PCA

1. Cara kerja algoritma PCA

Principal Component Analysis (PCA) adalah teknik statistik yang digunakan untuk mengurangi dimensi data dengan tetap mempertahankan sebanyak mungkin variabilitas yang ada di data asli. PCA melakukan transformasi data ke dalam set baru dari variabel yang disebut komponen utama (principal components), yang merupakan kombinasi linear dari variabel asli. Komponen utama diurutkan berdasarkan jumlah variabilitas yang mereka jelaskan dalam data. Langkah-langkah utama dalam algoritma PCA adalah sebagai berikut.

Standarisasi Data:

Langkah pertama dalam PCA adalah menstandarisasi data sehingga semua fitur memiliki rata-rata (mean) 0 dan variansi (variance) 1. Hal ini penting karena PCA sangat bergantung pada skala data, dan tanpa standarisasi, fitur dengan variabilitas lebih besar akan mendominasi komponen utama.

Membentuk Matriks Kovariansi:

Setelah data distandarisasi, langkah berikutnya adalah menghitung matriks kovariansi dari data. Matriks kovariansi ini menunjukkan seberapa banyak variabel berhubungan satu sama lain. Matriks ini membantu dalam menentukan arah di mana data memiliki varians terbesar.

Dekomposisi Eigen:

Dengan menggunakan matriks kovariansi, PCA selanjutnya melakukan dekomposisi eigen untuk menemukan eigenvalues (nilai eigen) dan eigenvectors (vektor eigen). Eigenvectors adalah arah (komponen utama) yang menunjukkan arah terbesar dari varians data, sementara eigenvalues menunjukkan besarnya varians dalam arah tersebut.

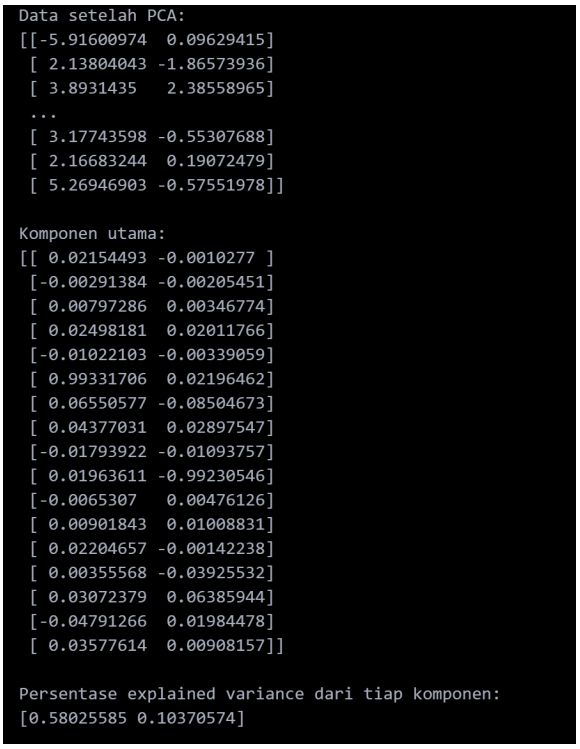
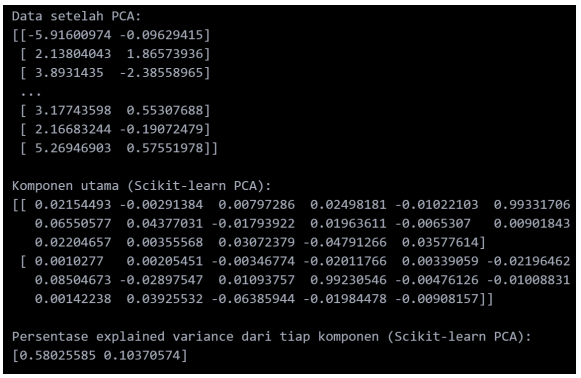
Urutkan Komponen Utama:

Komponen utama diurutkan berdasarkan nilai eigen dari terbesar ke terkecil. Komponen utama yang pertama (PC1) akan menjadi arah dengan varians terbesar, diikuti oleh komponen utama kedua (PC2), dan seterusnya. Jumlah komponen utama yang dipilih biasanya ditentukan berdasarkan seberapa banyak varians yang ingin dijelaskan oleh model.

Transformasi Data:

Terakhir, data asli diproyeksikan ke ruang baru yang dibentuk oleh komponen utama yang telah dipilih. Hal ini akan menghasilkan data baru yang berdimensi lebih rendah tetapi tetap mempertahankan sebagian besar informasi dari data asli.

4. Hasil Evaluasi Model

From Scratch	From Scikit-Learn
	

Hasil data setelah PCA dari implementasi scratch dan scikit-learn serupa dalam struktur, tetapi mungkin ada perbedaan kecil dalam nilai numerik karena perbedaan dalam cara implementasi dan pembulatan. Komponen utama (eigenvectors) dari PCAScratch dan SKPCA umumnya serupa, tetapi ada kemungkinan perbedaan kecil dalam tanda dan skala komponen. Hal ini dapat disebabkan oleh perbedaan dalam urutan eigenvectors (komponen utama) dan cara normalisasi yang diterapkan. Sementara itu, persentase explained variance yang diberikan oleh PCAScratch dan SKPCA cocok, menunjukkan bahwa kedua metode menghasilkan hasil yang konsisten dalam hal informasi yang dibawa oleh komponen utama.

Perbedaan kecil dalam hasil dapat disebabkan oleh perbedaan dalam precision numerik dan stabilitas algoritma, terutama jika menggunakan metode eigenvalue yang berbeda atau pemrosesan data. Implementasi manual mungkin memiliki perbedaan dalam cara penanganan data, normalisasi, dan cara menghitung matriks kovarians dan eigenvalues/eigenvectors dibandingkan dengan implementasi yang dioptimalkan seperti di scikit-learn. Sorting dari eigenvalues dan komponen utama serta penandaan (sign) dari komponen utama mungkin berbeda, yang dapat menyebabkan perbedaan dalam nilai komponen utama dan urutan.

5. *Improvement*

Untuk meningkatkan performa model dapat dilakukan optimasi hyperparameter dengan menggunakan grid search atau random search untuk menemukan hyperparameter terbaik, termasuk jumlah komponen PCA jika digunakan dalam model pembelajaran mesin. Kombinasi dengan teknik lain juga perlu dipertimbangkan. Jika data tidak linier, pertimbangkan untuk menggunakan kernel PCA yang dapat menangkap struktur non-linier dalam data. Optimasi juga dapat dilakukan pada implementasi PCA dengan menggunakan Singular Value Decomposition (SVD) yang seringkali lebih stabil secara numerik dibandingkan metode eigenvalue. Sementara untuk stabilitas numerik dapat menggunakan algoritma yang lebih stabil secara numerik untuk menghitung matriks kovarians dan eigenvalues. Terakhir, dapat juga menguji model dengan berbagai dataset untuk memastikan bahwa model yang dihasilkan dapat bekerja dengan baik pada berbagai data.