IF3170 Inteligensi Artifisial

Tugas Besar 2

Implementasi Algoritma Pembelajaran Mesin

Disiapkan oleh:

1. Panji Sri Kuncara Wisma 13522028
2. Zaki Yudhistira Candra 13522031
3. Ahmad Hasan Albana 13522041
4. Haikal Assyauqi 13522052

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2024/2025

Daftar Isi

Algoritma

1. KNN

a. Training (fit):
Pada tahap ini, data latih (X_train_transformed dan y_train_transformed) hanya disimpan. Tidak ada proses pembelajaran yang kompleks.

b. Membuat Prediksi (predict):
Saat memprediksi label untuk satu data uji (X_test_transformed) :

1. Menghitung jarak antara x_test dengan setiap titik data latih.
2. Mengurutkan data latih berdasarkan jarak terdekat ke titik uji.
3. Mengambil k titik terdekat (k_neighbours).
4. Menentukan label mayoritas dari k titik terdekat tersebut sebagai hasil prediksi.

c. Menghitung Jarak (calculate_distance) :
Tergantung pada parameter distance_type, jarak dapat dihitung dengan :

1. Euclidean

$$d = \sqrt{\sum (x_{test} - x_{train})^2}$$

2. Manhattan

$$d = \sum |x_{test} - x_{train}|$$

3. Minkowski

$$d = \left(\sum |x_{test} - x_{train}|^p \right)^{\frac{1}{p}}$$

d. Setelah mendapatkan k tetangga terdekat dan labelnya, pilih label yang paling sering muncul di antara tetangga tersebut sebagai prediksi akhir. Kalau misalkan jumlahnya sama, pilih salah satu.

2. Naive-Bayes

Dalam Naive-Bayes terdapat 3 tahap penting yang dilakukan yakni:

a. Penghitungan class probability P(y)
b. Penghitungan P(X|y), karena data kita numerikal, maka kita menggunakan rumus yang di bawah untuk menentukan nilainya

$$f(100000) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

c. Melakukan perhitungan prediksi tiap row

3. ID3

a. Inisialisasi:
   Data latih (X, y) dibaca. Label paling umum (most_common_label) disimpan sebagai label cadangan.
b. Diskritisasi Fitur Numerik:
   Setiap fitur numerik dicari titik potong terbaik (threshold) yang memaksimalkan Information Gain. Kemudian fitur diubah jadi kategori ("<threshold" atau ">=threshold").
c. Menghitung Entropy & Information Gain:
   1. *Entropy* mengukur ketidakpastian label.
   2. *Information Gain* dihitung untuk tiap fitur untuk melihat seberapa baik fitur tersebut dalam memisahkan data berdasarkan label.
d. Membangun Pohon:
   1. Memilih fitur dengan Information Gain tertinggi sebagai node.
   2. Membagi data menurut nilai fitur tersebut.
   3. Mengulangi proses pada cabang-cabangnya hingga semua data pada cabang homogen (satu label) atau tidak ada fitur yang tersisa.
e. Prediksi:
   1. Untuk memprediksi label sebuah sampel baru, kami menelusuri pohon dari akar hingga daun sesuai dengan nilai fitur pada sampel.
   2. Jika nilai sampel tidak ditemukan di suatu node, kami pake most_common_label sebagai *fallback*.

Cleaning dan *Preprocessing*

Data cleaning merupakan salah satu tahapan dalam proses pembelajaran mesin. Proses ini mencakup pembersihan data sebelum nanti diproses dan dipersiapkan lebih lanjut untuk pembelajaran mesin. Dalam proses ini, dapat dilakukan beberapa rangkaian pengolahan data:

1. Menghapus data yang tidak relevan
2. Menangani nilai yang hilang (missing values)
3. Menghapus atau menangani data duplikat
4. Menangani outlier
5. Memperbaiki format data yang salah

Data preprocessing merupakan proses untuk menyiapkan data, setelah data 'dibersihkan', data akan diubah menjadi sebuah *format* yang dapat diterima oleh mesin pembelajar. Proses *data preprocessing* meliputi

1. Normalisasi atau standarisasi data
2. *Encoding* variabel kategori (*one-hot encoding* atau *label encoding*)
3. Reduksi dimensi (*dimensionality reduction*)
4. Transformasi fitur (*feature transformation*)
5. Pembagian data menjadi *train*, *validation*, dan *test set*

Berikut adalah tahapan *data cleaning* yang kami implementasikan. Implementasi lebih lanjut mengenai setiap tahapan dapat dilihat pada file *notebook* dengan ekstensi .ipynb.

1. Pemisahan data (*Data splitting*)

   Dilakukan pembagian dataset menjadi *training data* dan *validation data*. Hal ini dilakukan untuk menentukan proporsi dan bagian data mana saja yang akan digunakan untuk melatih mesin dan mana yang dijadikan validator untuk menguji validitas mesin sebelum nanti akan diuji menggunakan data sesungguhnya. Proporsi yang digunakan adalah 20 % data validasi dan 80 % data *training* dari total data keseluruhan.

2. Pembersihan data (*Data cleaning*)
   a. Pengolahan nilai yang hilang

      Digunakan sebuah *imputer*, yakni kakas untuk mengisi nilai yang hilang. Nilai setiap fitur yang hilang akan diisi menggunakan 3 strategi pengisian.

      - Penggantian dengan median : mengganti nilai kosong dengan median dataset, dilakukan apabila distribusi data condong ke salah satu sisi.
      - Penggantian dengan mean : digunakan pada distribusi yang relatif normal, tidak ada kecondongan ke salah satu sisi.
      - Penggantian dengan modus : digunakan untuk fitur kategorikal.

   b. Pengolahan *outlier*

      Outlier dikelola dengan menggunakan IQR, nilai-nilai di luar IQR akan dipangkas dan disesuaikan dengan 1.5 kali IQR.

Dilakukan transformasi fitur demi mengurangi kecondongan / kemiringan distribusi. Intinya kami mencoba semua teknik disini, mulai dari imputasi, clip , log dan sqrt, terus kami cari yang menghasilkan nilai *skew* paling kecil. Nah itu yang kami pakai.

c. Penghapusan nilai duplikat

Data duplikat pada *dataset* dihapus.

d. Rekayasa fitur (*feature engineering*)

Dilakukan juga pemangkasan fitur-fitur pada *dataset*. Fitur yang memiliki korelasi yang tinggi akan dipangkas dan tidak digunakan pada pelatihan mesin karena redundan.

3. Data Preprocessing

a. Skala fitur

Diimplementasikan sebuah kelas bernama *custom scaler* yang melakukan normalisasi pada setiap nilai fitur. Terdapat 2 metode utama dalam melakukan normalisasi yang digunakan:

- Standard fit
- Robust fit

b. Pengkodean fitur (*encoding*)

Diterapkan *one-hot encoding* pada fitur kategorikal.

c. Pemrosesan data yang tidak seimbang

Digunakan metode *sampling* untuk mengatasi ketidak seimbangan pada *dataset.* Digunakan metode *under sampling*, yaitu pengurangan *instance* untuk menyeimbangkan dataset.

d. Pengurangan dimensi

Digunakan *Principal Component Analysis* (PCA) *transformer*. Diterapkan juga sebuah kelas yang memberikan pengguna kebebasan untuk menentukan komponen mana saja yang ingin dihilangkan dan yang ingin dijaga.

4. Pembentukan *pipeline* pemrosesan data

Semua kelas pada bagian sebelumnya akan dieksekusi pada kelas ini. *Pipeline* dibuat sebagai titik eksekusi keseluruhan proses baik *data cleaning* maupun *data preprocessing*. *Pipeline* dibentuk berdasarkan target pemrosesan data, untuk implementasi lebih lengkapnya dapat dilihat pada *file .ipynb*.

Untuk *feature scaling*, kami pake *robust* buat kolom yang outliernya tinggi, terus kami pake standarisasi buat fitur yang outliernya rendah sama fitur sisanya. Untuk feature encoding, kami itu pake one hot encoding, alasannya karena fitur kategorikalnya bukan ordinal, sempat bingung karena fiturnya menjadi banyak, tapi teknik encoding lain dirasa kurang cocok dan karena nantinya pake PCA jadi nanti fiturnya juga akan berkurang. Untuk handling imbalanced cukup *straightforward* penjelasannya karena barisnya terlalu banyak, jadi kami undersampling. Untuk

dimensionality reduction kami pake PCA karena merasa 2 teknik lain tidak cocok untuk kasus ini. Mungkin penjelasannya lebih detail nanti ketika demo, kalau ada demo.

Hasil Prediksi dan Perbandingan

1. KNN

From Scratch	0.34705096153860965
Library	0.3671453453963325
Penjelasan : Diperoleh nilai F1-Score yang lebih baik apabila menggunakan library.	

2. Naive Bayes

From Scratch	0.2599162594937131
Library	0.2595194803877786
Penjelasan : Diperoleh nilai F1-Score yang lebih baik apabila menggunakan algoritma from *scratch*.	

3. ID3

From Scratch	0.2701457551553185
Library	0.29230641832790427
Penjelasan : Diperoleh nilai F1-Score yang lebih baik apabila menggunakan algoritma bawaan *library*.	

Pembagian Tugas

No	NIM	Tugas
1	13522028	Data Cleaning, Data Preprocessing, KNN, Integration
2	13522031	EDA, Testing, Report, Submission, Assisting in Code Development, Error Analysis
3	13522041	ID3, Testing, Assisting in Code Development
4	13522052	Data Cleaning, Preprocessing, Naive Bayes