IF3170 Inteligensi Artifisial

Tugas Besar 2:
Implementasi Algoritma Pembelajaran Mesin

Disusun Oleh:

13522016 Zachary Samuel Tobing

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

BAB I

DESKRIPSI PERSOALAN

Pembelajaran mesin merupakan salah satu cabang dari kecerdasan buatan yang memungkinkan sistem untuk belajar dari data dan membuat prediksi atau keputusan tanpa diprogram secara eksplisit.

Dataset **UNSW-NB15** adalah kumpulan data lalu lintas jaringan yang mencakup berbagai jenis serangan siber dan aktivitas normal. Pada tugas ini, Anda diminta untuk mengimplementasikan algoritma pembelajaran mesin yang telah kalian pelajari di kuliah, yaitu **KNN, Gaussian Naive-Bayes, dan ID3** pada dataset **UNSW-NB15**.

Untuk menghasilkan prediksi yang berkualitas, Anda diharuskan untuk melakukan beberapa tahap berikut ini (tahapan lebih lengkap dapat dilihat di template notebook):

Data Cleaning

Tahap ini bertujuan untuk membersihkan dataset dari nilai yang hilang (missing values), data duplikat, atau data yang tidak valid sehingga dataset siap digunakan untuk analisis.

Data Transformation

Transformasi data melibatkan langkah-langkah seperti encoding variabel kategori, normalisasi atau standarisasi fitur numerik, serta penanganan ketidakseimbangan data (imbalanced data) untuk memastikan data berada dalam format yang sesuai dengan algoritma pembelajaran mesin.

Feature Selection

Pemilihan fitur yang relevan bertujuan untuk mengurangi kompleksitas model, menghindari overfitting, serta meningkatkan kinerja model. Langkah ini melibatkan identifikasi fitur yang memiliki pengaruh signifikan terhadap variabel target.

Dimensionality Reduction

Jika dataset memiliki jumlah fitur yang besar, reduksi dimensi dapat digunakan untuk mengurangi dimensi tanpa kehilangan informasi penting. Teknik seperti Principal Component Analysis (PCA) sering digunakan pada tahap ini.

Modeling dan Validation

Pada tahap ini, algoritma pembelajaran mesin seperti K-Nearest Neighbors (KNN), Naive Bayes, dan ID3 (Iterative Dichotomiser 3) diterapkan pada dataset. Anda akan melatih model pembelajaran mesin yang akan **mengklasifikasi kategori attack (attack_cat)** berdasarkan fitur-fitur lain yang telah diberikan. Model yang telah dibuat divalidasi menggunakan metode seperti **train-test split** atau **k-fold cross-validation** untuk memastikan kinerja yang optimal.

BAB II

IMPLEMENTASI

2.1 KNN

KNN (K-Nearest-Neighbors) adalah algoritma *machine learning* yang digunakan untuk klasifikasi dan regresi. Data yang baru dikategorikan berdasarkan mayoritas label dari k tetangga terdekat yang ada dalam data. Pada tahap *training*, yang dilakukan hanya pemasukkan data, tidak ada pembuatan model. Tetangga terdekat ditentukan oleh algoritma perhitungan tertentu, pada kasus ini yaitu *Euclidean distance*.

Langkah-langkah umum yang menjadi implementasi KNN adalah sebagai berikut:

- Menentukan jumlah tetangga

  Dilakukan dalam parameter inisialisasi kelas KNN dengan nilai *default* berupa 3.

- Menghitung jarak

  Data baru dihitung jaraknya terhadap semua titik data lainnya menggunakan rumus perhitungan jarak *Euclidean*.

- Memilih k tetangga terdekat

  Data memilih indeks dari titik data lain yang jarak *Euclidean* paling dekat dengan jumlah k sesuai yang didefinisikan.

- Mendapatkan label

  Indeks k tetangga terdekat yang telah diperoleh diambil label-labelnya.

- Majority voting

  Nilai atribut kemudian ditentukan berdasarkan *majority voting*, yaitu nilai atribut yang paling sering muncul diantara k tetangganya.

2.2 Naive Bayes

Naive Bayes adalah algoritma *machine learning* yang juga digunakan untuk klasifikasi, menggunakan prinsip *teorema Bayes* bahwa fitur yang digunakan dalam suatu model bersifat independen (*naive*), tidak bergantung satu sama lain. Ini membuat algoritmanya efektif dan cepat.

Pada tahap *training*, algoritma ini akan memberikan model berupa peluang, yaitu peluang *prior* yaitu peluang kondisinya, dan peluang *likelihood* yaitu peluang nilai suatu fitur. Probabilitas ini kemudian digunakan pada tahap *testing*. Tahap *testing* menggunakan data tersebut untuk menghitung peluang *posterior* dan pengklasifikasian dilakukan dengan memilih kelas dengan peluang *posterior* tertinggi.

Langkah-langkah yang dilakukan sebagai berikut:

- Menghitung probabilitas prior

  Perhitungannya yaitu proporsi jumlah sampel kelas dalam seluruh dataset.

- Menghitung probabilitas likelihood

  Dilakukan dengan perhitungan jumlah kemunculan sampel pada fitur dan kelas dibandingkan dengan jumlah total data dengan kelas tersebut.

- Menghitung peluang *posterior*

  Dilakukan dengan rumus yang ada serta ditentukannya peluang yang paling besar untuk yang dipilih ketika mengklasifikasi data.

2.3 ID3

ID3 (Iterative Dichotomiser 3) adalah algoritma pembelajaran mesin yang digunakan untuk membangun pohon keputusan berdasarkan informasi gain dan entropi. ID3 membagi dataset ke dalam subset yang lebih kecil berdasarkan fitur yang memberikan pengurangan terbesar dalam ketidakpastian (entropy).

Langkah-langkah Implementasi:

1. Menghitung Entropi:

- Fungsi entropy(y) menghitung tingkat ketidakpastian dalam dataset berdasarkan distribusi kelas. Entropi yang lebih tinggi berarti ketidakpastian lebih besar.

2. Menghitung Information Gain:
   - Fungsi information_gain(X, y, feature) menghitung berapa banyak informasi yang didapatkan dengan membagi dataset berdasarkan fitur tertentu. Information gain diukur sebagai pengurangan entropi yang terjadi setelah pemisahan berdasarkan fitur tersebut.
3. Membangun Pohon Keputusan:
   - Fungsi _build_tree(X, y) secara rekursif membangun pohon keputusan. Langkah-langkahnya adalah:
     - Jika semua label sama: Tidak perlu membagi lebih lanjut, pohon mengarah ke label tersebut.
     - Jika tidak ada fitur tersisa: Pohon mengarah ke label yang paling umum dalam dataset.
     - Pilih fitur dengan Information Gain tertinggi: Fitur terbaik untuk membagi dataset adalah yang memberikan pengurangan entropi terbesar.
     - Pecah dataset berdasarkan nilai fitur terbaik: Setiap cabang pohon dibuat berdasarkan nilai yang mungkin dari fitur tersebut.
4. Prediksi dengan Pohon Keputusan:
   - Fungsi _predict_single(row) melakukan prediksi untuk satu contoh data. Proses prediksi mengikuti cabang pohon keputusan berdasarkan nilai fitur yang ada.

BAB III

CLEANING DAN PREPROCESSING

3.1 Cleaning

3.1.1 Missing Values

Missing values are treated differently for numerical and categorical data:

- Numerical data = uses **mean imputation** because the percentage of outliers is not too large and it would be a great representation.
- Categorical data = uses **mode imputation** because the number of columns for categorical data are few and the proportions make it suitable to apply it.

3.1.2 Outliers

Outliers are only treated for numerical data where it uses **clipping**, capping the values of the lower and higher range. The source code implements a calculation based on the skewness of the feature, adjusting dynamically for each feature.

3.1.3 Duplicate Entries

Duplicates are **removed** because it does not reflect the actual situation and is not relevant in the domain of computer networks.

3.1.4 Feature Engineering

The method used is **feature selection** where the parameter used for selection is the correlation coefficient compared to the target variable. This ensures that features with low to no correlation are removed from the process and only those with high correlation are selected.

3.2 Data Preprocessing

3.2.1 Feature Scaling

The method used is **robust scaling** because there are quite a bunch of outliers and using other methods would affect the scaling greatly, hence robust scaling is used, a method that is strong against outliers.

3.2.2 Feature Encoding

Feature encoding uses the **target encoding** method because the categorical features have high cardinality, lots of possible values in each feature, hence other encoding methods such as label and one-hot encoding would not be suitable for the data.

3.2.3 Handling Imbalanced Dataset

Resampling methods are used because it would modify the dataset into better input for data processing. **SMOTE** is used for oversampling because it interpolates the data, not randomly sampling the data. **Tomek Links** is used for undersampling to handle values near the separator between values.

3.2.4 Data Normalization

After having many methods clean the data, another scaling method which is **min max scaling** method is used to correctly normalize the data due to the features having a large range of values and the presence of outliers has become minimal.

3.2.5 Dimensionality Reduction

For dimensional analysis, **PCA** is used because of the large dataset, some of the features which have linear relationship, and to preserve the nature of the data. Other methods are far too complex and too expensive to be done for this dataset.

BAB IV

ANALISIS

4.1 Cleaning

Dalam pemrosesan tahapan *cleaning*, banyak sekali faktor yang harus dipertimbangkan. Dari domain, sifat data, kemampuan algoritma, sumber daya yang dibutuhkan, dsb. Langkah-langkah yang digunakan seharusnya bisa ditentukan yang terbaik, terlepas dari semua batasan yang ada seperti durasi pemrosesannya.

Setiap langkah dapat diperbaiki dan ditingkatkan, dicari kombinasi dari metode yang sesuai untuk mencapai data yang terbaik untuk dipelajari. Akan tetapi, dapat sekilas diasumsikan dan dilihat bahwa penanganan *missing data* dengan *mean* untuk *numerical* dan *mode* untuk *categorical* terlalu menyederhanakan dan tidak merefleksikan data yang sebenarnya.

Parameter dalam penanganan *outlier* dengan menggunakan *clipping* juga kurang ideal karena kebutuhan parameternya untuk perhitungan tidak dapat dengan pasti dihitung dan ditentukan. Fitur yang memiliki korelasi rendah dengan *target* belum tentu tidak relevan.

4.2 Preprocessing

Dengan jawaban serupa, banyak sekali metode yang dapat digunakan untuk memproses datanya. Hal yang dapat langsung terlihat yaitu ketika *split* biasa tidak cukup, melainkan perlu dicobanya *k-fold splitting* untuk hasil yang lebih baik.

BAB V

KESIMPULAN DAN SARAN

3.1 Kesimpulan

Dapat disimpulkan dari percobaan di atas bahwa hasilnya masih dapat ditingkatkan lagi agar memberikan hasil yang lebih baik. Banyak sekali faktor dalam pemrosesan data yang dapat mempengaruhi hasil, model, dan prediksi yang dikeluarkan oleh algoritma.

Langkah yang terkandung dalam pemrosesan data sangat banyak, dari *cleaning*, *preprocessing*, *splitting* yang masing-masing juga terdiri dari beberapa langkah yang cukup kompleks dan komprehensif. Setiap langkah tersebut juga memiliki banyak opsi atas langkah yang dapat diterapkan. Dalam penentuan opsi tersebut juga terdapat berbagai faktor sesuai algoritmanya, ditambah dengan domain dari dataset akan sangat mempengaruhi hasilnya. Dampaknya yang tidak selalu langsung terlihat dampaknya akan membuat prosesnya sangat sulit.

Oleh karena itu, analisis mendalam dan percobaan yang banyak perlu diterapkan agar dapat memberikan hasil yang terbaik.

3.2 Saran

Sebaiknya program yang dibuat dikembangkan lebih baik lagi agar memiliki hasil prediksi yang lebih baik.

BAB V

REFERENSI

Russell, S., & Norvig, P. (2016). *Artificial intelligence: A Modern Approach, Global Edition*.