# Modul 9 ENSEMBLE CLASSIFICATION

Ensemble classification adalah teknik dalam machine learning yang menggabungkan beberapa model untuk menghasilkan prediksi yang lebih akurat dibandingkan hanya menggunakan satu model saja. Ide dasarnya sederhana, yaitu setiap model memiliki kelebihan dan kekurangan masing-masing, sehingga jika beberapa model digabungkan, kesalahan dari satu model dapat ditutupi oleh model lainnya. Dengan cara ini, hasil prediksi menjadi lebih stabil dan tidak mudah terpengaruh oleh bias atau kesalahan dari satu algoritma tertentu.

Dalam praktiknya, ensemble dapat dilakukan dengan berbagai cara, salah satunya adalah metode voting, yaitu setiap model memberikan hasil prediksi, lalu keputusan akhir ditentukan berdasarkan suara terbanyak. Pendekatan ini sering digunakan karena mudah diterapkan dan cukup efektif dalam meningkatkan performa model. Oleh karena itu, ensemble classification menjadi salah satu teknik penting, terutama ketika ingin mendapatkan hasil prediksi yang lebih andal pada data yang memiliki pola kompleks atau bervariasi.

## STUDI KASUS

Sebuah perusahaan e-commerce ingin memprediksi apakah seorang pelanggan akan melakukan pembelian (**1**) atau tidak (**0**) berdasarkan umur, pendapatan bulanan, dan jumlah kunjungan ke website per bulan. Permasalahan ini bertujuan membantu tim marketing dalam menentukan target promosi yang lebih tepat, sehingga sumber daya dapat difokuskan pada pelanggan dengan potensi pembelian tinggi. Untuk meningkatkan performa prediksi, digunakan beberapa model yaitu Random Forest, Support Vector Machine (SVM), dan XGBoost yang kemudian digabungkan menggunakan metode voting ensemble agar menghasilkan keputusan yang lebih stabil dan akurat dibandingkan model tunggal. Data yang digunakan sama dengan praktikum sebelumnya mengenai k-fold cross validation

---

## LANGKAH 1: IMPORT LIBRARY

Pada tahap ini dilakukan import library yang digunakan untuk pemodelan, evaluasi, dan visualisasi, di mana numpy digunakan untuk pengolahan data numerik, sklearn untuk model dan evaluasi, xgboost untuk boosting model, serta matplotlib untuk membuat grafik perbandingan performa model.

```
import numpy as np                                # library untuk array numerik
import time                                       # untuk menghitung waktu
komputasi

# Model machine learning
from sklearn.ensemble import RandomForestClassifier, VotingClassifier  #
Random Forest & Voting
from sklearn.svm import SVC                                             #
Support Vector Machine
from xgboost import XGBClassifier                                       #
XGBoost

# Evaluasi model
from sklearn.model_selection import cross_val_predict  # prediksi dengan k-
fold
from sklearn.metrics import confusion_matrix, accuracy_score  # metrik
evaluasi

# Visualisasi
import matplotlib.pyplot as plt                     # membuat grafik
```

---

**LANGKAH 2: INPUT DATA**

Data diubah ke dalam bentuk array agar dapat diproses oleh model, di mana X berisi fitur pelanggan dan y berisi label keputusan pembelian.

```
# Data fitur: [umur, pendapatan, kunjungan]
X = np.array([
[22,4,5],[25,5,6],[28,6,7],[30,6,8],[32,7,10],
[35,7,12],[38,8,13],[40,8,15],[42,9,16],[45,9,18],

[23,4,4],[26,5,7],[29,6,6],[31,6,9],[34,7,11],
[37,8,12],[39,8,14],[43,9,15],[46,10,17],[49,10,19],

[21,3,3],[24,4,5],[27,5,6],[30,6,7],[33,7,9],
[36,7,10],[38,8,11],[41,9,14],[44,9,16],[48,10,18],

[22,4,6],[25,5,5],[28,6,8],[31,6,7],[35,7,12],
[37,8,13],[40,8,14],[42,9,15],[45,9,17],[50,10,20],
```

```
[23,4,5],[26,5,6],[29,6,7],[32,7,9],[36,8,11],
[39,8,13],[41,9,14],[43,9,16],[47,10,18],[52,11,21]
])

# Label: 1 = beli, 0 = tidak beli
y = np.array([
0,0,0,0,1,1,1,1,1,1,
0,0,0,1,1,1,1,1,1,1,
0,0,0,0,1,1,1,1,1,1,
0,0,0,0,1,1,1,1,1,1,
0,0,0,1,1,1,1,1,1,1
])
```

---

**LANGKAH 3: MEMBUAT MODEL**

Pada tahap ini dibuat tiga model utama dengan karakteristik berbeda agar ensemble dapat bekerja optimal.

```
rf = RandomForestClassifier(n_estimators=50, random_state=42)  # model
Random Forest
svm = SVC(kernel='rbf')                                        # model SVM
dengan kernel RBF
xgb = XGBClassifier(use_label_encoder=False, eval_metric='logloss')  #
model XGBoost
```

---

**LANGKAH 4: VOTING ENSEMBLE**

Voting digunakan untuk menggabungkan hasil prediksi dari ketiga model dengan metode mayoritas suara. Pada tahap ini, **voting ensemble** bekerja dengan cara menggabungkan hasil prediksi dari beberapa model (Random Forest, SVM, dan XGBoost) untuk menentukan satu keputusan akhir. Mekanismenya dimulai ketika setiap model yang sudah dilatih menerima data input yang sama, kemudian masing-masing model akan menghasilkan prediksi berupa kelas, yaitu **0 (tidak beli)** atau **1 (beli)**. Karena setiap model memiliki cara belajar yang berbeda, hasil prediksinya juga bisa berbeda untuk data yang sama.

Selanjutnya, pada metode **hard voting**, sistem akan menghitung jumlah suara dari masing-masing prediksi tersebut dan memilih kelas yang memiliki suara terbanyak sebagai hasil akhir.

Misalnya, jika Random Forest memprediksi **1**, SVM memprediksi **0**, dan XGBoost memprediksi **1**, maka hasil akhirnya adalah **1** karena lebih banyak model yang memilih kelas tersebut (2 dari 3 model). Dengan cara ini, keputusan tidak bergantung pada satu model saja, melainkan hasil kolektif, sehingga kesalahan dari satu model dapat dikompensasi oleh model lainnya dan menghasilkan prediksi yang lebih stabil dan akurat.

```
voting = VotingClassifier(
    estimators=[
        ('rf', rf),     # model 1
        ('svm', svm),   # model 2
        ('xgb', xgb)    # model 3
    ],
    voting='hard'       # voting mayoritas
)
```

---

**LANGKAH 5: EVALUASI (k=5)**

Menggunakan k-fold cross validation untuk evaluasi dengan metrik accuracy, sensitivity, dan specificity.

```
models = {
    "Random Forest": rf,
    "SVM": svm,
    "XGBoost": xgb,
    "Voting": voting
}

results = []  # menyimpan hasil untuk visualisasi

for name, model in models.items():
    start = time.time()  # mulai waktu

    y_pred = cross_val_predict(model, X, y, cv=5)  # prediksi k-fold

    end = time.time()  # selesai waktu

    cm = confusion_matrix(y, y_pred)  # confusion matrix
```

```
    acc = accuracy_score(y, y_pred)   # accuracy

    # Ambil nilai dari confusion matrix
    TN, FP, FN, TP = cm.ravel()

    # Sensitivity (Recall)
    sensitivity = TP / (TP + FN)

    # Specificity
    specificity = TN / (TN + FP)

    results.append([name, acc, sensitivity, specificity])

    print(f"\nModel: {name}")
    print("Accuracy:", acc)
    print("Sensitivity:", sensitivity)
    print("Specificity:", specificity)
    print("Confusion Matrix:\n", cm)
    print("Waktu:", round(end-start,4), "detik")
```

---

## LANGKAH 6: VISUALISASI HASIL

Visualisasi digunakan untuk membandingkan performa masing-masing model secara lebih jelas.

```
# Konversi hasil ke array
results = np.array(results)

labels = results[:,0]                      # nama model
accuracy = results[:,1].astype(float)     # accuracy
sensitivity = results[:,2].astype(float)  # sensitivity
specificity = results[:,3].astype(float)  # specificity

x = np.arange(len(labels))  # posisi sumbu x

# Plot grafik
plt.figure(figsize=(8,5))  # atur ukuran biar lebih lega

plt.bar(x - 0.2, accuracy, width=0.2, label='Accuracy')
plt.bar(x, sensitivity, width=0.2, label='Sensitivity')
```

```
plt.bar(x + 0.2, specificity, width=0.2, label='Specificity')

plt.xticks(x, labels)
plt.ylabel("Nilai")
plt.title("Perbandingan Performa Model")

# Geser legend ke luar kanan
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))

plt.tight_layout()  # biar tidak kepotong
plt.show()
```

---

**INTERPRETASI HASIL**

Hasil menunjukkan bahwa setiap model memiliki keunggulan masing-masing, di mana Random Forest cenderung stabil dalam menangani data, SVM mampu membentuk batas keputusan yang optimal terutama pada data non-linear, dan XGBoost unggul dalam meningkatkan akurasi melalui mekanisme boosting; sementara itu voting ensemble menggabungkan ketiga model tersebut sehingga menghasilkan performa yang lebih seimbang antara sensitivity dan specificity, karena keputusan akhir diambil berdasarkan mayoritas suara sehingga kesalahan dari satu model dapat dikompensasi oleh model lainnya, dan hal ini terlihat dari grafik yang menunjukkan bahwa voting biasanya memiliki nilai yang lebih konsisten dibandingkan model individu.

---

**KESIMPULAN**

Metode ensemble voting terbukti efektif dalam meningkatkan performa klasifikasi karena mampu menggabungkan keunggulan berbagai model, serta evaluasi menggunakan k-fold cross validation memberikan hasil yang lebih stabil dan representatif terhadap keseluruhan data, sehingga pendekatan ini sangat direkomendasikan dalam kasus prediksi perilaku pelanggan untuk mendukung pengambilan keputusan bisnis.