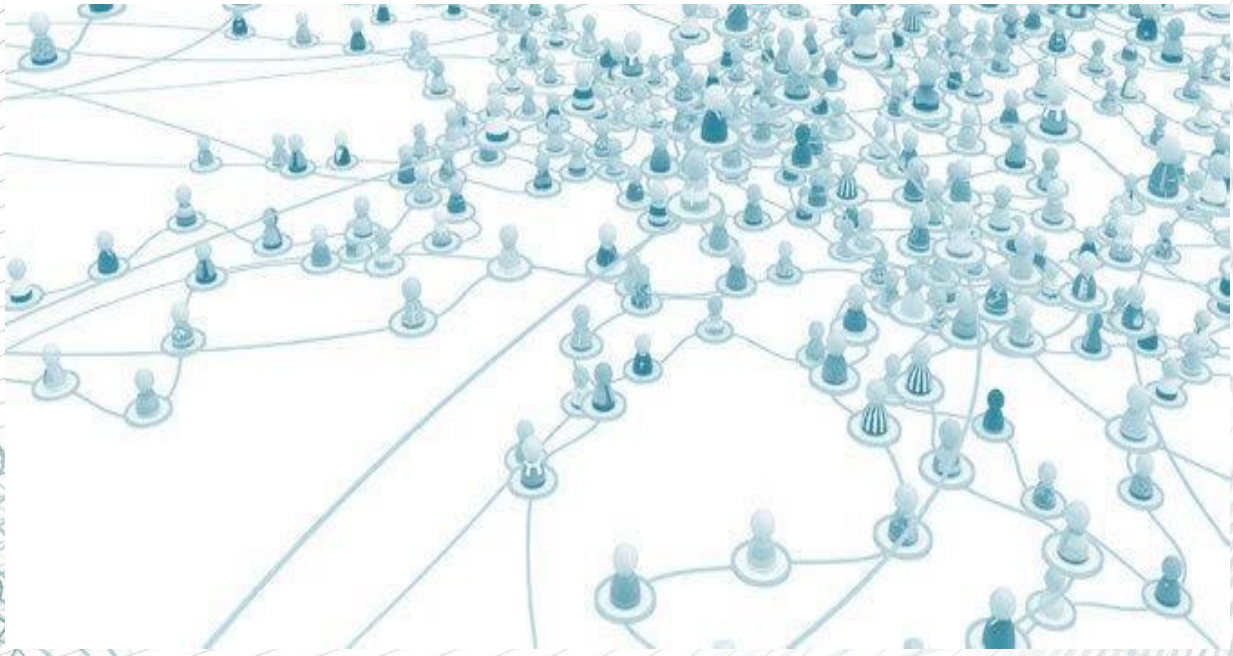
MODUL PRAKTIKUM

SD3107-Pembelajaran Mesin



Program Studi Sains Data
Fakultas Sains
Institut Teknologi Sumatera

2024

MODUL 9

Ensemble Method: Boosting

A. Konsep Dasar

a. Metode Ensemble: Boosting

Metode *ensemble* dalam *machine learning* adalah teknik yang menggabungkan beberapa model pembelajaran (atau *learners*) untuk meningkatkan kinerja prediksi dibandingkan dengan penggunaan model tunggal. Beberapa metode *ensemble* diantaranya *bagging* dan *boosting*. Pada *bagging*, tujuan dari penggabungan beberapa model adalah untuk meningkatkan model dan mengurangi variansi. Berbeda dengan *bagging*, penggabungan model pada *boosting* selain meningkatkan model pembelajaran juga untuk mengurangi bias pada model.

Pada *boosting* penggabungan model dilakukan secara sekuensial, dimana model pembelajaran yang terbentuk akan mengoreksi kekurangan dari model pembelajaran sebelumnya. Pada proses pelatihan, data yang salah dari model sebelumnya akan dilatih kembali hingga beberapa kali perulangan, sehingga saat pemilihan data latih, data yang masih diklasifikasikan salah pada pelatihan sebelumnya akan memiliki peluang lebih besar dipilih kembali pada pelatihan selanjutnya. Berikut adalah contoh data latih pada *boosting*.

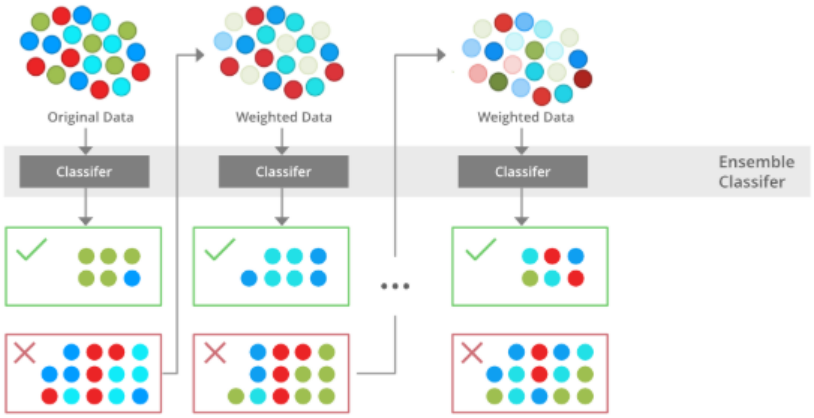
Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

Gambar 1 Data latih pada *boosting*

Tahapan pada metode *boosting* adalah sebagai berikut.

a) Inisialisasi bobot pada seluruh data. Lakukan pelatihan terhadap seluruh data latih yang ada.
b) Lakukan pengujian. Jika ada data yang terklasifikasi salah, data tersebut akan memiliki peluang lebih besar dipilih kembali saat pelatihan selanjutnya karena dilakukan pengambilan sampel acak berbobot dimana data yang terklasifikasi salah akan ditambah bobotnya dan data yang terklasifikasi benar akan berkurang bobotnya (*update* bobot).
c) Lakukan pelatihan menggunakan dataset baru yang berasal dari pemilihan sampel acak berbobot dari hasil pelatihan sebelumnya.
d) Lakukan pengujian ulang dan lihat apakah masih ada data yang terklasifikasi salah. Jika masih ada data yang salah, lakukan pelatihan kembali hingga dirasa sudah tidak ada data yang salah.

b. Adaboost



Gambar 2 Metode Adaboost

AdaBoost adalah salah satu model *boosting* yang dikembangkan paling awal. AdaBoost berfungsi dengan menggabungkan beberapa model **weak learners** untuk membentuk model yang kuat. Setiap model baru yang ditambahkan berfokus pada kesalahan yang dibuat oleh model sebelumnya, dengan memberikan bobot lebih pada data yang salah diklasifikasikan. Pada langkah awal, inisialisasi bobot akan dilakukan dengan pemberian nilai bobot awal sebesar $\frac{1}{n}$ dengan n adalah banyak data. Pada setiap pelatihan model, nilai eror akan dihitung menggunakan persamaan:

$$\varepsilon_i = \frac{1}{N}\sum_{j=1}^{N} w_j\, \delta\left(C_i(x_j) \neq y_j\right)$$

Dari persamaan di atas, nilai ε_i digunakan untuk menghitung nilai α dari model ke *i* yang merupakan konstanta untuk mengukur seberapa besar pengaruh model pada *ensemble*. Jika eror yang diperoleh lebih besar 50%, maka bobotnya kembali menjadi $\frac{1}{n}$ dan prosedur resampling kembali diulang. Jika tidak, hitung nilai α yang digunakan untuk update bobot. Perhitungan nilai α dapat dilakukan menggunakan persamaan berikut.

$$\alpha_i = \frac{1}{2}\ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right)$$

Jika model membuat kesalahan kecil (misalnya, ε_i kecil), maka α_i akan besar, yang berarti model ini memiliki kontribusi yang lebih besar pada model akhir. Untuk melakukan update bobot hitung dengan persamaan berikut.

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \times \begin{cases} e^{-a_j} & jika\ C_j(x_i) = y_i \\ e^{a_j} & jika\ C_j(x_i) \neq y_i \end{cases}$$

Dengan Z_j merupakan faktor normalisasi. Langkah-langkah tersebut akan berulang sebanyak nilai **k** yang telah ditentukan, kemudian dilakukan klasifikasi dengan persamaan berikut.

$$C^*(x) = \arg\max_y \sum_{j=1}^{T} \alpha_j \delta\big(C_j(x) = y\big)$$

Berikut adalah *pseudocode* dari Adaboost.

Algorithm 5.7 AdaBoost Algorithm

1: $\mathbf{w} = \{w_j = 1/n \mid j = 1, 2, \cdots, n\}$. {Initialize the weights for all n instances.}
2: Let k be the number of boosting rounds.
3: **for** $i = 1$ to k **do**
4: Create training set D_i by sampling (with replacement) from D according to w.
5: Train a base classifier C_i on D_i.
6: Apply C_i to all instances in the original training set, D.
7: $\epsilon_i = \frac{1}{n}\big[\sum_j w_j \ \delta\big(C_i(x_j) \neq y_j\big)\big]$ {Calculate the weighted error}
8: **if** $\epsilon_i > 0.5$ **then**
9: $\mathbf{w} = \{w_j = 1/n \mid j = 1, 2, \cdots, n\}$. {Reset the weights for all n instances.}
10: Go back to Step 4.
11: **end if**
12: $\alpha_i = \frac{1}{2} \ln \frac{1-\epsilon_i}{\epsilon_i}$.
13: Update the weight of each instance according to equation (5.88).
14: **end for**
15: $C^*(\mathbf{x}) = \arg\max_y \sum_{j=1}^{T} \alpha_j \delta(C_j(\mathbf{x}) = y)\big)$.

B. Tujuan Praktikum

I. Tujuan Instruksional Umum

Praktikum bertujuan untuk menerapkan *Boosting* pada klasifikasi data.

II. Tujuan Instruksional Khusus

1. Mahasiswa mampu menguasai konsep dasar Metode Ensemble: *Boosting*
2. Mahasiswa mampu menyelesaikan studi kasus klasifikasi menggunakan *Adaboost*
3. Mahasiswa mampu menganalisis hasil dari studi kasus menggunakan metode *Adaboost.*

C. Dataset dan Bahasa Pemrograman Python

C.1 Dataset

Dataset yang digunakan pada praktikum ini dapat dilihat pada tabel berikut.

X	y
1	1
2	-1
3	1
4	-1
5	1
6	-1
7	1
8	-1
9	1
10	-1
11	1
12	-1
13	1
14	1
15	1

C.2 Bahasa Pemrograman Python

Pada praktikum modul 9 ini, kita akan menggunakan bahasa pemrograman python dan beberapa library atau pustaka untuk memudahkan implementasi program. Pustaka yang akan digunakan diantaranya numpy, pandas, dan sklearn.

a. NumPy: digunakan untuk operasi matematika berbasis array dan matriks. NumPy sangat penting dalam komputasi numerik dengan Python pada praktikum ini.
b. Scikit-learn (sklearn): library untuk implementasi berbagai algoritma machine learning seperti klasifikasi, regresi, clustering, dan reduksi dimensi. Pada praktikum ini, kita akan menggunakan pustaka sklearn.svm.
c. Pandas : untuk manipulasi, analisis, dan pembersihan data berbasis struktuk DataFrame dan Series dengan berbagai fungsi efisien untuk membaca, mengolah, dan menyimpan data.

D. Implementasi Metode *Boosting*: *Adaboost*

Pada praktikum ini kita akan menggunakan Google colab yang dapat diakses menggunakan tautan https://colab.research.google.com/.

D.1 Adaboost pada Data Dimensi Kecil

1) Import pustaka atau library yang akan digunakan

```
# import library yang akan digunakan
import numpy as np
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

2) Masukkan dataset yang akan digunakan.

```
# Inisialisasi data
# X merupakan fitur data
X = np.array([[1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]])
# y merupakan label dari data
y = np.array([1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, 1, 1])
```

3) Bagi data latih dan data uji

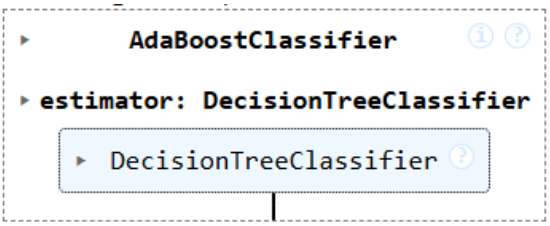
```
# Bagi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
```

4) Inisialisasi klasifier atau *weak learner*-nya. Di sini menggunakan Decision Tree sebagai *weak learner* pada Adaboost.

```
# Inisialisasi weak learner yang akan digunakan pada Adaboost
# Decision Tree dipilih sebagai weak learner dengan max_depth=1
weak_learner = DecisionTreeClassifier(max_depth=1)
# Buat model Adaboost. Gunakan weak_learner (Decision Tree) sebagai estimatornya
adaboost = AdaBoostClassifier(estimator=weak_learner, n_estimators=50)
```

5) Latih model Adaboost yang telah dibangun.

```
# Latih model AdaBoost yang telah dibangun
adaboost.fit(X_train, y_train)
```



6) Prediksi data uji menggunakan model yang sudah dilatih.

```
# Prediksi data uji menggunakan model yang sudah dilatih
y_pred = adaboost.predict(X_test)
```

7) Hitung akurasi menggunakan label hasil prediksi dengan label data uji

```
# Hitung akurasi
accuracy = accuracy_score(y_test, y_pred)
```

```
print(y_pred)
print(y_test)
print(f"Akurasi:{accuracy}")
```

```
[ 1  1 -1  1]
[-1 -1  1  1]
Akurasi:0.25
```

D.2 Adaboost pada Data Dimensi Tinggi

Pada praktikum D.1 dataset yang digunakan merupakan dataset dengan dimensi dan dataset berjumlah kecil, yaitu satu fitur saja dan hanya berjumlah 15 data. Pada praktikum D.2 akan dilakukan pelatihan dan pengujian Adaboost menggunakan dataset dengan dimensi lebih tinggi dan berjumlah lebih banyak. Tautan dataset yang digunakan adalah:

https://raw.githubusercontent.com/mirohmi/Heart_Disease_Diagnose/refs/heads/master/heart_diseases.csv

1) Import pustaka atau library yang akan digunakan

```
# Import library yang akan digunakan
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

2) Masukkan dataset yang akan digunakan.

```
url = 'https://raw.githubusercontent.com/mirohmi/Heart_Disease_Diagnose/refs/heads/master/heart_diseases.csv'
data = pd.read_csv(url)
data.head()
```

	age	gender	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
0	63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3	3.0	0.0	6.0	0
1	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.0	3.0	1
2	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0	2.0	7.0	1
3	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.0	3.0	0
4	63.0	1.0	4.0	130.0	254.0	0.0	2.0	147.0	0.0	1.4	2.0	1.0	7.0	1

3) Pisahkan fitur dan target. Pada dataset yang digunakan, terdapat 13 fitur yaitu dari kolom “age” hingga kolom “thal” dan label atau target yang memiliki dua kelas yaitu 1 dan 0 pada kolom “num”. Masukkan seluruh fitur ke dalam variabel X sedangkan target atau label ke dalam variabel y.

```
# Pisahkan fitur dan label
# Semua kolom kecuali kolom 'target' sebagai fitur
# Kolom target pada dataset ini adalah kolom num
X = data.drop('num', axis=1)
# Kolom 'num' sebagai label
y = data['num']
```

4) Bagi data latih dan data uji

```
# Bagi data menjadi data latih dan data uji (80% untuk pelatihan, 20% untuk pengujian)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

5) Inisialisasi klasifier atau *weak learner*-nya. Di sini menggunakan Decision Tree sebagai weak learner pada Adaboost.

```
# Inisialisasi model AdaBoost dengan Decision Tree sebagai klasifier atau weak learner
weak_learner = DecisionTreeClassifier(max_depth=1)  # Decision stump
adaboost = AdaBoostClassifier(estimator=weak_learner, n_estimators=50)
```

6) Latih model *Adaboost*

```
# Latih model AdaBoost dengan data pelatihan
adaboost.fit(X_train, y_train)
```

7) Prediksi data uji

```
# Prediksi dengan data uji
y_pred = adaboost.predict(X_test)
```

8) Hitung akurasi pada data uji

```
# Hitung akurasi pada data uji
accuracy = accuracy_score(y_test, y_pred)
```

9) Tampilkan hasil akurasi

```
# Tampilkan hasil
print("Akurasi AdaBoost pada dataset Heart Disease:", accuracy)
```

```
Akurasi AdaBoost pada dataset Heart Disease: 0.8703703703703703
```