Tugas Besar

IF3070 Dasar Inteligensi Artifisial

Implementasi Algoritma Pembelajaran Mesin

Dipersiakan oleh

Kelompok 5

Bihurin Salsabila Firdaus	(18222015)
Vini Putiasa	(18222030)
Regina Deva Carissa	(18222040)
Kezia Caren Cahyadi	(18222041)

PROGRAM STUDI SISTEM DAN TEKNOLOGI INFORMASI

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2024/2025

1. Data Cleaning and Preprocessing

A. Data Cleaning

I. Handling Missing Data

Pada proses ini, dilakukan imputasi nilai yang hilang pada kolom numerik menggunakan strategi mean. Kemudian, imputasi diterapkan pada data training menggunakan metode fit_transform. Kemudian, pada data validation, imputasi diterapkan menggunakan metode transform, yang menggunakan nilai rata-rata yang telah dihitung dari training set untuk mengisi nilai yang hilang. Setelah imputasi, hasilnya dicetak untuk memeriksa apakah nilai yang hilang pada kedua dataset telah berhasil diimputasi.

II. Dealing with Outliers

Dilakukan identifikasi dan penanganan outliers pada data numerik menggunakan metode Interquartile Range (IQR) dengan perhitungan seperti berikut:

```
Q1 = column.quantile(0.25)
Q3 = column.quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
```

Selanjutnya, untuk menangani outliers, dilakukan clipping pada nilai yang terdeteksi sebagai outlier, yaitu dengan mengubah nilai yang berada di bawah batas bawah (Q1 - 1.5 * IQR) menjadi nilai batas bawah dan yang berada di atas batas atas (Q3 + 1.5 * IQR) menjadi nilai batas atas.

III. Remove Duplicates

Dilakukan pengecekan duplikasi baris dalam DataFrame df dengan menggunakan fungsi duplicated(). Baris yang terdeteksi sebagai duplikat disaring dan ditampilkan dalam DataFrame duplicate_rows.

IV. Feature Engineering

Dilakukan perhitungan matriks korelasi untuk fitur numerik. Jika ada fitur yang memiliki korelasi sangat tinggi maka salah satu dari fitur tersebut akan dihapus.

B. Data Preprocessing

I. Feature Scaling

Kelas FeatureScaler digunakan untuk melakukan penskalaan (scaling) pada dataset. Terdapat dua jenis penskalaan yang diinginkan, yaitu StandardScaler (default) atau MinMaxScaler pada parameter scaling_type.

Min-Max Scaling (Normalization):

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Standardization (Z-score Scaling):

$$X' = \frac{X - \mu}{\sigma}$$

II. Feature Encoding

Kelas FeatureEncoder dibuat untuk mengkodekan fitur kategorikal menjadi format numerik dan dilakukan menggunakan teknik One-Hot Encoding. Metode fit digunakan untuk melatih encoder pada fitur kategorikal, sementara metode transform mengubah fitur kategorikal menjadi representasi numerik yang siap untuk digunakan dalam model machine learning.

III. Handling Imbalanced Dataset

Kelas HandleClassImbalance digunakan untuk menangani ketidakseimbangan kelas dalam dataset. Proses ini dapat dilakukan dengan dua metode, yaitu oversampling dan undersampling.

2. Modeling and Validation

KNN

N_neighbors merupakan jumlah tetangga terdekat yang digunakan untuk menentukan prediksi. Metric digunakan untuk menghitung kedekatan antar data (defaultnya digunakan 'euclidean'). Kemudian, pada predict, akan dihitung jarak antar dua titik menggunakan euclidean. Perhitungan tersebut menggunakan rumus berikut:

$$dis\ (x_1, x_2) = \sqrt{\sum_{i=0}^{n} (x_{1i} - x_{2i})^2}$$

Naive Bayes

Teorema Bayes:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

In our case:

$$P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)}$$

with Feature vector $X = (x_1, x_2, x_3, ..., x_n)$

Asumsikan fitur-fiturnya mutually independent

$$P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)}$$

$$\downarrow$$

$$P(y|X) = \frac{P(x_1|y) \cdot P(x_2|y) \cdot ... \cdot P(x_n|y) \cdot P(y)}{P(X)}$$

Pilih kelas dengan probabilitas posterior tertinggi

$$P(y|X) = \frac{P(x_1|y) \cdot P(x_2|y) \cdot ... \cdot P(x_n|y) \cdot P(y)}{P(X)}$$

$$\downarrow$$

$$y = argmax_y P(y|X) = argmax_y \frac{P(x_1|y) \quad P(x_2|y) \quad ... \quad P(x_n|y) \quad P(y)}{P(X)}$$

$$y = argmax_y P(x_1|y) \cdot P(x_2|y) \quad ... \cdot P(x_n|y) \cdot P(y)$$

$$y = argmax_y \log(P(x_1|y)) + \log(P(x_2|y)) + ... + \log(P(x_n|y)) + \log(P(y))$$

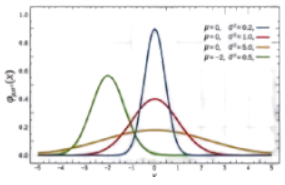
Prior and class conditional

$$y = argmax_y \log(P(x_1|y)) + \log(P(x_2|y)) + ... + \log(P(x_n|y)) + \log(P(y))$$

$P(y)$ Prior probability --> Frequency of each class

$P(x_i|y)$ Class conditional probability --> Model with Gaussian

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \cdot exp(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2})$$



Berikut merupakan langkah pembuatan modelnya:

Training:

- Calculate mean, var, and prior (frequency) for each class

Predictions:

- Calculate posterior for each class with

$$y = argmax_y \log(P(x_1|y)) + \log(P(x_2|y)) + ... + \log(P(x_n|y)) + \log(P(y))$$

and Gaussian formula

- Choose class with highest posterior probability

3. Hasil

Accuracy KNN (scratch): 0.9895
Accuracy KNN (scikit-learn): 0.9895
Accuracy Gaussian Naive-Bayes (scratch): 1.0
Accuracy Gaussian Naive-Bayes (scikit-learn): 0.994

Terlihat bahwa tingkat akurasi Naive-Bayes Naive-Bayes lebih tinggi dibandingkan KNN. Kemudian pada KNN, akurasi fungsi menggunakan scratch dan scikit-learn tidak berbeda. Bisa jadi sebenarnya terdapat perbedaan namun sangat kecil sehingga tidak terlihat jika hanya tiga angka di belakang koma. Lalu, untuk Gaussian Naive-Bayes, fungsi from scratch memiliki akurasi lebih tinggi dibandingkan yang scikit-learn.

Namun, tingkat akurasi tersebut juga tergantung dataframe yang digunakan dan nilai yang tinggi tersebut bisa jadi karena terjadinya data leakage.

Nama	**NIM**	**Kontribusi**
Bihurin Salsabila Firdaus	18222015	1. Melakukan data cleaning 2. Melakukan pre-processing
Vini Putiasa	18222030	1. Melakukan data cleaning 2. Membuat model KNN from scratch 3. Membuat model KNN menggunakan sklearn 4. Membuat model Naive Bayes from scratch 5. Membuat model Naive Bayes menggunakan sklearn

		6. Membuat implementasi dan testing 7. Membuat laporan
Regina Deva Carissa	18222040	belum
Kezia Caren Cahyadi	18222041	Nol