

## 1. Cara kerja algoritma PCA dari scratch pada repository ini

### a. Inisialisasi dan standarisasi data

Algoritma PCA dimulai dengan menerima data input yang terdiri dari beberapa fitur. Langkah pertama adalah menginisialisasi jumlah komponen utama (principal components) yang diinginkan oleh pengguna melalui parameter `n_components`. Kemudian, data input  $X$  di-center-kan dengan mengurangkan rata-rata dari setiap fitur. Itu sebagai standarisasi data yang bertujuan agar setiap fitur memiliki *mean* nol.

### b. Komputasi matriks kovarians

Setelah standarisasi data, algoritma menghitung matriks kovarians dari data yang sudah di-center-kan. Matriks kovarians dihitung menggunakan formula:

$$\text{Cov}(X) = \frac{1}{n-1} (X - \bar{X})^T (X - \bar{X})$$

di mana  $X$  adalah data yang di-center-kan.

### c. Dekomposisi eigendecomposition

Langkah berikutnya adalah melakukan dekomposisi eigendecomposition pada matriks kovarians untuk mendapatkan eigenvalues (nilai eigen) dan eigenvectors (vektor eigen).

### d. Memilih komponen utama

Setelah menghitung semua eigenvalues dan eigenvectors, eigenvalues diurutkan dari yang terbesar ke yang terkecil. Komponen utama atau principal components adalah eigenvectors yang sesuai dengan eigenvalues terbesar. Jumlah komponen utama yang dipilih ditentukan oleh parameter `n_components`. Persentase variansi yang dijelaskan oleh setiap komponen utama juga dihitung menggunakan formula:

$$\text{Explained Variance Ratio} = \frac{\text{Eigenvalue}_i}{\sum \text{Eigenvalue}}$$

### e. Proyeksi data ke dimensi baru

Langkah terakhir adalah memproyeksikan data asli ke dimensi baru menggunakan komponen utama yang telah dipilih. Proyeksi ini dilakukan dengan mengalikan data yang di-center-kan dengan eigenvectors yang dipilih. Proses ini seharusnya menghasilkan data baru dalam ruang komponen utama yang memiliki dimensi lebih rendah namun tetap mempertahankan sebanyak mungkin variansi dari data asli.

### f. Hasil PCA

Algoritma PCA mengembalikan dua hasil utama yaitu data yang telah ditransformasikan dalam ruang komponen utama dan persentase variansi yang dijelaskan oleh setiap komponen utama.

## 4. Perbandingan hasil evaluasi model

Berdasarkan hasil evaluasi model `PCA_Scratch` dan `PCA_Sklearn`, kedua model memberikan hasil yang hampir identik dalam hal explained variance ratio dan transformasi data. Explained variance ratio untuk kedua model adalah sama, yaitu sekitar 0.3435 untuk komponen utama

pertama (PC1) dan sekitar 0.1898 untuk komponen utama kedua (PC2). Grafik explained variance ratio juga menunjukkan pola yang identik, mengindikasikan bahwa kedua implementasi model ini berhasil mengekstraksi komponen utama dengan tingkat variabilitas yang sama dari dataset. Meskipun hasil akhir dari transformasi data sangat mirip, terdapat sedikit perbedaan pada tanda (positif atau negatif) dari nilai komponen utama. Hal ini mungkin terjadi karena orientasi eigenvector yang berbeda antara kedua implementasi. Namun, secara matematis, seharusnya hasilnya tetap setara karena PCA bersifat tidak sensitif terhadap perubahan tanda pada komponen utama. Dengan demikian, bisa dibilang tidak ada perbedaan signifikan dalam hasil yang diperoleh dari PCA\_Scratch dan PCA\_Sklearn dalam hal varians yang dijelaskan dan interpretasi komponen utama.

Perbedaan kecil atau tidak adanya perbedaan antara kedua model ini dapat dijelaskan oleh beberapa faktor. Pertama, algoritma PCA secara matematis sederhana dan langsung dalam menghitung eigenvalues dan eigenvectors dari covariance matrix data. Kedua, perhitungan yang dilakukan menggunakan pustaka NumPy sudah cukup stabil dan akurat, hampir setara dengan pustaka yang dioptimalkan seperti scikit-learn. Namun, implementasi scikit-learn mungkin menggunakan optimisasi tambahan di balik layar yang dapat meningkatkan efisiensi komputasi, terutama untuk dataset yang lebih besar atau kompleks.