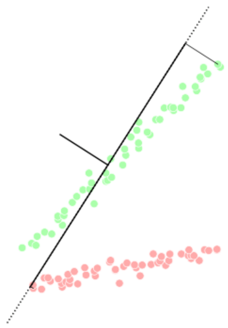
Dimensionality Reduction

PCA

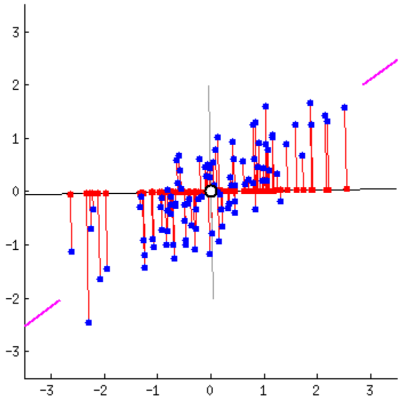
Presented by: Eng. Doaa Fahmy

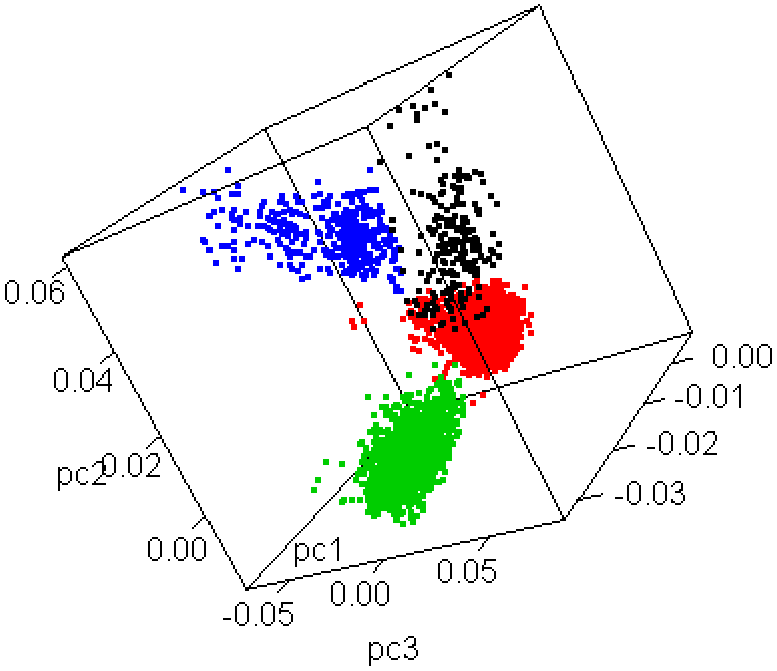
Principal Component Analysis (PCA)

is a dimensionality reduction technique commonly used in machine learning. It transforms a dataset with many features into a smaller set of uncorrelated features called principal components, while retaining most of the data's original variance.



PCA uses linear algebra to transform data into new features called principal components. It finds these by calculating eigenvectors (directions) and eigenvalues (importance) from the covariance matrix. PCA selects the top components with the highest eigenvalues and projects the data onto them simplify the dataset.





Applications of PCA

Dimensionality Reduction

Feature Extraction

PCA can be used to extract the most important features from a dataset, which can then be used as input for other machine learning model

Data Exploration

PCA can help visualize high-dimensional data by reducing it to two or three dimensions for plotting

Data Preprocessing

Image Compression

Advantages of Principal Component Analysis (PCA)

Multicollinearity Handling: Creates new, uncorrelated variables to address issues when original features are highly correlated.

Noise Reduction: Eliminates components with low variance enhance data clarity.

Data Compression: Represents data with fewer components reduce storage needs and speeding up processing.

Outlier Detection: Identifies unusual data points by showing which ones deviate significantly in the reduced space.

Disadvantages of Principal Component Analysis

1. **Interpretation Challenges**: The new components are combinations of original variables which can be hard to explain.

2. **Data Scaling Sensitivity**: Requires proper scaling of data before application or results may be misleading.

3. **Information Loss**: Reducing dimensions may lose some important information if too few components are kept.

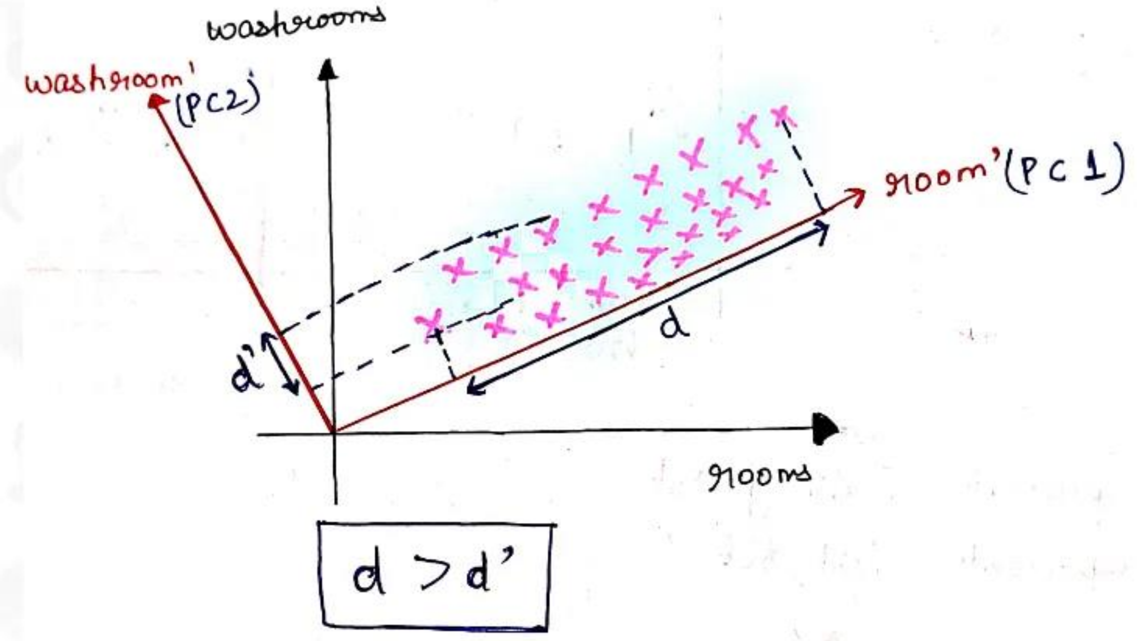
4. **Assumption of Linearity**: Works best when relationships between variables are linear and may struggle with non-linear data.

5. **Computational Complexity**: Can be slow and resource-intensive on very large datasets.

6. **Risk of Overfitting**: Using too many components or working with a small dataset might lead to models that don't generalize well.

How Principal Component Analysis Works ?

1. Standardize the data:
Ensure all features have a similar scale by standardizing the data, which prevents features with larger values from dominating the analysis.



2. Calculate the covariance matrix:

The covariance matrix represents the relationships between different features.

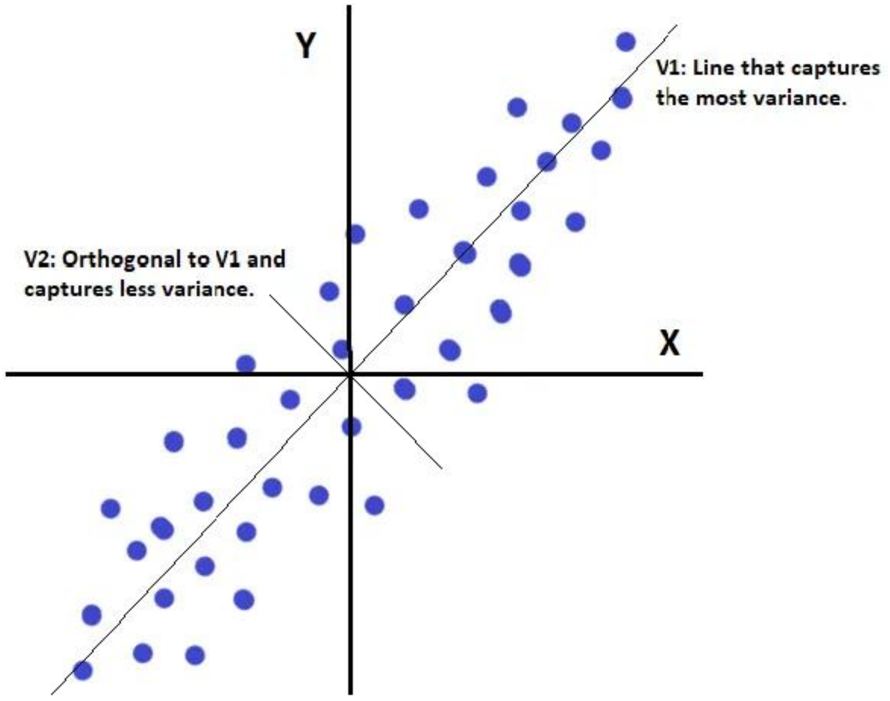
$$Var(X) = \frac{\Sigma(X_i - X)^2}{N} = \frac{\Sigma x_i}{N}$$

$$Cov(X, Y) = \frac{\Sigma(X_i - X)(Y_i - Y)}{N} = \frac{\Sigma x_i y_i}{N}$$

The covariance between two variables measures how they change together. The covariance matrix for a dataset with n features is an n x n matrix that summarizes the relationships between all pairs of features.

3. Compute the eigenvectors and eigenvalues:

Eigenvectors represent the principal components, and eigenvalues represent the amount of variance explained by each component



- **Eigenvalue**:

  An eigenvalue (λ) represents a scalar that indicates how much variance is explained by the corresponding eigenvector. In PCA, eigenvalues quantify the importance of each principal component. They are always non-negative, and the eigenvalue corresponding to a principal component measures the proportion of the total variance in the data explained by that component.

Eigenvector:

An eigenvector (v) is a vector associated with an eigenvalue. In PCA, eigenvectors represent the directions along which the data varies the most. Each eigenvector points in a specific direction in the feature space and corresponds to a principal component. Eigenvectors are typically normalized, meaning their length is 1

How Principal Component Analysis Works ?

4. Sort the eigenvectors:

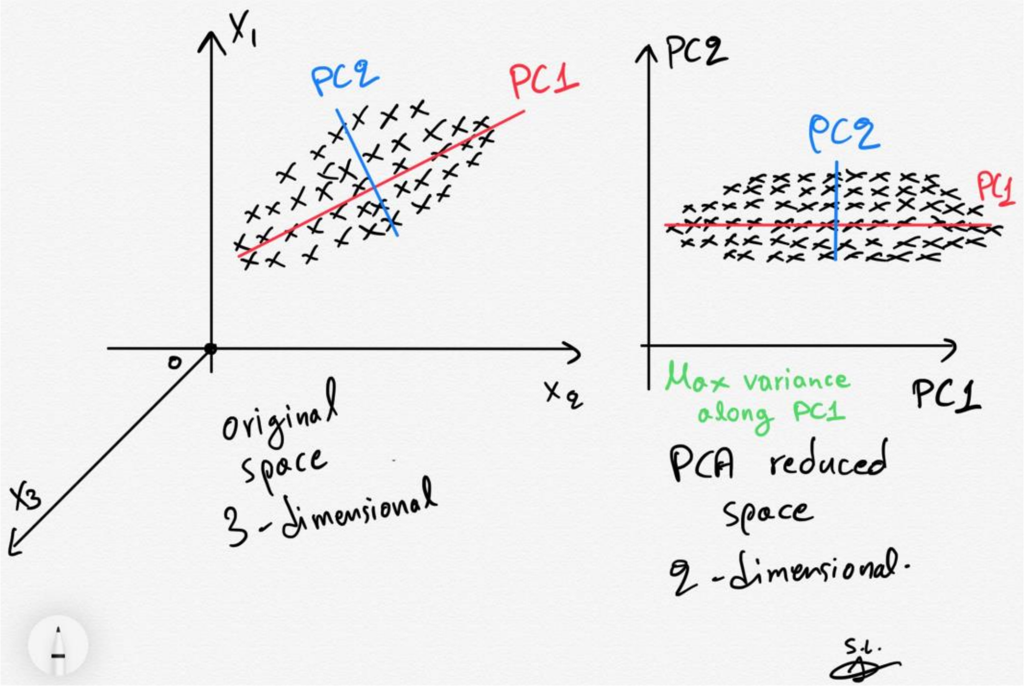
To identify the most significant principal components, sort the eigenvalues in descending order. The corresponding eigenvectors are also sorted accordingly. The first principal component explains the most variance, the second explains the second most, and so on.

5. Select the top k eigenvectors:

Choose the top k eigenvectors corresponding to the k largest eigenvalues to form the new feature subspace

6. Project the data:

Project the original data onto the new subspace defined by the selected eigenvectors



References

Hands-on Machine Learning, Aurelien Geron

Introduction to Machine Learning with Python, Andreas C. Muller