INFORMATICS ENGINEERING STUDY PROGRAM
SCHOOL OF ELECTRICAL ENGINEERING DAN INFORMATICS
BANDUNG INSTITUTE OF TECHNOLOGY

Tugas Besar 2 IF3170 Inteligensi Artifisial
Implementasi Algoritma Pembelajaran Mesin

Reported By:

AIB

Member:

Yusuf Ardian Sandi (13522015)

Dhidit Abdi Aziz (13522040)

Sa'ad Abdul Hakim (13522092)

Muhamad Rifki Virziadeili Harisman (13522120)

I. Tahap Cleaning dan Preprocessing

A. Handling Missing Data

Pada data binary, kami melakukan pengisian NaN dengan “new value” yaitu -1. Begitupun pada data kategorikal, NaN juga diisi dengan kategori baru yaitu “Unknown”. Hal ini dilakukan untuk tetap memberikan makna pada data yang tidak memiliki nilai. Adapun pada data numerikal, NaN diisi dengan nilai median.

B. Dealing with Outliers

Pada data kategorikal, tidak ditemukan outlier. Sedangkan pada data numerik, outlier didefinisikan sebagai nilai yang berada di bawah Q1 - 1.5*IQR atau di atas Q3 + 1.5*IQR. Nilai-nilai outlier tersebut diganti dengan nilai bawah Q1 - 1.5*IQR dan nilai atas Q3 + 1.5*IQR. Adapun pada data binary, ternyata terdapat outlier (lebih tepatnya eror) pada fitur “is_ftp_login” dimana terdapat nilai selain 0,1 (yaitu 2 dan 4). Nilai ini diisi dengan new value yaitu -1

C. Remove Duplicates

Untuk mencegah redundansi, dilakukan pembersihan terdapat data duplikat. Ternyata tidak ditemukan data duplikat sama sekali sehingga tahapan ini dapat diabaikan

D. Feature Engineering

Kami melakukan pemilihan feature dengan cara memilih yang korelasinya sangat tinggi terhadap target. Kami juga menghilangkan highly-correlated feature untuk mencegah dua fitur yang “sama persis”.

E. Feature Scaling

Kami melakukan scalling pada data numerikal dengan metode min-max sehingga rentangnya menjadi 0-1.

F. Encoding Categorical Variables

Untuk data kategorikal selain attack_cat kami lakukan encoding dengan metode “one-hot encoding” sebab kategori tersebut bersifat nominal. Adapun fitur target dilakukan encoding dengan label encoding sebab fitur target harus tetap ada satu kolom.

G. Handling Imbalanced Classes

Untuk memaksimalkan data fitur target yang minoritas, serta mengefisiensikan data fitur target yang mayoritas, dilakukan fit dengan SMOTE dan RandomUnderSampler

H. Dimensionality Reduction

Kami melakukan dimensionality reduction dengan PCA, TSNE, dan Autoencoder. Pada akhirnya kami memilih autoencoder sebab waktu eksekusinya paling singkat.

Autoencoders adalah jenis jaringan saraf tiruan yang digunakan untuk mempelajari pengkodean efisien dari data yang tidak berlabel (pembelajaran tanpa pengawasan). Mereka berfungsi untuk mengompresi data input menjadi representasi berdimensi lebih rendah, dan kemudian merekonstruksi data dari representasi tersebut. Autoencoders sangat berguna untuk reduksi dimensi karena mereka dapat mengubah data berdimensi tinggi menjadi ruang laten berdimensi lebih rendah sambil mempertahankan fitur penting.

Autoencoders terdiri dari dua bagian utama: encoder yang mengompresi data input, dan decoder yang merekonstruksi data asli dari representasi berdimensi lebih rendah. Autoencoders dilatih untuk meminimalkan perbedaan antara data input dan data yang direkonstruksi, sering diukur menggunakan fungsi kerugian seperti Mean Squared Error (MSE). Autoencoders memiliki berbagai aplikasi, termasuk reduksi dimensi, pembersihan data (denoising), deteksi anomali, dan ekstraksi fitur.

I. Normalization

Normalisasi pada preprocessing menggunakan Yeo Johnson Transformation. Yeo-Johnson Transformation adalah teknik transformasi data yang digunakan dalam metode normalisasi untuk mengubah data yang tidak berdistribusi normal menjadi lebih mendekati distribusi normal. Transformasi ini dapat diterapkan pada data yang mengandung nilai positif maupun negatif, yang membuatnya lebih fleksibel dibandingkan transformasi Box-Cox yang hanya berlaku untuk nilai positif.

Proses Yeo-Johnson melibatkan penggunaan parameter lambda (λ) yang dioptimalkan untuk memaksimalkan likelihood dari data yang sudah ditransformasikan agar mendekati distribusi normal. Dengan mengubah bentuk distribusi data, Yeo-Johnson Transformation membantu meningkatkan performa model statistik atau

machine learning yang sensitif terhadap asumsi normalitas, serta membantu dalam mengurangi skewness dan memperbaiki homoscedasticity (kesamaan varians) dalam data.

II. Implementasi Model KNN

K-Nearest Neighbors (KNN) adalah salah satu algoritma machine learning yang paling sederhana dan sering digunakan untuk masalah klasifikasi dan regresi. Pada dasarnya, KNN bekerja dengan mencari sejumlah k tetangga terdekat (nearest neighbors) dari data yang akan diprediksi, kemudian menentukan kelas dari data tersebut berdasarkan mayoritas kelas dari tetangga terdekatnya.

Implementasi KNN

Implementasi KNN dari scratch dalam bahasa C++ ini mencakup beberapa komponen utama, yaitu:

- Kelas Data: Untuk menyimpan data fitur, target, dan jarak.
- Kelas Metrik: Untuk menghitung jarak antara dua vektor menggunakan berbagai metrik seperti Euclidean, Manhattan, dan Minkowski.
- Kelas KNN: Untuk melatih model, menghitung jarak, dan memprediksi kelas dari data baru.

Cara Kerja

Model akan melakukan fit dengan data hasil preprocessing dengan cara menghitung jarak dengan 3 opsi (euclidean, manhattan, dan minkowski). Setelah itu, data akan diurutkan membesar berdasarkan jarak masing masing features data terhadap data target. Setelah itu kita akan memilih sejumlah K data teratas untuk diambil sebagai hasil prediksi.

III. Implementasi Model Naive Bayes

Naive Bayes adalah algoritma machine learning berbasis probabilitas yang digunakan untuk klasifikasi. Algoritma ini mengasumsikan bahwa setiap fitur bersifat independen (naive assumption), meskipun dalam kenyataannya tidak selalu demikian.

Naive Bayes bekerja dengan menghitung probabilitas suatu kelas berdasarkan data latih, yaitu dengan mengombinasikan probabilitas awal (prior probability) kelas dengan probabilitas fitur yang muncul pada kelas tersebut (likelihood). Algoritma ini sederhana, cepat, dan sering digunakan dalam tugas seperti klasifikasi teks, seperti deteksi spam atau analisis sentimen.

Prosesnya terdiri dari beberapa langkah utama:

- Training Model (Fungsi fit_naive_bayes):

Data latih digunakan untuk menghitung probabilitas awal setiap kelas (class priors) dan kemungkinan nilai fitur muncul dalam setiap kelas (feature likelihoods).

- Probabilitas kelas dihitung dari proporsi data untuk setiap kelas.
- Probabilitas fitur dihitung untuk setiap fitur pada tiap kelas.

- Prediksi Probabilitas (Fungsi predict_proba_naive_bayes):

Probabilitas setiap sampel untuk setiap kelas dihitung berdasarkan probabilitas awal dan probabilitas fitur. Jika nilai fitur tidak ditemukan, digunakan probabilitas kecil (smoothing) untuk menghindari nilai nol.

- Prediksi Kelas (Fungsi predict_naive_bayes):

Kelas dengan probabilitas tertinggi untuk setiap sampel dipilih sebagai hasil prediksi.

IV. Implementasi Model ID3

Iterative Dichotomiser 3 (ID3) merupakan salah satu algoritma Decision Tree Learning untuk Machine Learning. Dengan memilih karakteristik terbaik di setiap simpul untuk mempartisi data tergantung pada information gain, algoritma ini membangun pohon secara rekursif. Tujuannya adalah untuk membuat subset akhir sehomogen mungkin. Dengan memilih fitur yang menawarkan pengurangan entropi atau ketidakpastian terbesar, ID3 mengembangkan pohon secara iteratif. Prosedur ini terus berlanjut hingga persyaratan penghentian terpenuhi, seperti ukuran subset minimum atau kedalaman pohon maksimum. Berikut langkah-langkah implementasinya:

- Perhitungan Entropi:

Entropi mengukur tingkat ketidakpastian dalam dataset. Jika sebuah node hanya memiliki satu kelas, entropinya adalah 0. Entropi digunakan untuk menilai tingkat "kemurnian" data.

- Pemecahan Data:

Dataset dibagi menjadi dua subset berdasarkan fitur tertentu dan nilai ambang (threshold). Satu subset berisi data yang memenuhi kondisi, sementara subset lainnya berisi data yang tidak memenuhi kondisi.

- Penghitungan Information Gain:

Information gain dihitung untuk setiap kemungkinan pembagian. Nilai ini menunjukkan seberapa besar ketidakpastian berkurang setelah data dibagi berdasarkan fitur tertentu.

- Pembangunan Pohon Secara Rekursif:

Proses dimulai dari akar pohon dengan memilih fitur dan nilai ambang yang memberikan information gain tertinggi. Kemudian dibuat dua cabang: satu untuk subset yang memenuhi kondisi dan satu untuk subset yang tidak. Proses ini diulang hingga semua data di node memiliki kelas yang sama atau tidak ada lagi information gain yang signifikan.

- Struktur Node:

Setiap node pada pohon menyimpan:

    - Fitur dan nilai ambang yang digunakan untuk membagi data.
    - Cabang hasil pembagian (data true dan false).
    - Label kelas jika node tersebut adalah node daun (leaf node).

- Prediksi:

Untuk mengklasifikasi, algoritma melintasi pohon dari akar ke daun berdasarkan nilai fitur sampel dan mengembalikan label kelas pada node daun.

V. Perbandingan Hasil Prediksi dengan Pustaka

Secara overall implementasi dengan scratch lebih buruk dari library karena library sudah di optimize secara advanced

Hasil prediksi kami tidak seakurat hasil prediksi dengan pustaka. Hal ini disebabkan oleh

VI. Kontribusi Anggota

NIM	Nama	Workload
13522015	Yusuf Ardian Sandi	• Setup • EDA • ID3 • Feature Engineering • Handling Imbalanced Data • Remove Duplicates
13522040	Dhidit Abdi Aziz	• Data Understanding (khusus binary features) • Handling Missing Data • Dealing with Outlier • Feature Encoding

		• Pipeline
13522092	Sa'ad Abdul Hakim	• Feature Scalling • Naive Bayes
13522120	M Rifki Virziadeili Harisman	• Normalization • Dimensionality Reduction • KNN

VII. Referensi

Baihaqiyazid,
https://medium.com/@baihaqiyazid16/data-transform-box-cox-yeo-johnson-2fd28735e5e

Geeksforgeeks
https://www.geeksforgeeks.org/iterative-dichotomiser-3-id3-algorithm-from-scratch/

VIII. Github

https://github.com/dhiabziz/Tubes2AI