Tugas Besar 2 IF3070 Dasar Inteligensi Artifisial
Implementasi Algoritma Pembelajaran Mesin

Disusun Oleh:
Yoga Putra Pratama - 18222073

Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 40132

Daftar Isi

Penjelasan Implementasi KNN

Pada implementasi algoritma K-Nearest Neighbors (KNN), digunakan pustaka `sklearn` untuk melakukan klasifikasi. KNN adalah algoritma yang digunakan untuk mengklasifikasikan data berdasarkan kedekatannya dengan data lain yang sudah terlabeli. Dalam KNN, setiap data yang ingin diprediksi akan dihitung jaraknya terhadap data lainnya dalam ruang fitur, dan kelas yang paling banyak muncul pada k tetangga terdekatnya akan dipilih sebagai kelas prediksi.

Untuk implementasinya, digunakan kelas `KNeighborsClassifier` dari pustaka `sklearn.neighbors`. Sebelum melakukan pelatihan model, dilakukan pemisahan data menjadi data pelatihan dan pengujian menggunakan `train_test_split` dari `sklearn.model_selection`. Dalam parameter KNN, dipilih nilai `k` yang optimal melalui eksperimen, yang menentukan jumlah tetangga terdekat yang digunakan dalam proses prediksi. Fungsi `fit` digunakan untuk melatih model dengan data pelatihan, sementara fungsi `predict` digunakan untuk memprediksi label pada data pengujian.

Dalam penerapan KNN, salah satu tantangan utama adalah pemilihan nilai `k` yang tepat, karena nilai yang terlalu kecil bisa menyebabkan model menjadi sangat sensitif terhadap noise, sementara nilai yang terlalu besar bisa menyebabkan model kurang sensitif terhadap perbedaan kelas minoritas. Oleh karena itu, pemilihan nilai `k` harus melalui proses validasi silang (cross-validation) untuk mendapatkan hasil yang optimal.

Penjelasan Implementasi Naive-Bayes

Implementasi algoritma Naive-Bayes dilakukan dengan menggunakan kelas `GaussianNB` dari pustaka `sklearn.naive_bayes`. Naive-Bayes adalah algoritma klasifikasi berbasis teorema Bayes yang mengasumsikan bahwa setiap fitur independen satu sama lain (asumsi "naive"). Meskipun asumsi ini jarang berlaku dalam dunia nyata, Naive-Bayes tetap efektif dalam banyak kasus, terutama untuk data dengan banyak fitur kategorikal dan klasifikasi teks.

Pada implementasi ini, digunakan `GaussianNB` yang diasumsikan cocok untuk data kontinu, di mana setiap fitur mengikuti distribusi normal (Gaussian). Sebelum melatih model, data dibagi menjadi data pelatihan dan pengujian menggunakan `train_test_split`. Proses pelatihan dilakukan dengan menggunakan fungsi `fit`, dan prediksi dilakukan dengan menggunakan fungsi `predict`. Naive-Bayes bekerja dengan menghitung probabilitas kondisi dari setiap fitur, lalu mengalikan probabilitas tersebut untuk menentukan kelas yang paling mungkin.

Naive-Bayes memiliki keuntungan dalam hal kecepatan komputasi dan performa yang baik meskipun dengan asumsi yang tidak selalu valid. Namun, kekurangannya adalah ketidakmampuannya untuk menangani interaksi antar fitur secara langsung, yang dapat menjadi masalah pada dataset yang lebih kompleks.

Penjelasan Tahap Cleaning dan Preprocessing

Pada tahap cleaning dan preprocessing data, beberapa langkah penting dilakukan untuk memastikan kualitas data yang baik dan siap digunakan untuk model. Langkah pertama adalah menghapus data duplikat untuk menghindari bias yang dapat muncul jika data yang sama digunakan lebih dari satu kali dalam pelatihan model. Data duplikat dihapus menggunakan `drop_duplicates()` pada dataset.

Setelah itu, dilakukan penanganan missing values. Data yang hilang pada kolom numerik diimputasi dengan nilai rata-rata menggunakan metode `SimpleImputer` dari `sklearn.impute`, sedangkan data hilang pada kolom kategorikal diimputasi dengan nilai yang paling sering muncul (mode). Imputasi ini penting untuk menjaga integritas dataset dan menghindari kehilangan informasi yang berharga.

Selanjutnya, dilakukan encoding pada variabel kategorikal seperti `NoOfSubDomain` menggunakan `LabelEncoder` dari `sklearn.preprocessing`. Encoding ini mengubah variabel kategorikal menjadi numerik, yang penting agar dapat digunakan dalam model klasifikasi yang mengharuskan input numerik. Fitur baru juga dibuat, yaitu `url_length`, yang diambil dari panjang URL. Fitur ini ditambahkan karena panjang URL bisa memberikan informasi tambahan mengenai karakteristik dari situs yang dianalisis.

Penting untuk dicatat bahwa beberapa model, seperti KNN, memerlukan fitur yang diskalakan agar performa mereka optimal. Oleh karena itu, dilakukan penerapan feature scaling menggunakan `StandardScaler` untuk memastikan bahwa semua fitur berada dalam skala yang sama, menghindari fitur dengan skala yang lebih besar mendominasi perhitungan jarak.

Perbandingan Hasil Prediksi dengan Hasil Pustaka

Setelah menerapkan model KNN dan Naive-Bayes, hasil prediksi dari kedua algoritma dibandingkan dengan hasil yang didapat menggunakan pustaka seperti `sklearn.metrics` untuk mengevaluasi performa model. Evaluasi dilakukan dengan menggunakan metrik yang sesuai dengan masalah ini, seperti *accuracy*, *precision*, *recall*, dan *F1-score*.

Pada model KNN, dengan menggunakan validasi silang untuk memilih nilai k yang optimal, didapatkan hasil yang cukup baik, dengan F1-score yang lebih tinggi dibandingkan dengan Naive-Bayes, meskipun dengan sedikit overfitting pada kelas minoritas. Kelebihan KNN adalah kemampuannya untuk menangani data non-linear, sehingga dapat mengatasi beberapa kompleksitas dalam dataset.

Di sisi lain, Naive-Bayes, meskipun lebih cepat dan sederhana, tidak mampu menangani interaksi antar fitur dengan baik, yang menyebabkan performanya sedikit lebih rendah dibandingkan KNN pada dataset ini. Namun, Naive-Bayes menunjukkan keunggulan dalam hal waktu pelatihan yang lebih singkat dan lebih tahan terhadap noise pada data.

Dalam perbandingan ini, dapat disimpulkan bahwa KNN lebih cocok untuk dataset yang memiliki hubungan non-linear antar fitur dan memerlukan hasil yang lebih sensitif terhadap kelas minoritas, sedangkan Naive-Bayes lebih cocok untuk kasus di mana kecepatan dan kesederhanaan model lebih diutamakan, meskipun performanya sedikit lebih rendah. Secara keseluruhan, pemilihan model tergantung pada kebutuhan spesifik dari tugas klasifikasi ini, dan evaluasi yang lebih mendalam terhadap hyperparameter dan teknik ensemble bisa lebih meningkatkan hasil akhir.

Pembagian Tugas

NIM	Nama	Tugas
18222073	Yoga Putra Pratama	Source code, laporan

Referensi

Stuart J Russell & Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Prentice-Hall International, Inc, 2022.