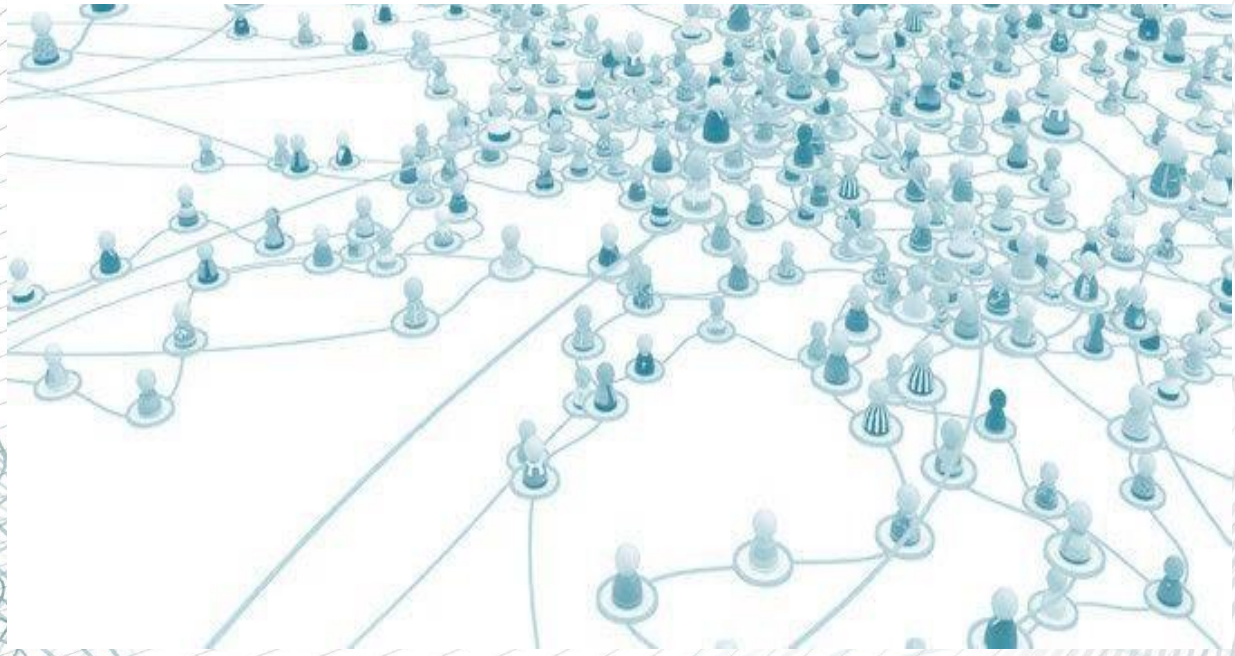
MODUL PRAKTIKUM

SD2531001-Data Mining



Program Studi Sains Data
Fakultas Sains
Institut Teknologi Sumatera

2025

MODUL 4

Feature Selection

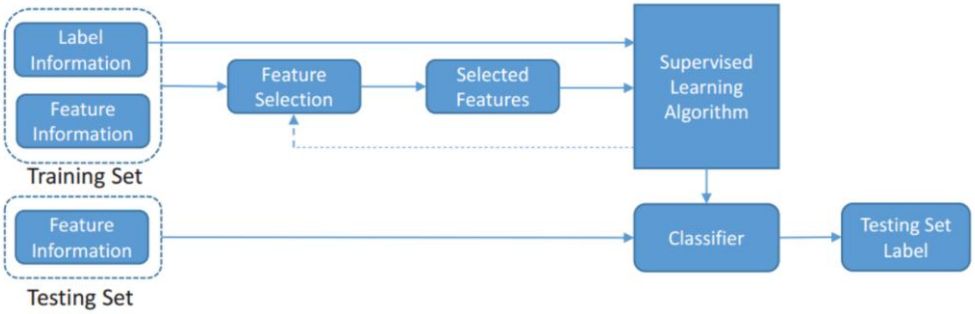
A. Konsep Dasar

1) Konsep Information Extraction dan Feature Selection

Information Extraction (IE) adalah proses mengidentifikasi, mengekstrak, dan menyusun informasi terstruktur dari data tidak terstruktur atau semi-terstruktur (misalnya teks, dokumen, web, media sosial). Tujuannya adalah mengubah data mentah menjadi fakta yang bisa dianalisis. Pada bidang Kecerdasan Buatan (*Artificial Intelligence*/AI), khususnya Pembelajaran Mesin (*Machine Learning*), proses ekstraksi informasi dibutuhkan untuk:

1. **Menyediakan data yang relevan bagi model pembelajaran mesin.** Sebelum model dapat belajar, informasi penting harus diidentifikasi dan diekstraksi dari sumber data mentah seperti teks berita, laporan, atau komentar pengguna.
2. **Meningkatkan kualitas fitur (feature engineering).** Hasil ekstraksi informasi sering digunakan sebagai fitur masukan (input features) untuk model klasifikasi, prediksi, atau rekomendasi.
3. **Mendukung otomatisasi analisis data besar (big data).** Dengan IE, sistem dapat secara otomatis mengenali entitas, hubungan, dan peristiwa tanpa perlu campur tangan manusia.

Dalam sebuah model pembelajaran mesin (*machine learning*), khususnya ketika menggunakan dataset berukuran besar (*big data*), sering kali seluruh fitur yang tersedia tidak dapat digunakan secara langsung. Hal ini disebabkan karena tidak semua fitur memiliki relevansi yang signifikan terhadap target atau label yang ingin diprediksi. Fitur adalah atribut, variabel, atau karakteristik yang menggambarkan suatu objek atau entitas dalam dataset dan digunakan sebagai input dalam model pembelajaran mesin (*machine learning*). Penggunaan seluruh fitur tanpa seleksi dapat menimbulkan berbagai masalah, seperti meningkatnya kompleksitas model, waktu komputasi yang lama, dan bahkan penurunan akurasi akibat *overfitting*. Sebagai contoh, dalam klasifikasi Indeks Massa Tubuh (*Body Mass Index*/BMI) seseorang, fitur yang benar-benar berpengaruh hanyalah berat badan (weight) dan tinggi badan (height). Fitur lain seperti warna rambut, pekerjaan, atau alamat tidak memiliki hubungan langsung dengan nilai BMI. Oleh karena itu, diperlukan proses seleksi fitur (feature selection) untuk memilih hanya fitur-fitur yang paling relevan sebelum dilakukan pelatihan (training) dan pengujian (testing) model. Berikut adalah gambaran sebuah proses klasifikasi *machine learning*.



Gambar 1 Alur proses sebuah model klasifikasi

Tujuan dari proses seleksi fitur pada sebuah model adalah:

- Mengurangi *dimensionality* (jumlah fitur).
- Mengurangi *noise* atau data tidak relevan.
- Meningkatkan akurasi model.
- Mempercepat pelatihan dan inferensi.
- Menghindari *overfitting.*

2) Metode Seleksi Fitur

a) Filter Method

Filter methods adalah pendekatan seleksi fitur yang memilih fitur berdasarkan ukuran statistik atau hubungan matematis antara fitur dengan target (label), tanpa melibatkan pelatihan algoritma *machine learning.* Teknik umum pada metode filter yang biasanya digunakan adalah:

1. Korelasi (*Correlation Coefficient*)
2. *Chi-Square Test*
3. *Algoritma Relief*
4. ANOVA (*Analysis of Variance*)
5. *Information Gain / Mutual Information*
6. *Variance Threshold*

b) Wrapper Method

Pendekatan ini menggunakan model machine learning untuk menilai kualitas subset fitur. Model dilatih berulang kali dengan kombinasi fitur yang berbeda, dan performa terbaik dipilih. Jenis seleksi fitur yang termasuk ke dalam metode *wrapper* diantaranya:

1. *Forward Selection.*
   Forward Selection membangun subset fitur secara bertahap dari nol, kemudian menambahkan fitur satu per satu hingga mencapai kinerja terbaik.
2. *Backward Elimination*
   Jika Forward Selection mulai dari subset kosong dan menambah fitur satu per satu, maka Backward Elimination mulai dengan semua fitur lalu menghapus fitur satu per satu hingga diperoleh kinerja paling optimum.
3. *Recursive Feature Elimination (RFE)*
   Recursive Feature Elimination bekerja dengan cara menghapus fitur yang kontribusinya paling kecil secara bertahap sampai tersisa jumlah fitur yang diinginkan.

c) Embedded Method

Seleksi fitur dilakukan selama proses pelatihan model. Model secara otomatis memberi bobot pada fitur berdasarkan kontribusinya terhadap hasil prediksi. Algoritma dalam Embedded Methods:

- Regularisasi (L1, L2)
- Random Forest Importance

B. Tujuan Praktikum

I. Tujuan Instruksional Umum

Praktikum bertujuan untuk menerapkan metode seleksi fitur pada model pembelajaran.

II. Tujuan Instruksional Khusus

1. Mahasiswa mampu menguasai konsep dasar seleksi fitur.
2. Mahasiswa mampu memilih dan menganalisa metode seleksi fitur yang digunakan.

C. Dataset dan bahasa pemrograman Python

C.1 Dataset

Pada praktikum ini, metode yang akan dipraktikkan adalah metode *filter* dan *wrapper*. Dataset yang digunakan digunakan pada praktikum ini dapat dilihat pada tabel berikut.

a. Dataset BMI

Dataset BMI merupakan data numerik sehingga pada praktikum ini dataset akan digunakan pada tahapan seleksi fitur menggunakan metode Korelasi Pearson.

No	Tinggi (cm)	Berat (kg)	Usia (tahun)	BMI
1	150	45	10	20
2	160	55	25	21.5
3	170	65	30	22.5
4	180	75	18	23.1
5	190	85	20	23.5

b. Dataset Churn

Dataset churn berikut merupakan data kategorik sehingga pada praktikum ini dataset akan digunakan pada tahapan seleksi fitur menggunakan metode Chi Square.

No	Jenis_Kelamin	Usia	Lama_Berlangganan	Keluhan	Pembayaran_Tepat	Churn
1	Pria	25	12	Tidak	Ya	Tidak
2	Wanita	30	6	Ya	Tidak	Ya
3	Pria	40	24	Tidak	Ya	Tidak
4	Wanita	22	3	Ya	Tidak	Ya
5	Pria	35	18	Tidak	Ya	Tidak
6	Wanita	27	5	Ya	Tidak	Ya
7	Pria	50	36	Tidak	Ya	Tidak
8	Wanita	29	8	Ya	Tidak	Ya
9	Pria	33	20	Tidak	Ya	Tidak
10	Wanita	24	4	Ya	Tidak	Ya

c. Dataset Penyakit Hipertensi

Pada praktikum seleksi fitur menggunakan metode *wrapper,* dataset dapat diunduh terlebih dahulu pada tautan berikut dengan format berkas .csv.

https://docs.google.com/spreadsheets/d/1vtl_clejRPZhm9XRlR8yvoqGd04z_HxokcURNOvM0nY/edit?usp=sharing

C.2 Bahasa Pemrograman Python

Pada praktikum modul 3 ini, kita akan menggunakan bahasa pemrograman python dan beberapa library atau pustaka untuk memudahkan implementasi program. Pustaka yang akan digunakan diantaranya numpy, pandas, sklearn, dan matplotlib.

a. NumPy: digunakan untuk operasi matematika berbasis array dan matriks. NumPy sangat penting dalam komputasi numerik dengan Python pada praktikum ini.
b. Pandas : library untuk pengolahan data berbasis tabel (DataFrame). Pandas juga dapat melakukan pengolahan komputasi numerik seperti numpy, contohnya seperti rata-rata dan standar deviasi.
c. Scikit-learn (sklearn): library untuk implementasi berbagai algoritma machine learning seperti klasifikasi, regresi, clustering, seleksi fitur, dan reduksi dimensi. Pada praktikum ini, kita akan menggunakan pustaka sklearn.svm.

D. Implementasi Metode Seleksi Fitur

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

i. Metode Filter dengan Korelasi Pearson

1) Import pustaka atau library yang akan digunakan

```
# Import library
import pandas as pd
import numpy as np
```

2) Masukkan dataset yang akan digunakan.

```
# Masukkan dataset yang akan digunakan
data = {
    'Tinggi': [150, 160, 170, 180, 190],
    'Berat': [45, 55, 65, 75, 85],
    'Usia': [10, 25, 30, 18, 20],
    'BMI': [20.0, 21.5, 22.5, 23.1, 23.5]  # Target
}
df = pd.DataFrame(data)
```

3) Hitung korelasi tiap fitur terhadap target menggunakan korelasi Pearson.

```
# Hitung korelasi Pearson antar fitur dan target
corr = df.corr(method='pearson')['BMI'].drop('BMI')
```

4) Tetapkan nilai ambang batas r.

```
# Tentukan ambang batas (misalnya |r| > 0.5)
threshold = 0.8
selected_features = corr[abs(corr) > threshold].index.tolist()

print("Nilai Korelasi Fitur terhadap BMI:")
print(corr)
print("\nFitur yang Dipilih (|r| > 0.5):", selected_features)
```

5) Lihat hasil seleksi fitur.

```
# Dataset baru hanya berisi fitur terpilih
X_selected = df[selected_features]
y = df['BMI']

print("\nDataset setelah seleksi fitur:")
print(X_selected)
```

ii. **Metode Filter dengan Chi Square Test**

1. Import pustaka atau library yang akan digunakan.

```
import pandas as pd
# Untuk mengubah data kategorik menjadi format numerik
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_selection import chi2
```

2. Masukkan dataset yang akan digunakan.

```
# Dataset  churn sederhana
data = {
    'Jenis_Kelamin': ['Pria','Wanita','Pria','Wanita','Pria',
                      'Wanita','Pria','Wanita','Pria','Wanita'],
    'Usia': [25,30,40,22,35,27,50,29,33,24],
    'Lama_Berlangganan': [12,6,24,3,18,5,36,8,20,4],
    'Keluhan': ['Tidak','Ya','Tidak','Ya','Tidak','Ya',
                'Tidak','Ya','Tidak','Ya'],
    'Pembayaran_Tepat': ['Ya','Tidak','Ya','Tidak','Ya',
                         'Tidak','Ya','Tidak','Ya','Tidak'],
    'Churn': ['Tidak','Ya','Tidak','Ya',
              'Tidak','Ya','Tidak','Ya','Tidak','Ya']
}

df = pd.DataFrame(data)

print("Dataset Awal:")
print(df)
```

3. Ubah kolom kategorikal menjadi numerik menggunakan LabelEncoder..

```
# Encode semua kolom kategorikal
le = LabelEncoder()
df_encoded = df.apply(le.fit_transform)
```

4. Pisahkan fitur dan target.

```
# Pisahkan fitur dan target
X = df_encoded.drop('Churn', axis=1)
y = df_encoded['Churn']
```

5. Lihat hasil Chi-Square Test.

```
# Hitung Chi-Square test
chi_scores, p_values = chi2(X, y)

# Buat tabel hasil
chi2_results = pd.DataFrame({
    'Fitur': X.columns,
    'Chi2_Score': chi_scores,
    'p_value': p_values
}).sort_values(by='Chi2_Score', ascending=False)

print("\nHasil Seleksi Fitur (Chi-Square Test):")
print(chi2_results)
```

6. Seleksi fitur dengan nilai p-value<0.05

```
# Seleksi fitur signifikan (p-value < 0.05)
signif_features = chi2_results[chi2_results['p_value'] < 0.05]
print("\n=== Fitur yang Signifikan (p < 0.05) ===")
print(signif_features)
```

7. Hasil seleksi fitur

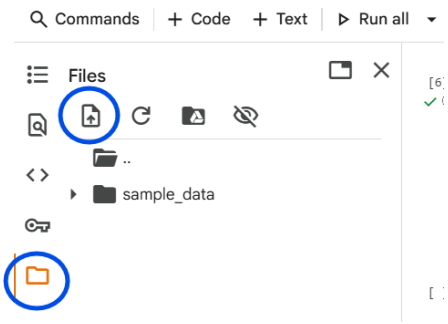
```
=== Fitur yang Signifikan (p < 0.05) ===
               Fitur  Chi2_Score   p_value
2  Lama_Berlangganan   13.888889  0.000194
1               Usia    8.022222  0.004621
0      Jenis_Kelamin    5.000000  0.025347
3            Keluhan    5.000000  0.025347
4   Pembayaran_Tepat    5.000000  0.025347
```

iii. Metode Wrapper dengan Forward Selection dan Backward Elimination

1. Import pustaka atau library yang akan digunakan.

```
import pandas as pd
# Pada praktikum ini algoritma ML yang digunakan adalah Logistic Regression
from sklearn.linear_model import LogisticRegression
# Seleksi fitur forward & backward menggunakan pustaka sklearn.feature_selection
from sklearn.feature_selection import SequentialFeatureSelector
# Pembagian data latih dan data uji menggunakan pustaka sklearn.model_selection
from sklearn.model_selection import train_test_split
```

2. Unggah dataset yang akan digunakan dengan klik pada ikon yang dilingkari.



3. Import dataset. Sesuaikan nama dan direktori file masing-masing.

```
df = pd.read_csv('/content/heart - heart.csv')
df.head()
```

4. Tentukan fitur dan target.

```
# Tentukan fitur dan target
X = df.drop(columns=['output']) # pilih semua kolom kecuali kolom 'class' sebagai fitur
y = df['output'] # kolom class sebagai target
```

5. Bagi data latih dan data uji. Pada bagian ini, dataset melibatkan pelatihan dan pengujian model, sehingga perlu dilakukan pembagian dataset. Pada seleksi fitur ini, algoritma yang digunakan adalah Regresi Logistik dengan menggunakan pustaka sklearn.

```
# Bagi data latih dan data uji dengan perbandingan 70:30
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Model dasar
model = LogisticRegression(max_iter=1000, solver='liblinear')
```

6. Implementasi forward seleksi menggunakan pustaka `SequentialFeatureSelector`. Pada praktikum ini, hasil seleksi fitur metode forward adalah 'cp', 'chol', 'restecg', 'exng', 'caa', dan 'thall'.

```
# Forward Selection (pilih fitur terbaik)
forward_selector = SequentialFeatureSelector(model,direction='forward')
forward_selector.fit(X_train, y_train)

print("Fitur terpilih (Forward):")
print(list(X_train.columns[forward_selector.get_support()]))
```

7. Implementasi forward seleksi menggunakan pustaka `SequentialFeatureSelector`. Pada praktikum ini, hasil seleksi fitur metode backward adalah 'sex', 'cp', 'thalachh', 'exng', 'slp', 'caa', 'thall'.

```
# Implementasi Backward Elimination
backward_selector = SequentialFeatureSelector(model, direction='backward')
backward_selector.fit(X_train, y_train)
feat_backward = X_train.columns[backward_selector.get_support()]
print("Fitur terpilih (Backward):", list(feat_backward))
```

8. Hitung kinerja hasil seleksi fitur dengan melatih model menggunakan fitur hasil seleksi.

```
# Evaluasi performa hasil Forward Selection
from sklearn.metrics import accuracy_score
model.fit(X_train[feat_forward], y_train)
y_pred_f = model.predict(X_test[feat_forward])
acc_f = accuracy_score(y_test, y_pred_f)
print("Akurasi (Forward):", acc_f)

# Evaluasi performa hasil Bacward Elimination
model.fit(X_train[feat_backward], y_train)
y_pred_b = model.predict(X_test[feat_backward])
acc_b = accuracy_score(y_test, y_pred_b)
print("Akurasi (Backward):", acc_b)

Akurasi (Forward): 0.7692307692307693
Akurasi (Backward): 0.8021978021978022
```