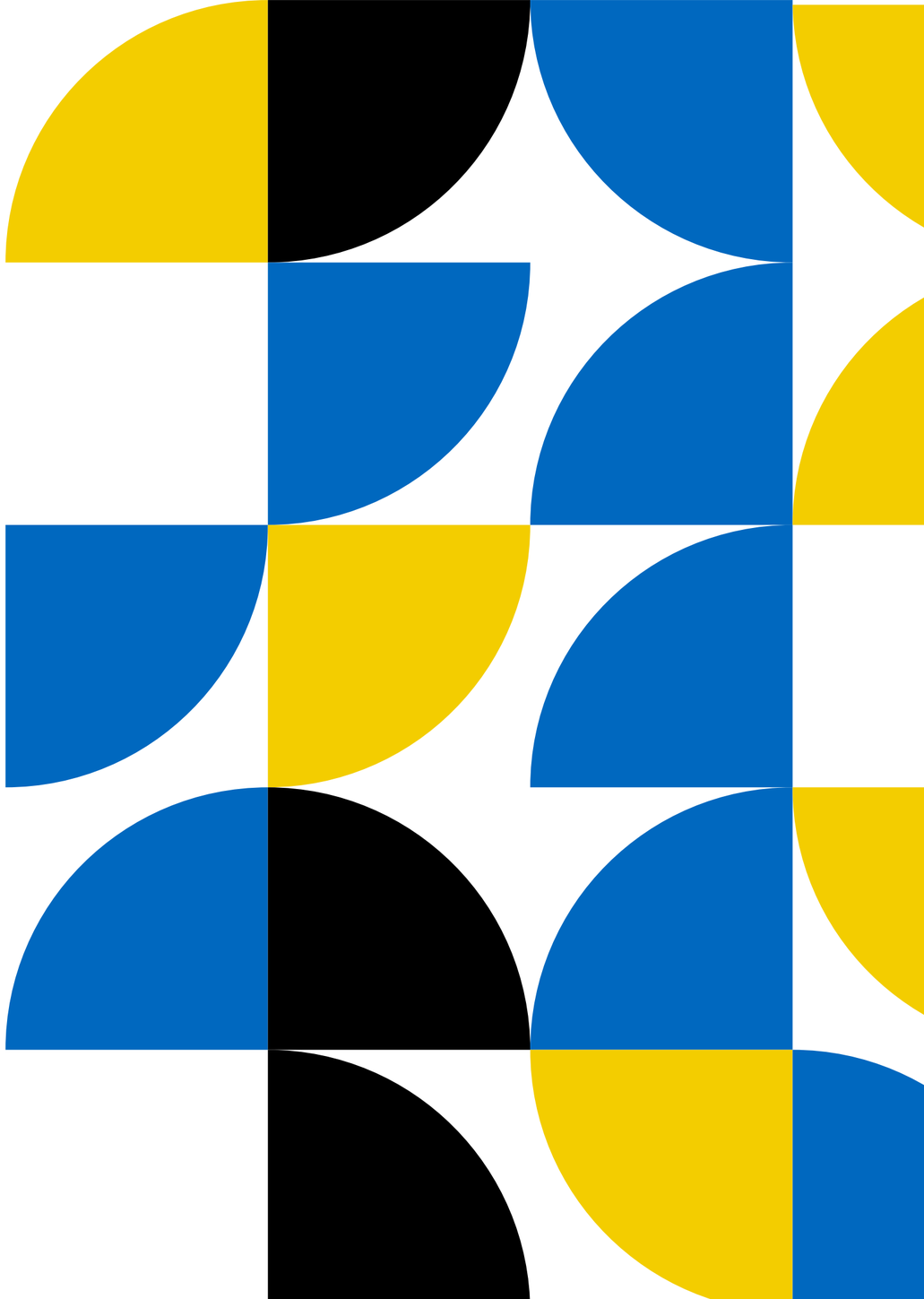
Support Vector Machine

Data Science - TRPL 5A

Oleh Kelompok 3

LANJUT



Anggota

01 Arya Yudha Prasetya - 234311007

02 Richo Novian Saputra - 234311024

03 Shaffa Dwiaji Feryansyah Putra - 234311028

Implementasi Support Vector Machine (SVM) untuk Klasifikasi Breast Cancer

Studi Kasus : UCI Breast Cancer Wisconsin Diagnostic Dataset

Konteks Masalah :

- Tujuan Utama: Membangun model yang dapat mengklasifikasikan benjolan (tumor) sebagai Ganas (Malignant) atau Jinak (Benign).
- Pentingnya SVM: Klasifikasi biner yang akurat sangat krusial dalam domain medis. SVM unggul dalam menemukan batas pemisah yang jelas (hyperplane) di antara dua kelas.
- **Tantangan**: Jumlah fitur yang banyak (30 dimensi) membuat masalah ini bersifat non-linear, sehingga membutuhkan Kernel Trick.

Latar Belakang

- Deteksi kanker payudara berbasis data membantu diagnosis lebih cepat.
- Machine Learning dapat mengidentifikasi pola pada data medis.
- SVM dikenal sebagai model yang kuat dalam memisahkan kelas secara optimal.
- Tujuan penelitian:
  a. Melakukan klasifikasi malign/benign menggunakan SVM
  b. Melakukan hyperparameter tuning
  c. Melihat visualisasi boundary model

Dataset

Breast Cancer Wisconsin (Diagnostic)

Sumber: UCI Machine Learning Repository (ID: 17)

Informasi dataset :

- Jumlah sampel: X.shape[0]
- Jumlah fitur: X.shape[1]
- Total fitur: 30
- Target:
  a. 0 = Benign
  b. 1 = Malignant

```
INFORMASI DATASET BREAST CANCER WISCONSIN (DIAGNOSTIC)

Jumlah sample (baris): 569
Jumlah fitur (kolom): 30

Daftar Nama Fitur
1. radius1
2. texture1
3. perimeter1
4. area1
5. smoothness1
6. compactness1
7. concavity1
8. concave_points1
9. symmetry1
10. fractal_dimension1
11. radius2
12. texture2
13. perimeter2
14. area2
15. smoothness2
16. compactness2
17. concavity2
18. concave_points2
19. symmetry2
20. fractal_dimension2
21. radius3
22. texture3
23. perimeter3
24. area3
25. smoothness3
26. compactness3
27. concavity3
28. concave_points3
29. symmetry3
30. fractal_dimension3

Total fitur: 30
```

Distribusi Kelas

- Mapping label:
  a. 0 = Benign (tidak berbahaya)
  b. 1 = Malignant (ganas)
- Jumlah masing-masing kelas ditampilkan dengan value_counts()
- Persentase kelas relatif seimbang → cocok untuk SVM

```
Informasi Target
Mapping label:
0 = Benign (tidak berbahaya)
1 = Malignant (ganas)

Jumlah masing-masing kelas:
Diagnosis
0    357
1    212
Name: count, dtype: int64

Persentase kelas:
Diagnosis
0    62.74
1    37.26
Name: proportion, dtype: float64
```

Pembagian Dataset

- Data dibagi menjadi tiga bagian:
- Train: 70%
- Validation: 15%
- Test: 15%
- Dengan stratified split → memastikan proporsi kelas tetap seimbang.

```
# EVALUASI PADA TRAIN SET
print("\nTRAIN RESULT")
y_train_pred = best_model.predict(X_train)

print(f"Train Accuracy: {accuracy_score(y_train, y_train_pred):.2%}")
print(classification_report(y_train, y_train_pred))

# EVALUASI PADA VALIDATION SET
print("\nVALIDATION RESULT")
y_val_pred = best_model.predict(X_val)

print(f"Validation Accuracy: {accuracy_score(y_val, y_val_pred):.2%}")
print(classification_report(y_val, y_val_pred))

# FINAL EVALUATION PADA TEST SET
print("\nTEST RESULT")
y_test_pred = best_model.predict(X_test)

print(f"Test Accuracy: {accuracy_score(y_test, y_test_pred):.2%}")
print(classification_report(y_test, y_test_pred))
```

Pipeline Model

Menggunakan pipeline untuk memastikan proses ML rapi dan terstruktur:

Pipeline berisi:

1. StandardScaler → menormalkan fitur
2. SVC (RBF Kernel) → algoritma klasifikasi utama

Kode :

```
# PIPELINE UNTUK SCALER + SVM
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('svm', SVC(kernel='rbf'))
])
```

Kelebihan Pipeline:

- Semua proses otomatis berjalan berurutan
- Menghindari data leakage
- Memudahkan hyperparameter tuning

Hyperparameter Tuning (Grid Search)

Parameter yang diuji:

- C: [0.1, 1, 10, 100]
- gamma: ['scale', 0.1, 0.01, 0.001]

Setting GridSearch:

- CV = 5
- Scoring = F1-score (bagus untuk data medis)
- n_jobs = -1 (parallel processing)
- verbose = 1

Tujuan: Menemukan kombinasi parameter terbaik untuk model SVM.

```
# GRID SEARCH UNTUK HYPERPARAMETER TUNING
param_grid = {
    'svm__C': [0.1, 1, 10, 100],
    'svm__gamma': ['scale', 0.1, 0.01, 0.001]
}

grid = GridSearchCV(
    pipeline,
    param_grid,
    cv=5,
    scoring='f1',
    n_jobs=-1,
    verbose=1
)

grid.fit(X_train, y_train)

print("\nBest Parameters:", grid.best_params_)
best_model = grid.best_estimator_
```

Hasil Hyperparameter Tuning

Output:

```
...  Fitting 5 folds for each of 16 candidates, totalling 80 fits

     Best Parameters: {'svm__C': 10, 'svm__gamma': 'scale'}
```

Evaluasi Pada Data Training

Kode :

```
# EVALUASI PADA TRAIN SET
print("\nTRAIN RESULT")
y_train_pred = best_model.predict(X_train)

print(f"Train Accuracy: {accuracy_score(y_train, y_train_pred):.2%}")
print(classification_report(y_train, y_train_pred))
```

Hasil evaluasi train:

```
TRAIN RESULT
Train Accuracy: 98.99%
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       250
           1       1.00      0.97      0.99       148

    accuracy                           0.99       398
   macro avg       0.99      0.99      0.99       398
weighted avg       0.99      0.99      0.99       398
```

Evaluasi Pada Validation Set

Kode :

```
# EVALUASI PADA VALIDATION SET
print("\nVALIDATION RESULT")
y_val_pred = best_model.predict(X_val)

print(f"Validation Accuracy: {accuracy_score(y_val, y_val_pred):.2%}")
print(classification_report(y_val, y_val_pred))
```

Hasil evaluasi validation:

```
VALIDATION RESULT
Validation Accuracy: 97.65%
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        53
           1       1.00      0.94      0.97        32

    accuracy                           0.98        85
   macro avg       0.98      0.97      0.97        85
weighted avg       0.98      0.98      0.98        85
```

Evaluasi Final Pada Test Set

Kode :

```
# FINAL EVALUATION PADA TEST SET
print("\nTEST RESULT")
y_test_pred = best_model.predict(X_test)

print(f"Test Accuracy: {accuracy_score(y_test, y_test_pred):.2%}")
print(classification_report(y_test, y_test_pred))
```

Hasil evaluasi test:

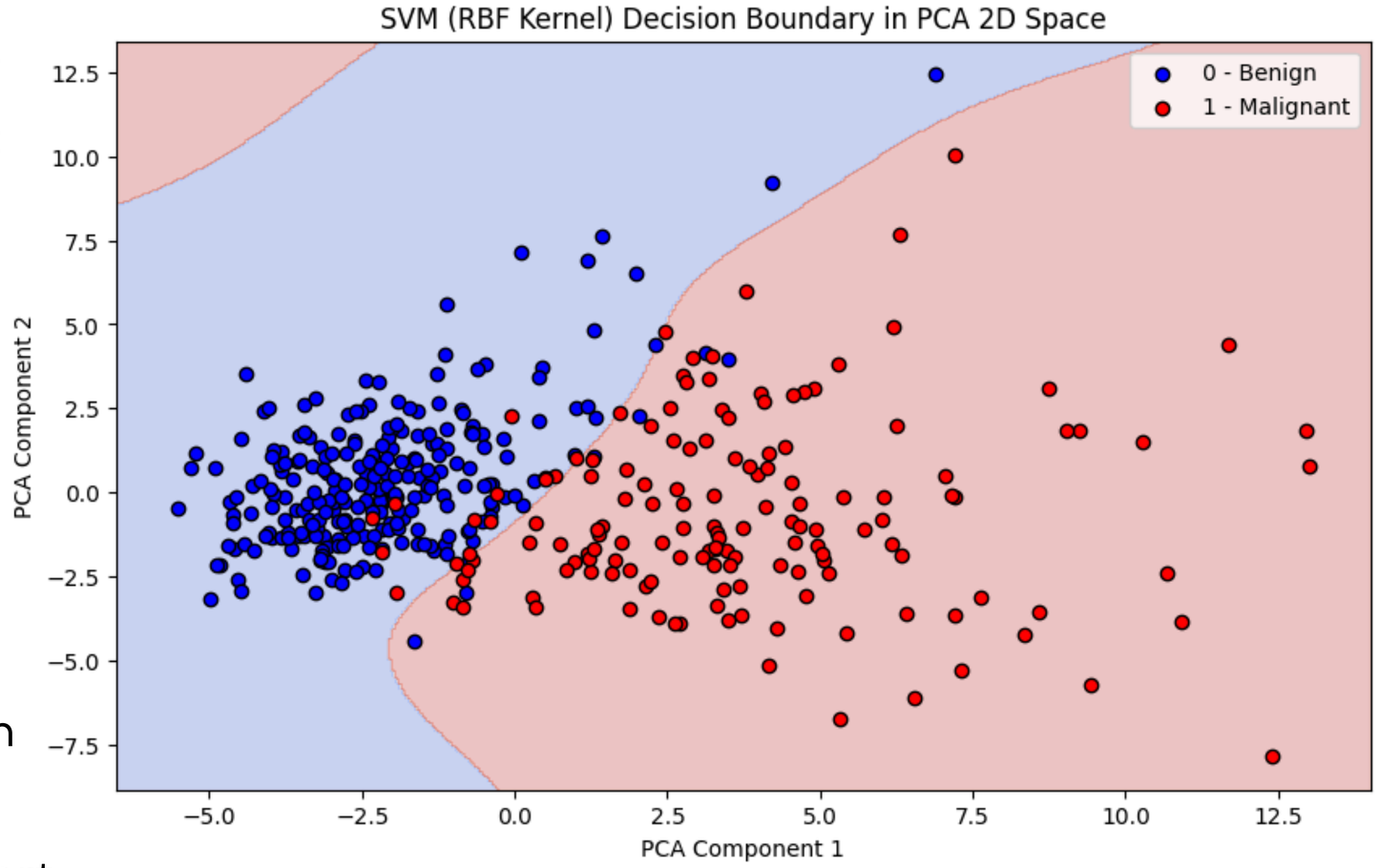
```
TEST RESULT
Test Accuracy: 96.51%
              precision    recall  f1-score   support

           0       0.96      0.98      0.97        54
           1       0.97      0.94      0.95        32

    accuracy                           0.97        86
   macro avg       0.97      0.96      0.96        86
weighted avg       0.97      0.97      0.96        86
```

Visualisasi Model (Decision Boundary)

- Visualisasi dilakukan dengan:
- PCA → reduksi dimensi ke 2 komponer
- Model SVM dengan parameter terbaik dilatih kembali pada ruang PCA
- Plot menunjukkan:
  a. Data benign (biru)
  b. Data malignant (merah)
  c. Boundary hasil model RBF



- Makna visualisasi:
  1. Boundary melengkung menunjukkan pola data non-linear
  2. Banyak titik dekat boundary → support vectors

Kesimpulan

- SVM berhasil mengklasifikasikan tumor benign/malignant dengan performa tinggi
- Hyperparameter tuning meningkatkan akurasi dan F1-score
- Visualisasi PCA menunjukkan pemisahan kelas yang jelas
- SVM cocok sebagai baseline model klasifikasi pada data medis

Terima Kasih