scratch")
```

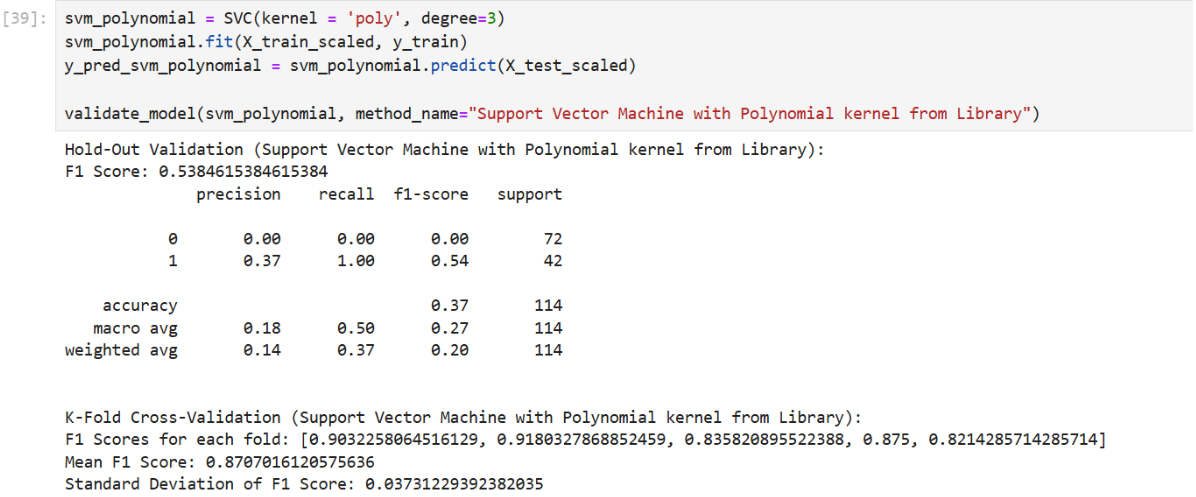
```
Hold-Out Validation (Support Vector Machine with Polynomial kernel from Scratch):
F1 Score: 0.5384615384615384
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        72
           1       0.37      1.00      0.54        42

    accuracy                           0.37       114
   macro avg       0.18      0.50      0.27       114
weighted avg       0.14      0.37      0.20       114


K-Fold Cross-Validation (Support Vector Machine with Polynomial kernel from Scratch):
F1 Scores for each fold: [0.90625, 0.9, 0.9117647058823529, 0.9117647058823529, 0.8]
Mean F1 Score: 0.8859558823529412
Standard Deviation of F1 Score: 0.04319651121693599
```

- *Library*
  - *Linear Kernel*



```
[53]: svm_linear = SVC(kernel = 'linear')
      svm_linear.fit(X_train_scaled, y_train)
      y_pred_svm_linear = svm_linear.predict(X_test_scaled)

      validate_model(svm_linear, method_name="Support Vector Machine with Linear kernel from Library")

      Hold-Out Validation (Support Vector Machine with Linear kernel from Library):
      F1 Score: 0.5384615384615384
                    precision    recall  f1-score   support

                 0       0.00      0.00      0.00        72
                 1       0.37      1.00      0.54        42

          accuracy                           0.37       114
         macro avg       0.18      0.50      0.27       114
      weighted avg       0.14      0.37      0.20       114


      K-Fold Cross-Validation (Support Vector Machine with Linear kernel from Library):
      F1 Scores for each fold: [0.9705882352941176, 0.96875, 0.9428571428571428, 0.9166666666666666, 0.9333333333333333]
      Mean F1 Score: 0.9464390756302521
      Standard Deviation of F1 Score: 0.020745825207702802
```

  - *Polynomial Kernel*



```
[39]: svm_polynomial = SVC(kernel = 'poly', degree=3)
      svm_polynomial.fit(X_train_scaled, y_train)
      y_pred_svm_polynomial = svm_polynomial.predict(X_test_scaled)

      validate_model(svm_polynomial, method_name="Support Vector Machine with Polynomial kernel from Library")

      Hold-Out Validation (Support Vector Machine with Polynomial kernel from Library):
      F1 Score: 0.5384615384615384
                    precision    recall  f1-score   support

                 0       0.00      0.00      0.00        72
                 1       0.37      1.00      0.54        42

          accuracy                           0.37       114
         macro avg       0.18      0.50      0.27       114
      weighted avg       0.14      0.37      0.20       114


      K-Fold Cross-Validation (Support Vector Machine with Polynomial kernel from Library):
      F1 Scores for each fold: [0.9032258064516129, 0.9180327868852459, 0.835820895522388, 0.875, 0.8214285714285714]
      Mean F1 Score: 0.8707016120575636
      Standard Deviation of F1 Score: 0.03731229392382035
```

Dari implementasi secara *scratch dan library*, untuk penggunaan *polnomial kernel*, implementasi *scratch* memiliki *F1 score* yang lebih tinggi jika dibandingkan dengan implementasi menggunakan *library*. Namun, untuk *linear kernel*, implementasi menggunakan *library* memiliki *F1 score* yang lebih tinggi jika dibandingkan dengan implementasi secara *scratch*.

Improvement:

Improvement yang dapat dilakukan pada algoritma *Support Vector Machine* secara *scratch* dapat dilakukan dengan memperbaiki *learning rate, regularization parameter*, dan gamma. Selain itu, cara optimisasi harus diperbaiki untuk memperoleh hasil yang maksimal.