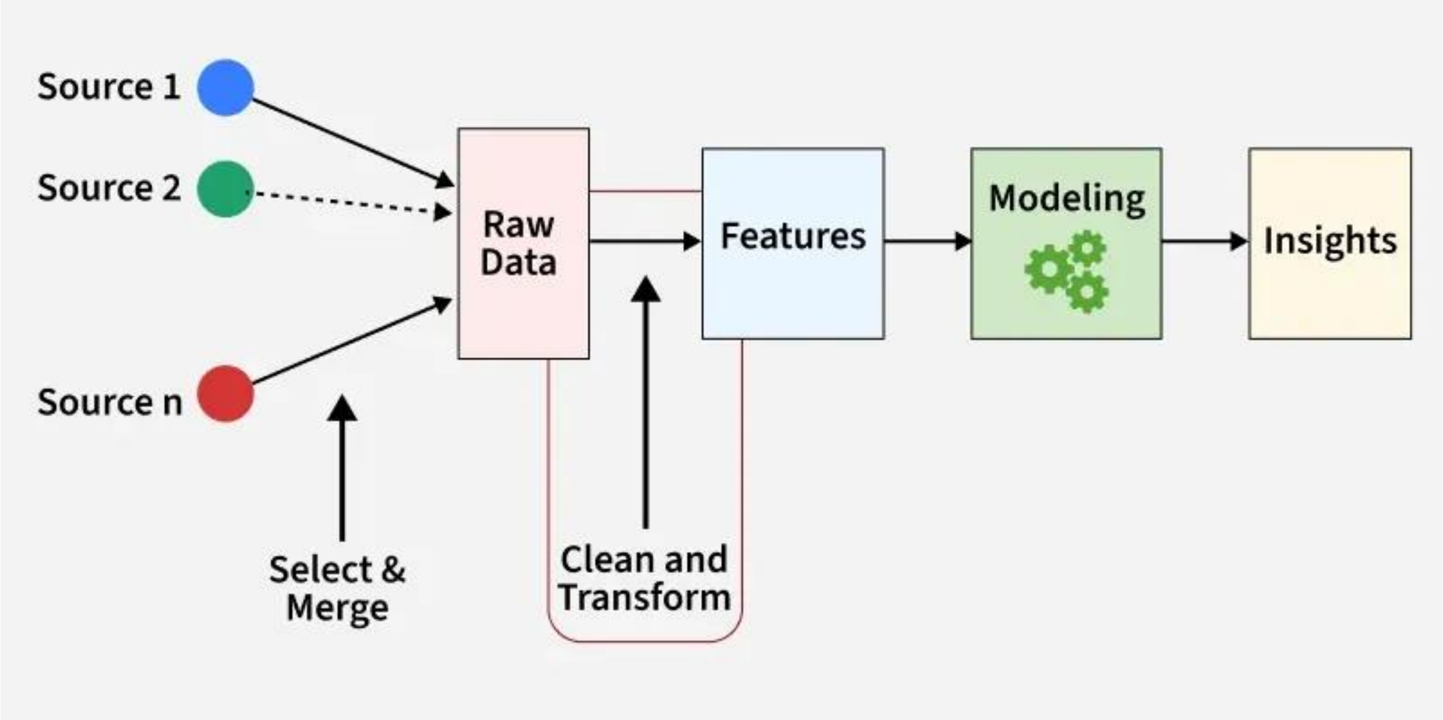
Feature Engineering

Feature Selection and Dimensionality Reduction

What is Feature Engineering?

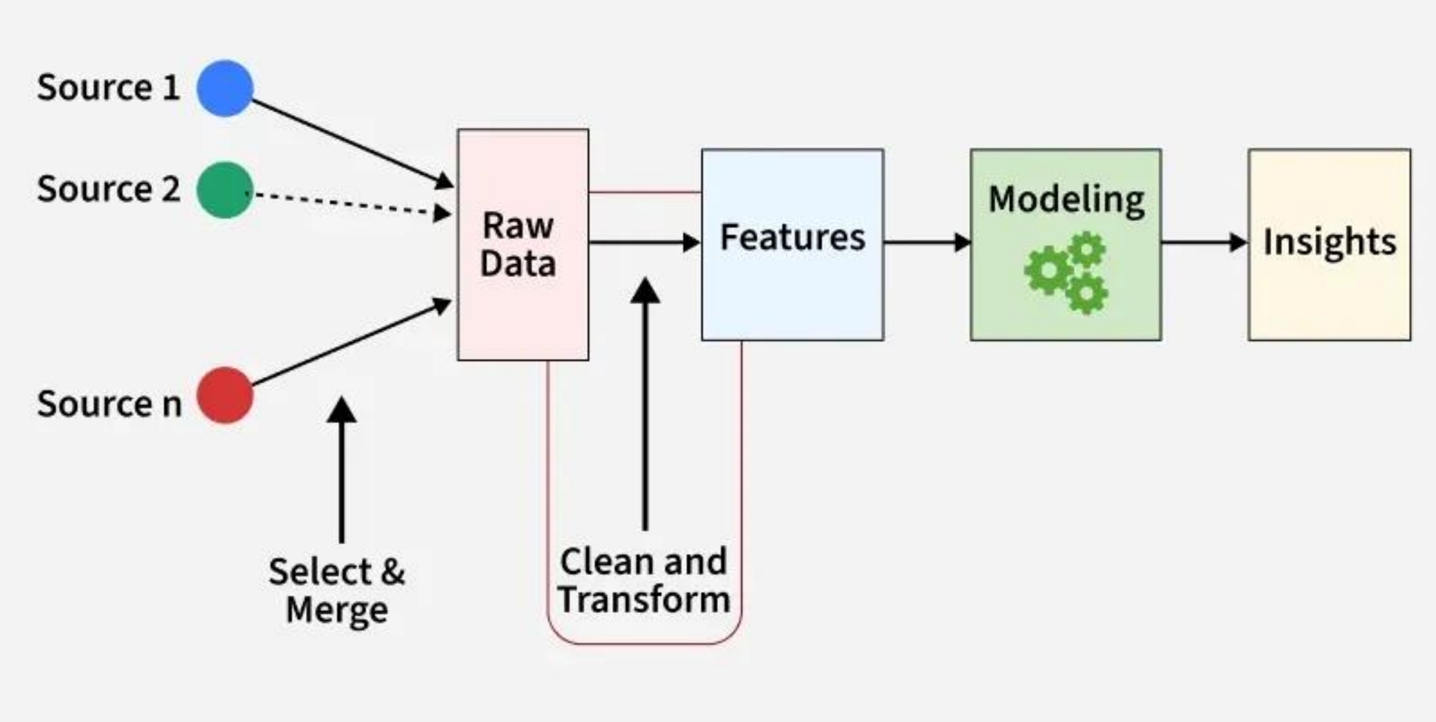
- Process of creating, transforming, and selecting input variables to improve model performance

- Involves:
  - Handling missing values
  - Encoding categorical data
  - Scaling and normalization
  - Derive new features



What is Feature Engineering?

- Better features → Better models
- Reduces bias/variance trade-off
- Makes patterns more learnable
- Example:
  - – Converting “Date” into:
    - **Day, Month, Year, Weekday**

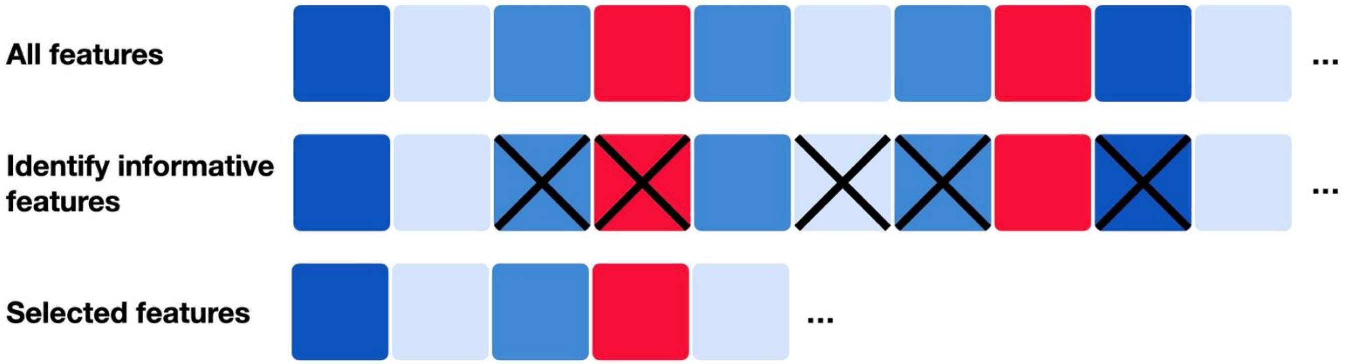


Feature Selection

- **Goal**: Keep only the most relevant features

- **Methods:**
  - **Filter**: Statistical tests (Chi-square, correlation, mutual information)
  - **Wrapper**: RFE (Recursive Feature Elimination)
  - **Embedded**: Lasso, Random Forest importance

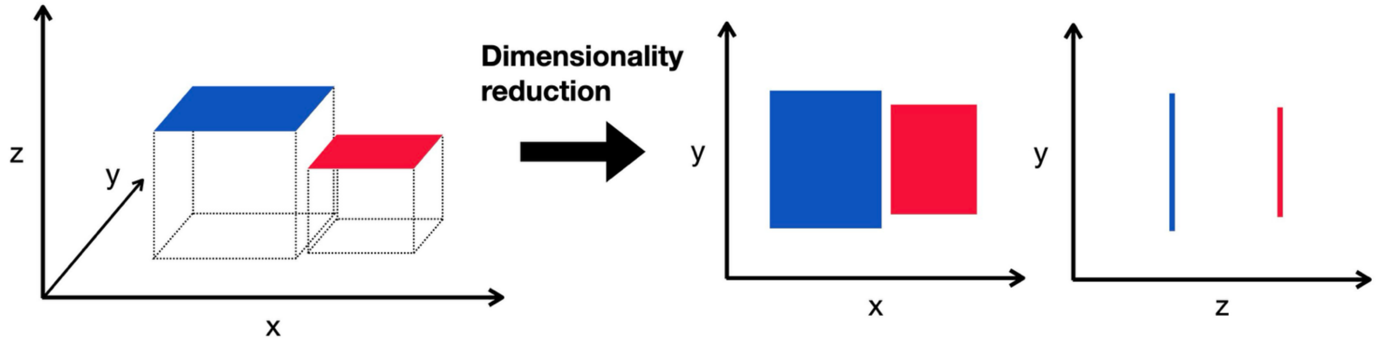
- Helps in interpretability & performance



Dimensionality Reduction

- **Definition:** Reducing number of features while preserving essential information

- **Why?**
  - Avoid overfitting
  - Improve computational efficiency



- Better visualization of data

- **Techniques:**
  - **PCA (Principal Component Analysis)**
  - **t-SNE, UMAP** (for visualization)

Putting it Together

- **Feature Engineering** is the broad process of **creating, transforming, or refining features** to make them more useful for machine learning.

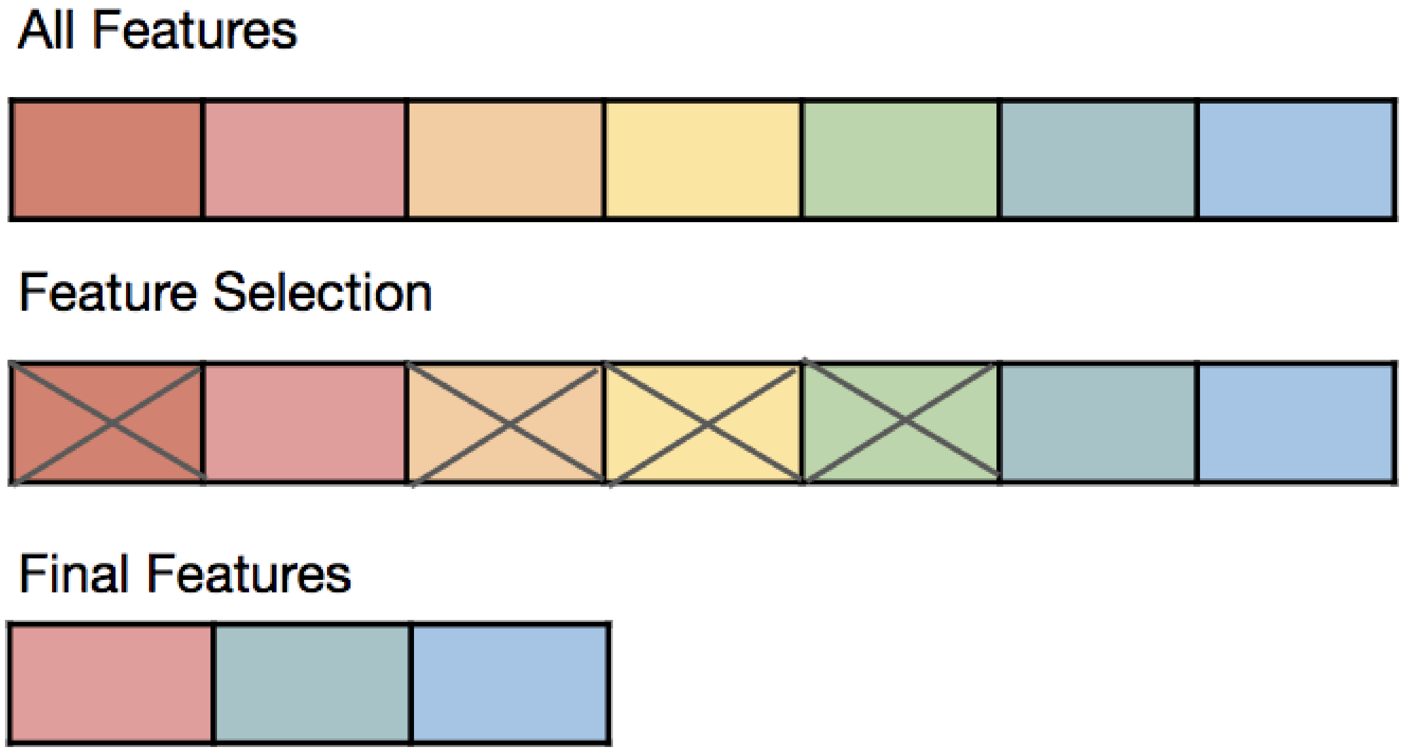
- **Feature Selection** and **Dimensionality Reduction** are **subtasks** within feature engineering that specifically focus on **reducing the feature space**.

- Pipeline:
  - Start with **Feature Engineering** → enrich raw data
  - Apply **Feature Selection** → remove irrelevant features
  - Use **Dimensionality Reduction** → compress into fewer dimensions
  - Combined → robust, interpretable, and efficient models

Feature Selection

Feature Selection

- Process of selecting essential features that are more uniform, non-redundant, and relevant for your ML model



All Features

Feature Selection

Final Features

Why do we need feature selection?

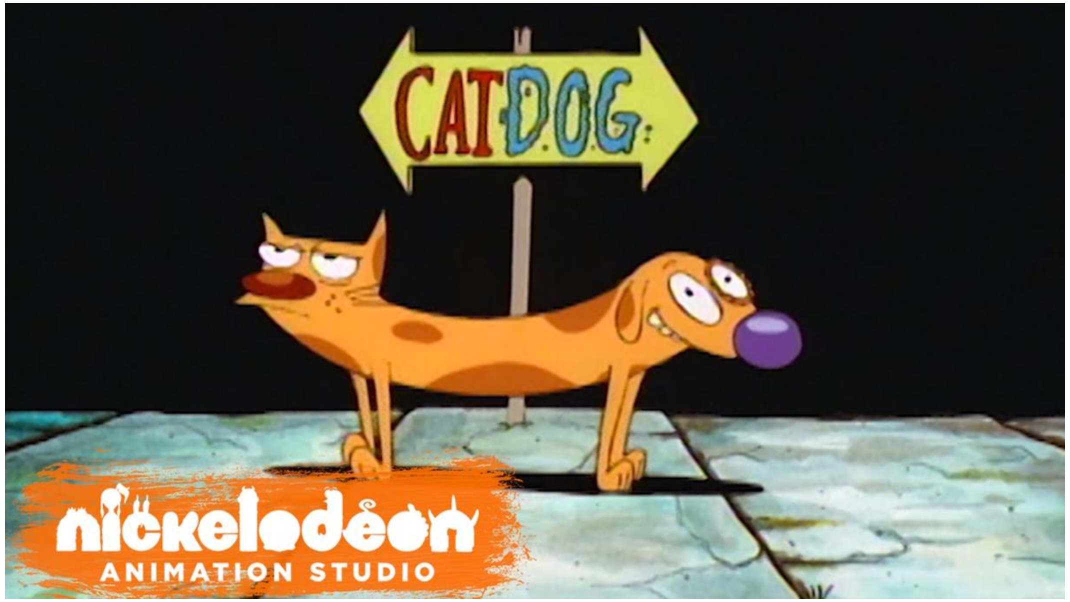
Example: CatDog Classifier

- Train a **linear classifier** to distinguish ***dogs*** from ***cats***
  - A set of images, each of which depicts either a cat or a dog

- Find a **descriptor** for each ***object class*** that can be expressed by numbers
  - Algorithm can use it to recognize the object
  - Example:

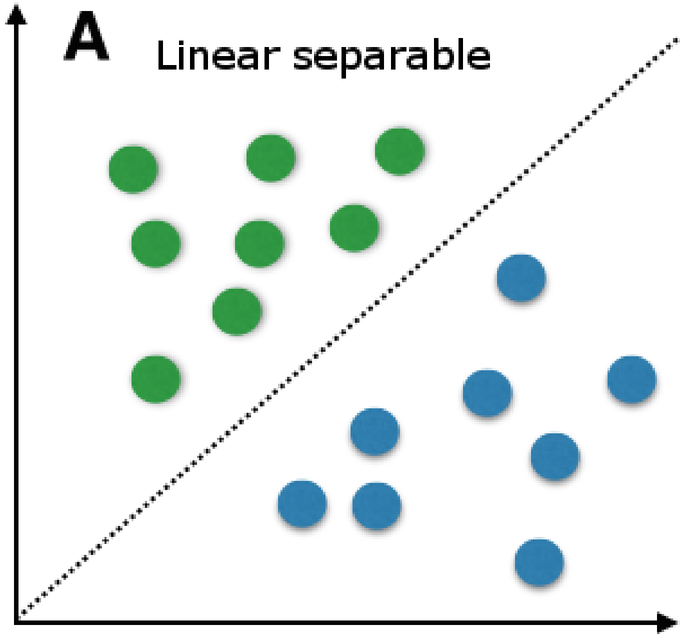
features

1. *the average red color*
2. *the average green color* and
3. *the average blue color* of the image under consideration



Train a Linear Classifier

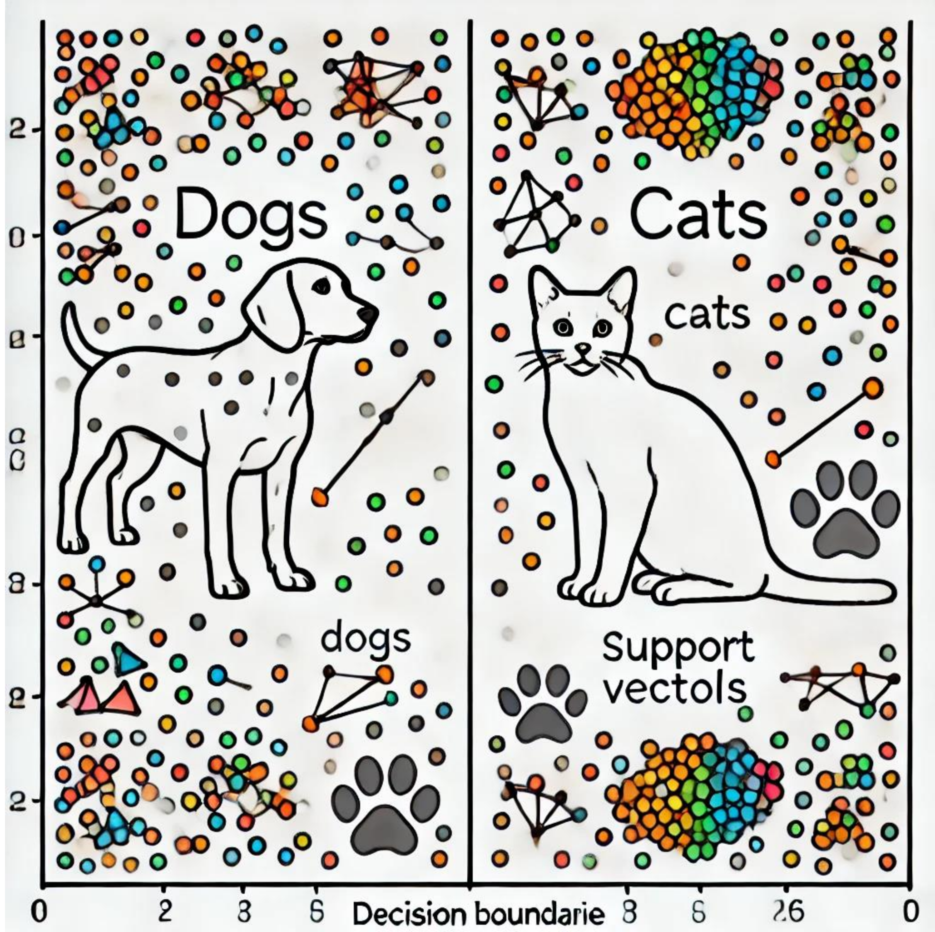
- Linear classifiers are called "linear" because they rely on linear relationships between the input features and the output class labels
  - Find the optimal hyperplane that separates the classes in the feature space



Inputs on one side of the hyperplane are classified into one category, while those on the other side are classified into another

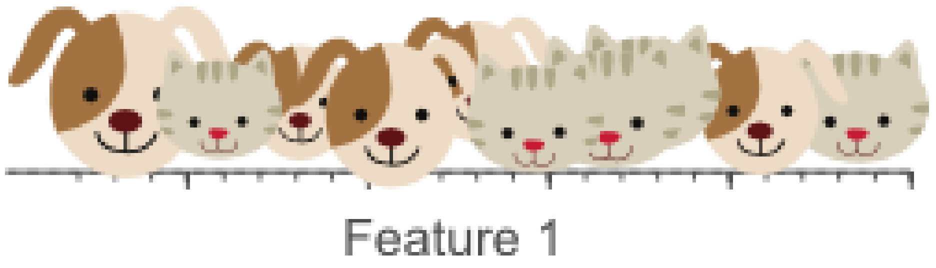
Train a Linear Classifier

- Train a **SVM** to distinguish ***dogs*** from ***cats***
  - The previous three features will not suffice!

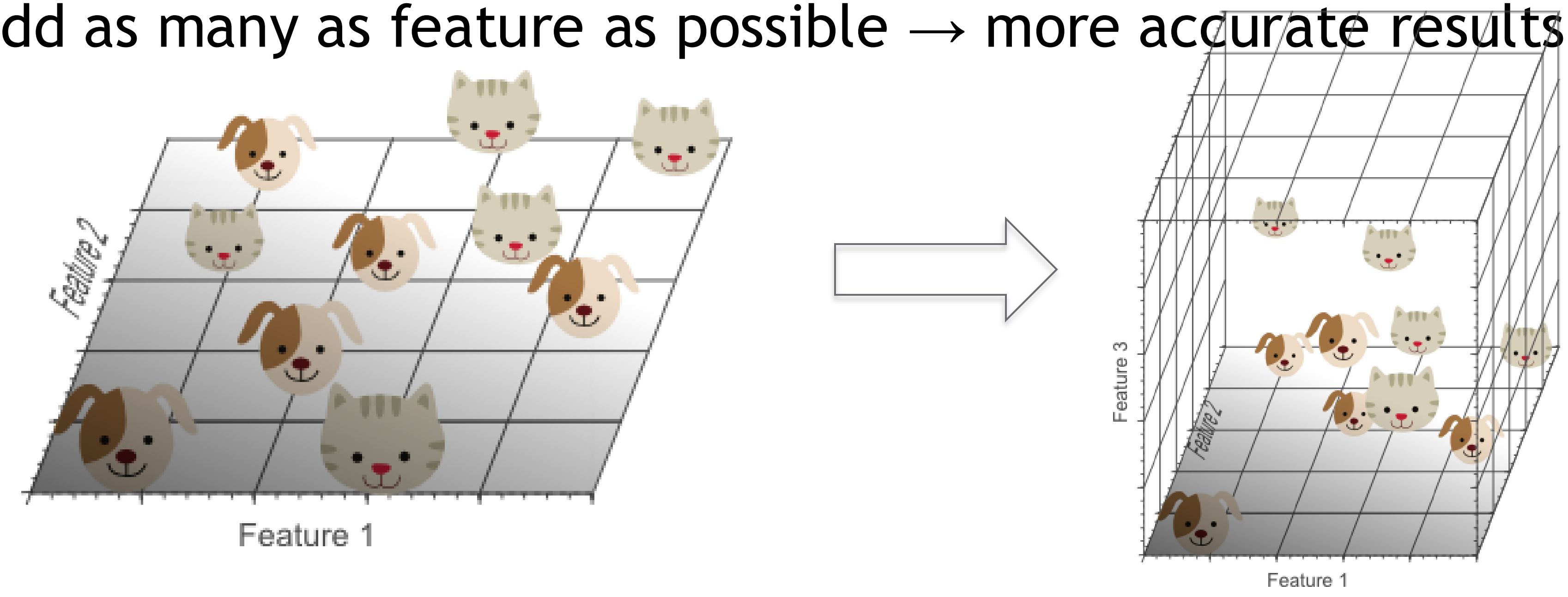


Curse of Dimensionality → Overfitting

- A single feature does not result in a perfect separation of our training data

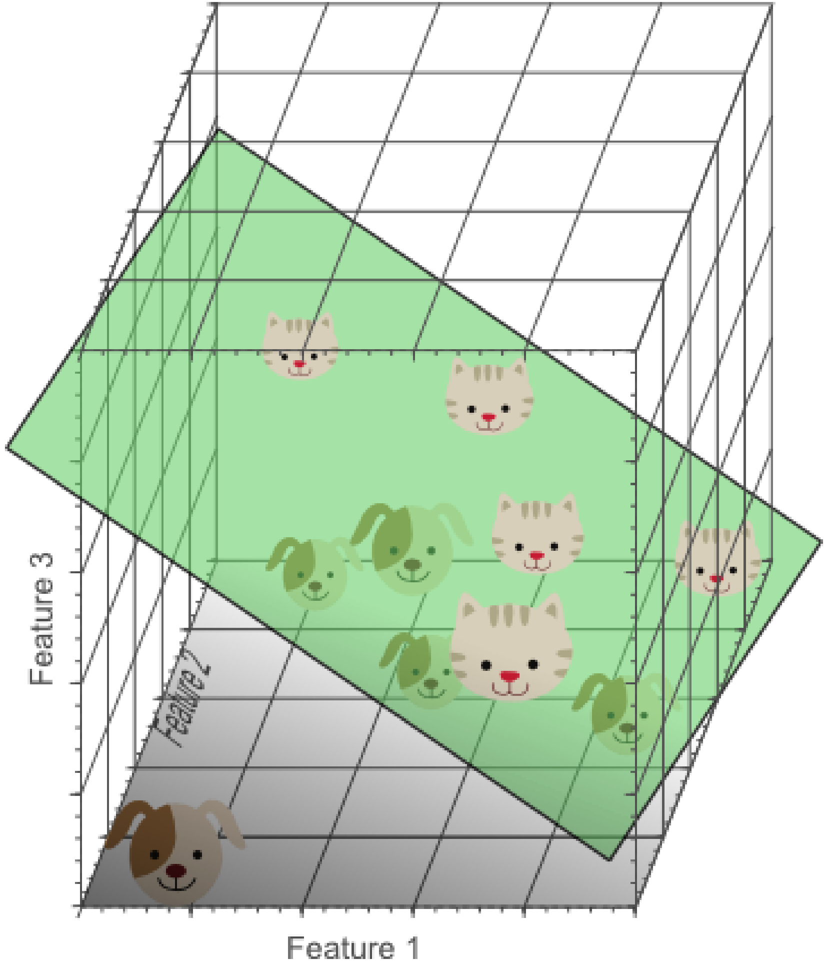


- Add as many as feature as possible → more accurate results



Curse of Dimensionality → Overfitting

- Adding a third feature results in a linearly separable classification problem
  - A plane exists that perfectly separates dogs from cats

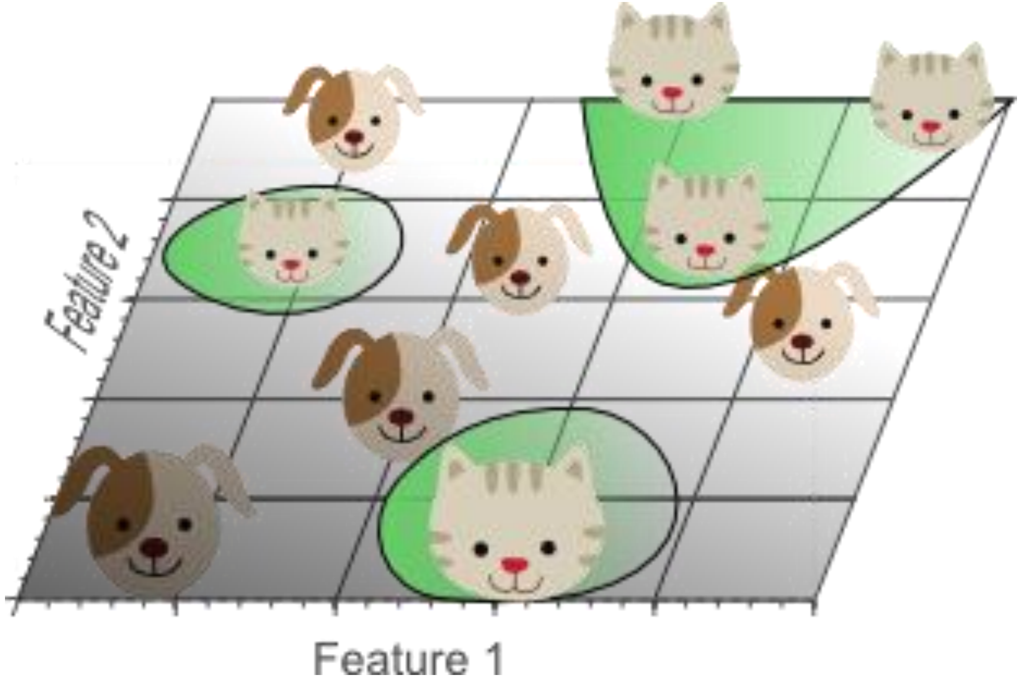


The more features we use, the higher the likelihood that we can successfully separate the classes perfectly

Curse of Dimensionality → Overfitting

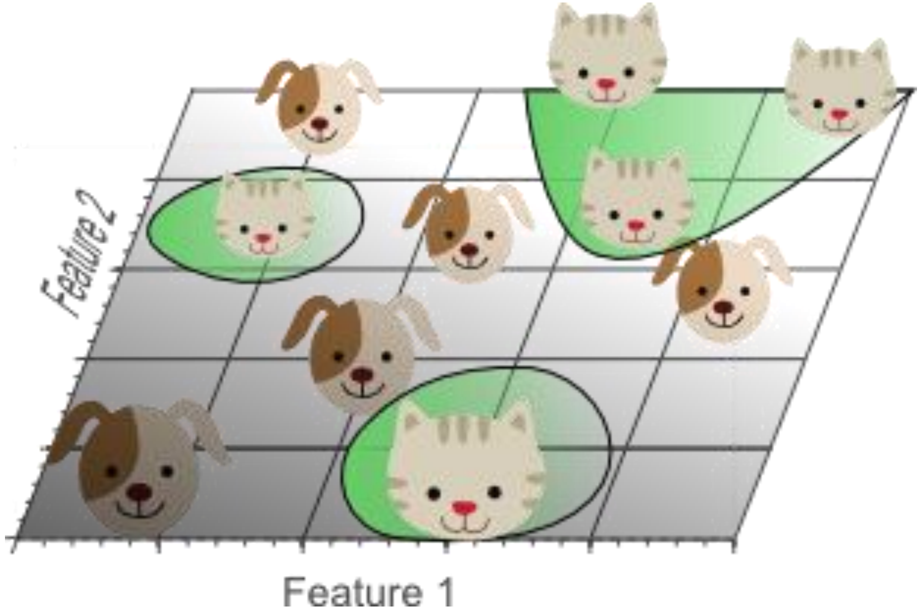
- The **dimensionality of the feature space** grows and becomes sparser and sparse if we keep adding features

- It becomes much easier to find a separable hyperplane
  - the likelihood that a training sample lies on the wrong side of the best hyperplane becomes infinitely small when the number of features becomes infinitely large



Curse of Dimensionality → Overfitting

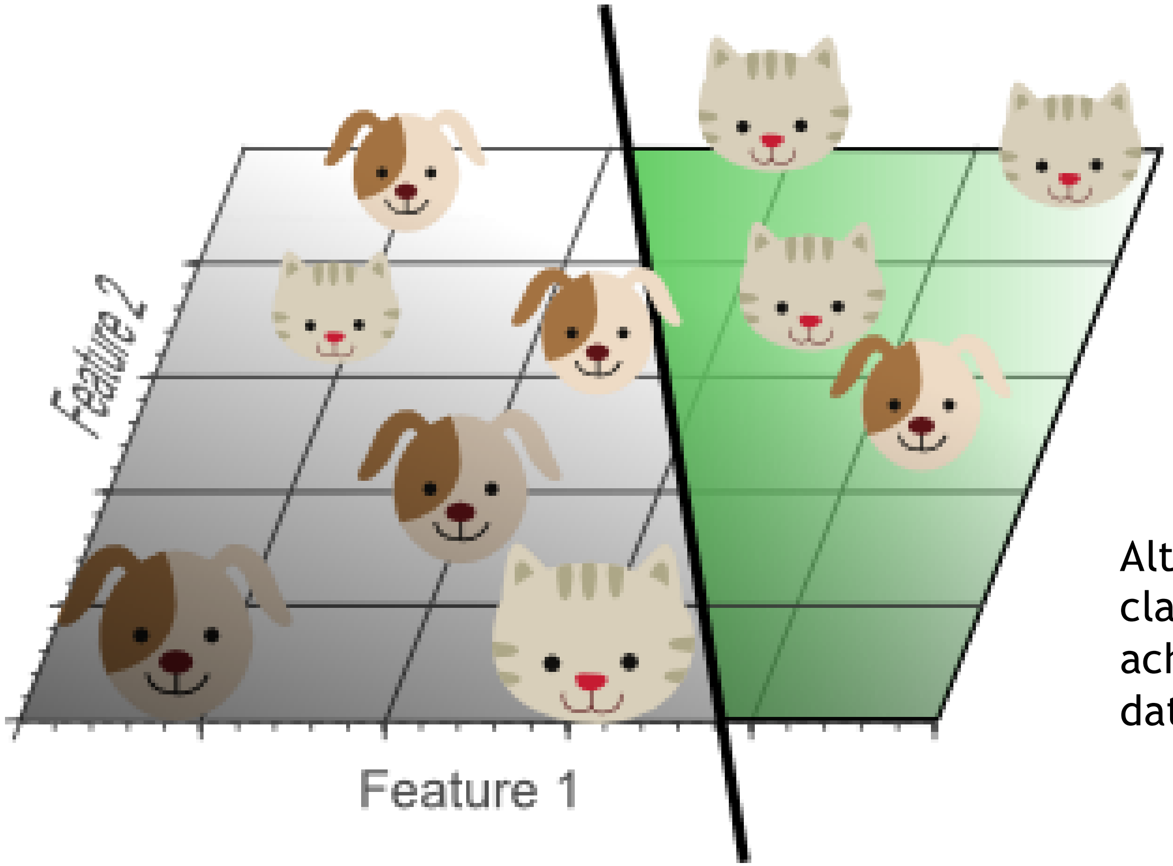
- After a certain point, the performance of the model will decrease with the increasing number of features



- Using too many features results in overfitting
  - Classifier starts learning exceptions that are specific to the training data
  - Classifier does not generalize well when new data is encountered

Important Features Only

- Linear classifier that has been trained using only 2 features instead of 3



Although the training data is not classified perfectly, this classifier achieves better results on unseen data

Feature Selection: Benefits

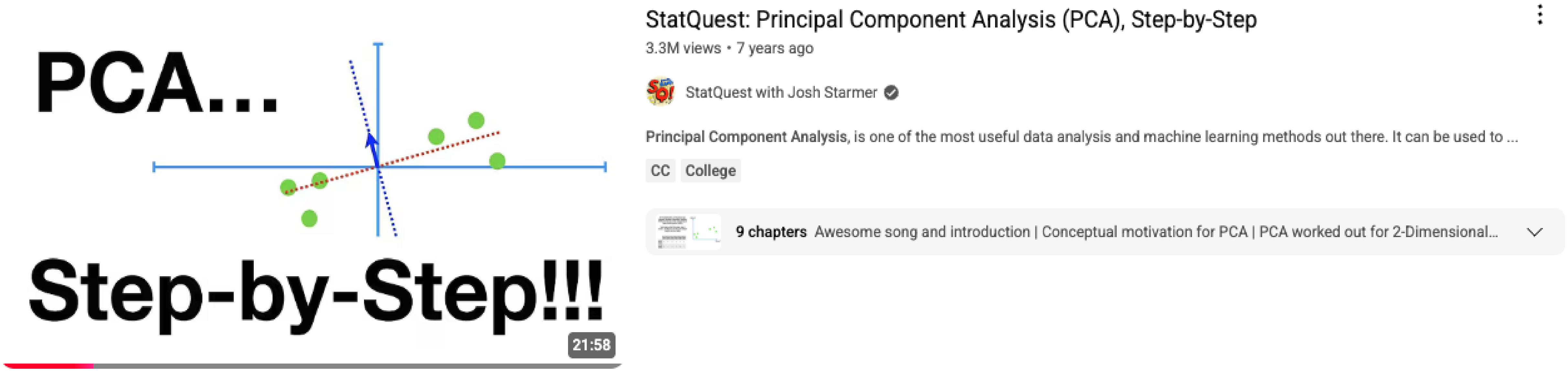
- It helps to reduce the size and complexity of the dataset
  - Consequently, less computation cost to train models

- Simple ML models with lesser features are easier to understand and explain

- It helps to avoid overfitting
  - More features → complex models → curse of dimensionality

Dimensionality Reduction

Watch this Youtube Video



StatQuest: Principal Component Analysis (PCA), Step-by-Step

3.3M views • 7 years ago

StatQuest with Josh Starmer

Principal Component Analysis, is one of the most useful data analysis and machine learning methods out there. It can be used to ...

CC College

9 chapters Awesome song and introduction | Conceptual motivation for PCA | PCA worked out for 2-Dimensional...

Dimensionality Reduction

- Dimensionality reduction is the process of reducing the number of features (or dimensions) in a dataset while retaining as much information as possible.

- This can be done for a variety of reasons,
  - such as to reduce the complexity of a model,
  - to improve the performance of a learning algorithm,
  - or to make it easier to visualize the data.

Curse of Dimensionality

High-Dimensional Data

In machine learning, high-dimensional data refers to data with a large number of features or variables.

Model Complexity

The complexity of the model increases with the number of features, and it becomes more difficult to find a good solution.

Overfitting

High-dimensional data can also lead to overfitting, where the model fits the training data too closely and does not generalize well to new data.

What is Dimensionality Reduction?

High-Dimensional Data

Datasets often have many features, making them difficult to analyze.

Reduced Dimensions

Dimensionality reduction transforms data into a **lower-dimensional space.**

Preserving Information

This process aims to retain as much important information as possible.

Benefits of Dimensiolaty Reduciton

Reduced Complexity

Dimensionality reduction can simplify the model, making it easier to understand and interpret.

Improved Generalization

By reducing complexity, the model can generalize better to new data.

Dimensionality Reduction Tecniques

1

1. Principal Component Analysis (PCA)

PCA is a widely used technique that identifies the principal components, which are the directions of greatest variance in the data.

2

2. Singular Value Decomposition (SVD)

SVD is a matrix factorization technique that decomposes a matrix into three matrices, revealing the underlying structure of the data.

3

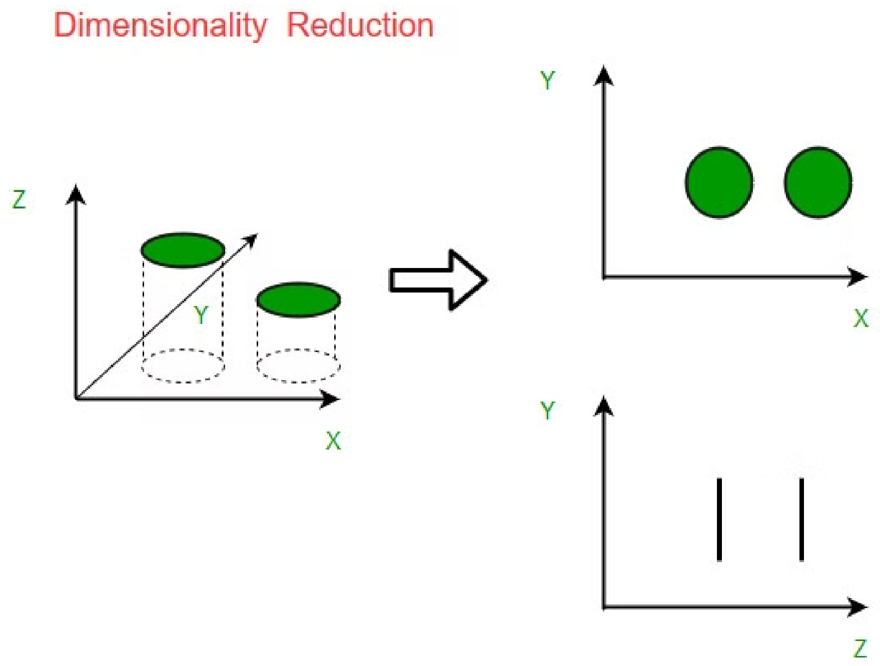
3. Linear Discriminant Analysis (LDA)

LDA is a supervised technique that aims to find the linear combinations of features that best separate different classes.

Importance in Machine Learning

- Dimensionality reduction is crucial in machine learning and predictive modeling as it simplifies complex datasets by reducing the number of features.

- This simplification improves model performance, reduces computational cost, and enhances interpretability.

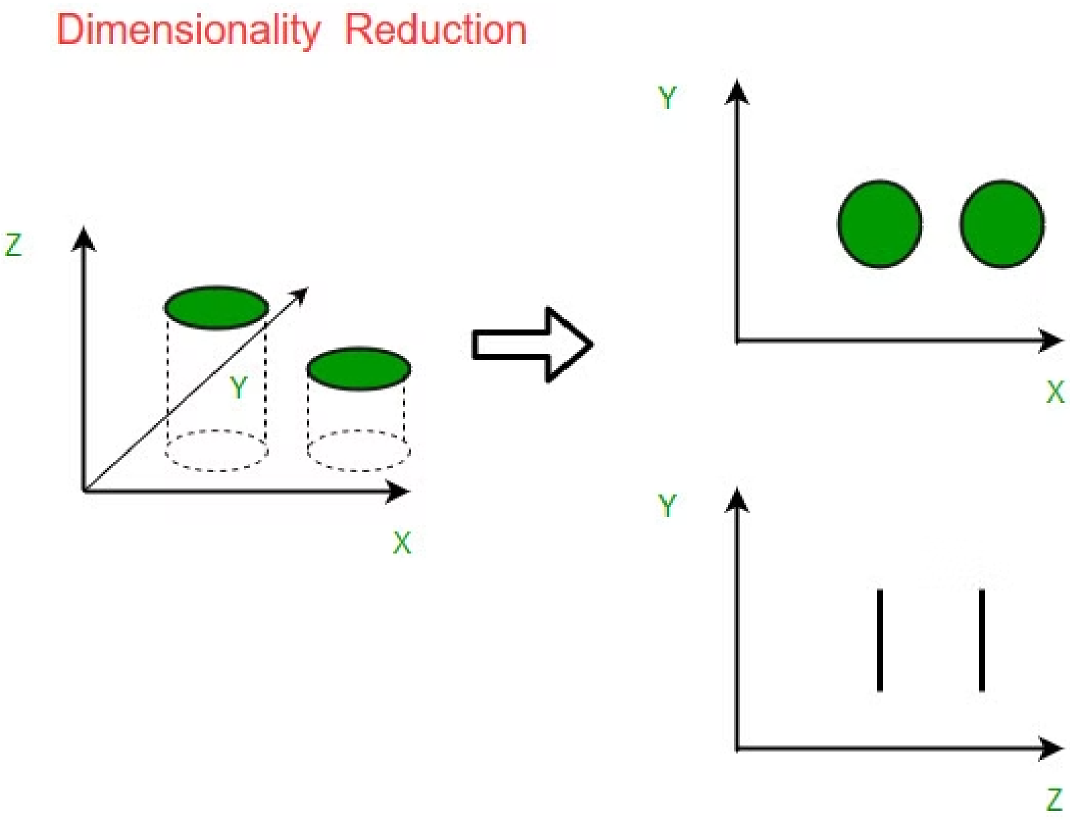


Dimensionality Reduction Example

- In email classification (spam vs. not spam)
  - many features like generic titles, content, and templates can overlap
  - Similarly, in weather prediction, features like humidity and rainfall may be highly correlated

- Reducing such overlapping or correlated features simplifies the problem

- For example, a complex 3D problem is harder to visualize but reducing it to 2D or even 1D makes it more manageable.



The figure below illustrates this by splitting a 3D feature space into two 2D spaces, and further reduction is possible when features are correlated.

Components of Dimensionality Reduction

Feature Selection

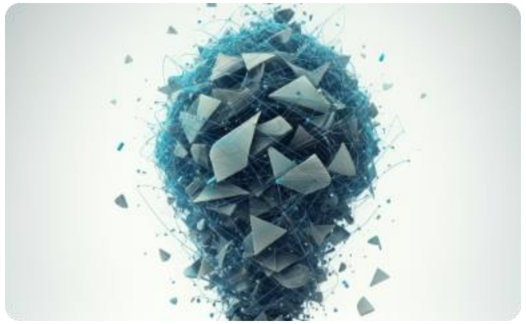
This involves finding a subset of the original set of variables, or features, to get a smaller subset which can be used to model the problem.

1. Filter
2. Wrapper
3. Embedded

Feature Extraction

This reduces the data in a high dimensional space to a lower dimension space, i.e. a space with lesser no. of dimensions.

Advantages of Dimensonality Reduction



Data Compression

Dimensionality reduction helps compress data, reducing storage space.

Reduced Computation Time

Dimensionality reduction reduces computation time by simplifying data.

Improved Visualization

Dimensionality reduction allows for easier visualization of high-dimensional data.

Overfitting Prevention

Dimensionality reduction can help prevent overfitting in machine learning models.

Disadvantages of Dimensionality Reduction

Data Loss

Dimensionality reduction can lead to some amount of data loss.

Linear Correlations

PCA tends to find linear correlations between variables, which is sometimes undesirable.

Interpretability

The reduced dimensions may not be easily interpretable, and it may be difficult to understand the relationship between the original features and the reduced dimensions.

Overfitting

In some cases, dimensionality reduction may lead to overfitting, especially when the number of components is chosen based on the training data.

Key Takeaways

Techniques

Techniques include principal component analysis (PCA), singular value decomposition (SVD), and linear discriminant analysis (LDA).

Information Preservation

Each technique projects data onto a lower-dimensional space while preserving important information.

Pre-processing Stage

Dimensionality reduction is performed during the pre-processing stage before building a model to improve performance.

Potential Loss

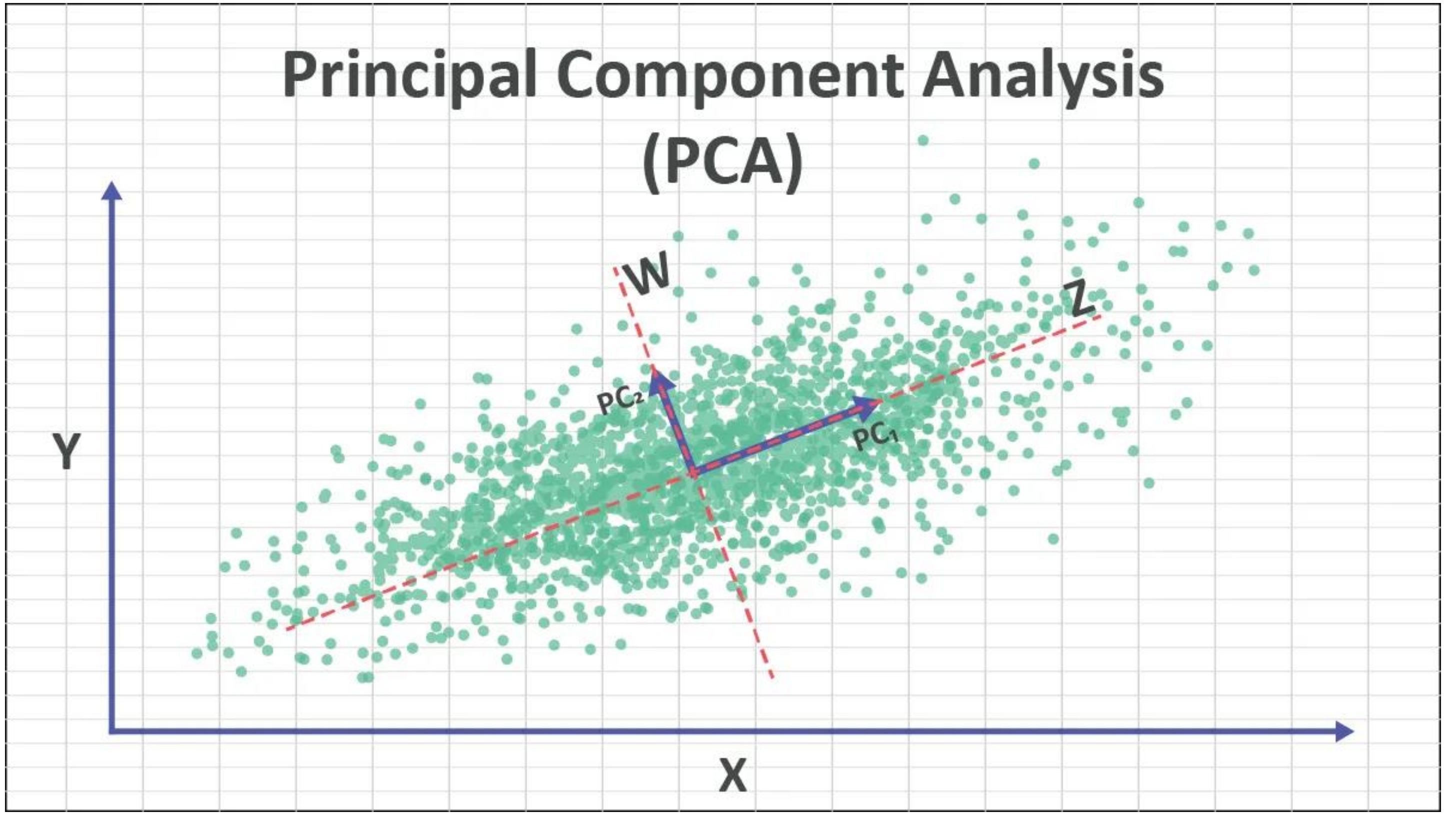
It is important to note that dimensionality reduction can also discard useful information, so care must be taken when applying these techniques.

Principal Component Analysis

Principal Component Analysis (PCA)

- PCA is statistical technique used for **dimensionality reduction**

- It transforms high-dimensional data into a **lower-dimensional representation** by identifying the principal components, which **capture the most variance in the data**



Understanding Dimensionality Reduction

Dimensionality reduction aims to simplify data by reducing the number of variables while preserving as much information as possible

Original Data

Data with many variables (dimensions) can be difficult to analyze and visualize.

Reduced Data

PCA finds new variables (principal components) that capture most of the variance, reducing the dimensionality.

Covariance and Correlation Matrices

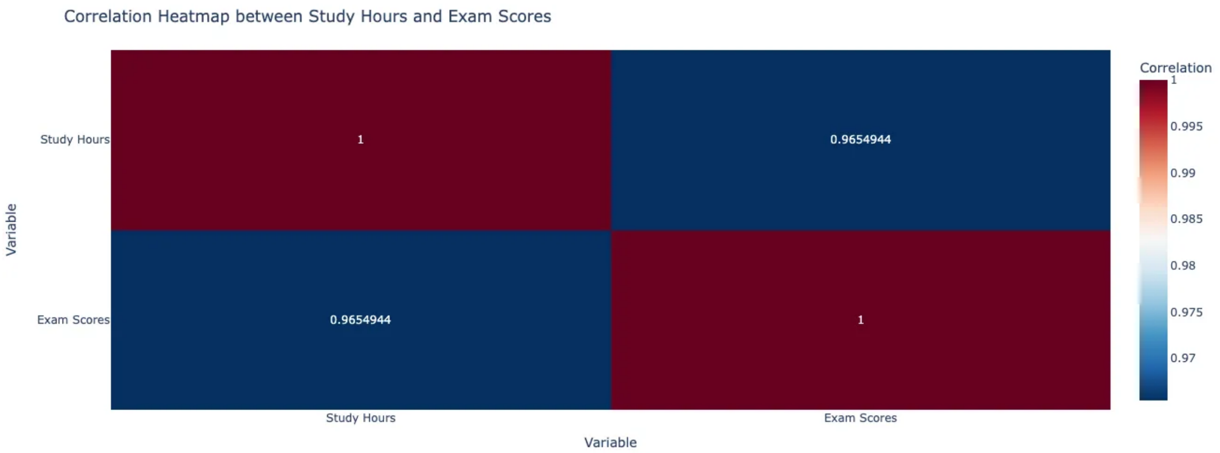
PCA uses the covariance or correlation matrix to understand the relationships between variables in the data.

Covariance Matrix

Measures the degree of linear association between variables.

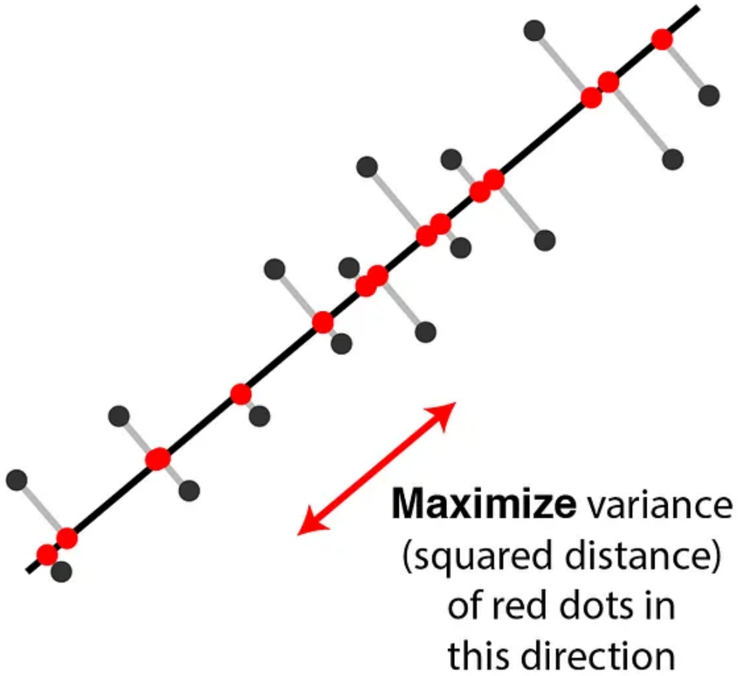
Correlation Matrix

Measures the strength and direction of linear relationships, normalized to a scale of -1 to 1.

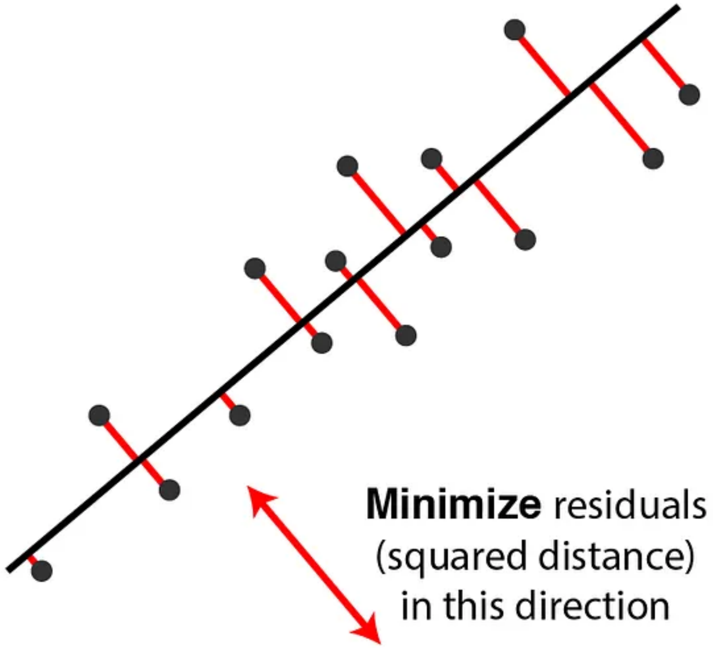


How does PCA work?

- The goal is to get the best principal components that capture maximum variance



Maximize variance (squared distance) of red dots in this direction



Minimize residuals (squared distance) in this direction

How does PCA work?

1 Standardization

The first step in PCA is to standardize the features. This involves scaling the features to have a mean of 0 and a standard deviation of 1. This is optional but recommended, as it can help to improve the performance of PCA.

2 Compute Covariance Matrix

The next step is to compute the covariance matrix. This matrix captures how the features vary together. The covariance matrix is a square matrix, where each element represents the covariance between two features.

3 Compute Eigenvectors and Eigenvalues

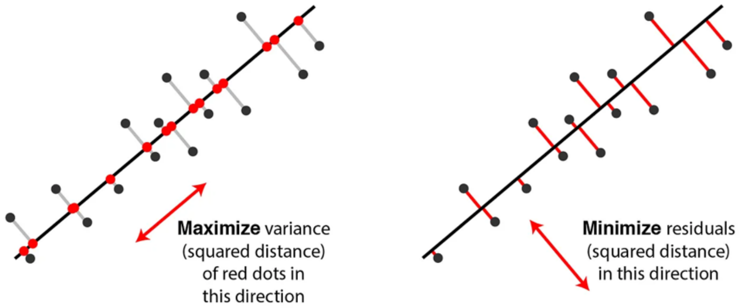
The next step is to compute the eigenvectors and eigenvalues of the covariance matrix. Eigenvectors represent the directions of greatest variance, and eigenvalues tell you how much variance each eigenvector explains.

4 Select Principal Components

The next step is to select the top _k_ eigenvectors based on their corresponding eigenvalues to form the new feature subspace. The number of principal components to select is a decision that needs to be made based on the specific application.

5 Projection

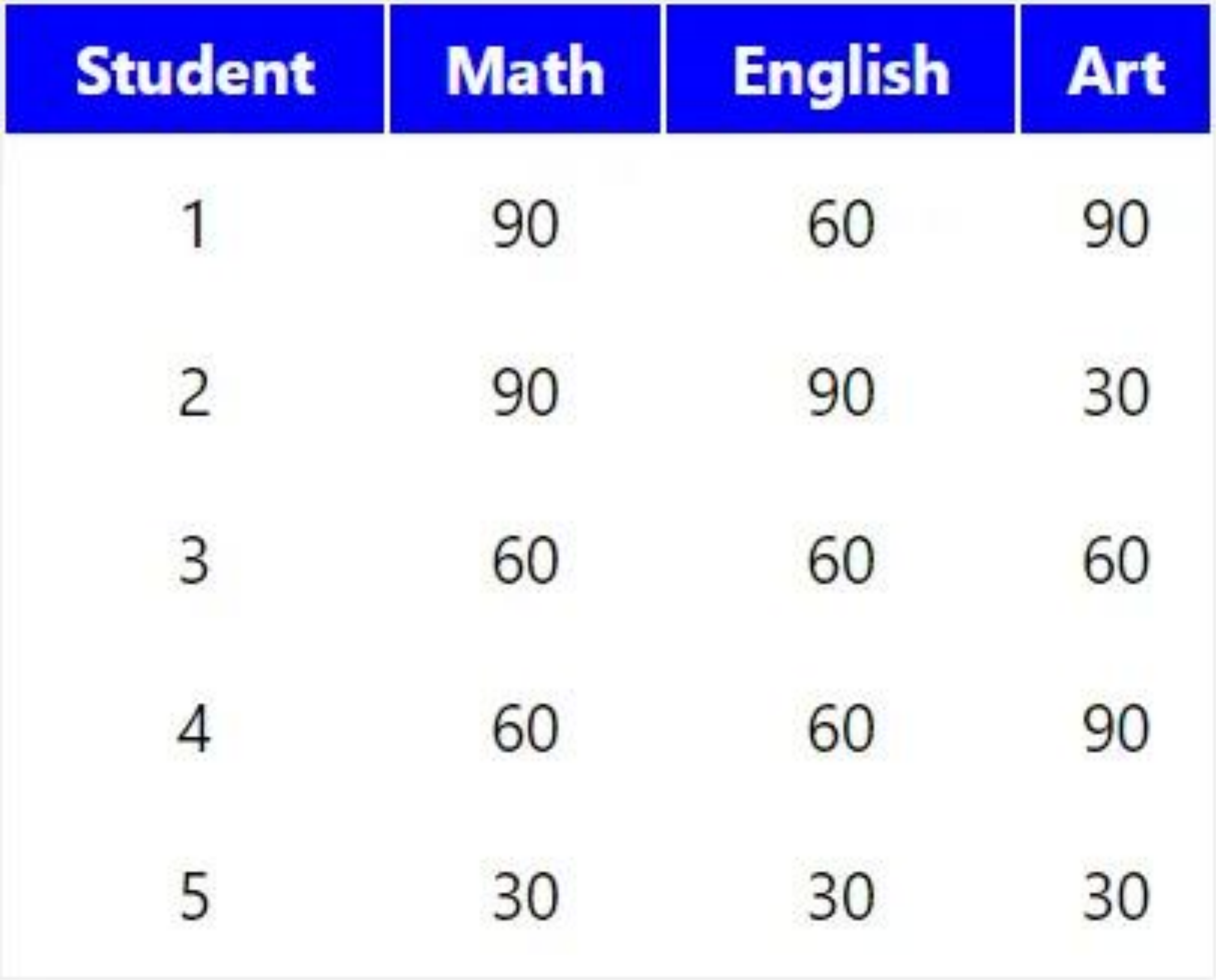
The final step is to project the original data onto the new feature subspace formed by the selected eigenvectors. This can be done by multiplying the original data matrix by the matrix of selected eigenvectors.



Maximize variance (squared distance) of red dots in this direction

Minimize residuals (squared distance) in this direction

Example: Students Score



Student	Math	English	Art
1	90	60	90
2	90	90	30
3	60	60	60
4	60	60	90
5	30	30	30

Step 1: Compute Covariance Matrix

- When we compute the covariance matrix for matrix A, our output is shown above
  - The diagonal elements represent the variance of scores for each test.
  - The off-diagonal elements represent the covariance between pairs of tests.

$$cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$$\begin{array}{cc} & \begin{array}{ccc} Math & English & Art \end{array} \\ \begin{array}{c} Math \\ English \\ Art \end{array} & \begin{bmatrix} 504 & 360 & 180 \\ 360 & 360 & 0 \\ 180 & 0 & 720 \end{bmatrix} \end{array}$$

Step 2: Compute Eigenvectors and Eigenvalues

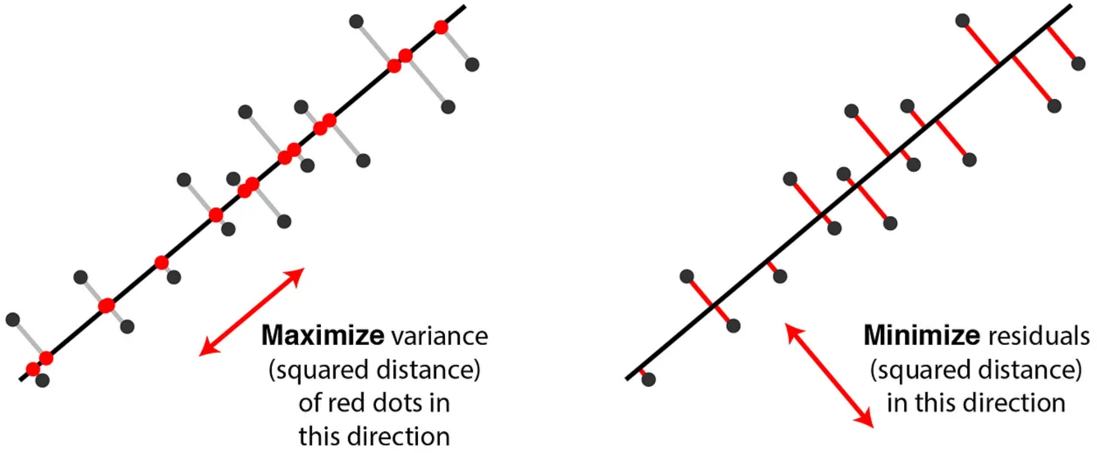
- Eigenvalues and eigenvectors represent the directions of maximum variance in the data

1 **Eigenvectors**

Direction of maximum variance in the data, representing the principal components.

2 **Eigenvalues**

Magnitude of the variance along each eigenvector.



Step 3: Sorting

$$\begin{pmatrix} 910.06995 \\ 629.11039 \\ 44.81966 \end{pmatrix}$$

1

Sorting Eigenvalues

The eigenvectors are sorted based on their corresponding eigenvalues in descending order. This means the eigenvector with the highest eigenvalue comes first, followed by the eigenvector with the second highest eigenvalue, and so on.

2

Choosing Top Eigenvectors

The top _k_ eigenvectors are chosen based on their corresponding eigenvalues. These eigenvectors form the new feature subspace. The number of eigenvectors chosen, _k_, determines the dimensionality of the new feature space.

3

Dropping Least Informative Eigenvectors

The lowest eigenvalues bear the least information about the distribution of the data. These eigenvalues are dropped, and their corresponding eigenvectors are not included in the new feature subspace.

Step 4: Projection onto New Subspace

Transform Samples

The last step involves transforming our samples onto the new subspace.

Matrix Multiplication

This transformation is achieved through matrix multiplication using the transpose of the matrix W.

$$w = \begin{bmatrix} 1.05594 & -0.50494 \\ 0.69108 & -0.67548 \\ 1 & 1 \end{bmatrix}$$

Equation

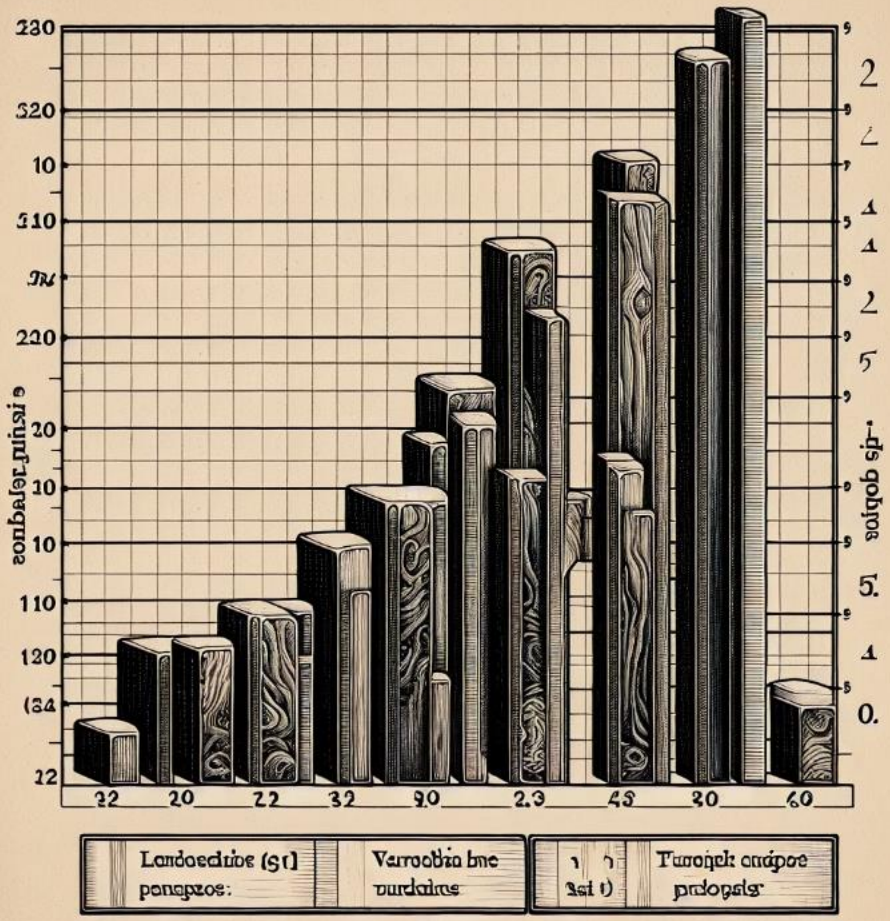
The equation for this transformation is y = W′ × x, where W′ is the transpose of W.

The matrix W, which we calculated in the previous step, is a 3x2 dimensional matrix. This matrix is used to project our samples onto the new subspace. The transpose of W, denoted as W′, is used in the equation y = W′ × x to perform this projection.

Interpreting Principal Components

The loadings of each original variable on the principal components help interpret the meaning of the reduced dimensions.

Variable	Principal Component 1	Principal Component 2
Feature 1	0.8	-0.2
Feature 2	-0.1	0.9
Feature 3	0.3	0.5



Pros of PCA

Reduced Complexity

PCA simplifies data by reducing the number of features. This makes it easier to visualize, manage, and interpret. It helps to focus on the most important information and eliminate noise.

Improved Algorithm Performance

Many machine learning algorithms struggle with high-dimensional data. PCA reduces training time and improves accuracy by reducing irrelevant features. This can lead to more efficient and effective models.

Reduced Redundancy

PCA eliminates redundant features, focusing on the unique information each piece of data provides. This is especially helpful when dealing with datasets with a lot of correlated features. It helps to avoid bias and improve the overall quality of the data.

Cons of PCA

Information loss

PCA discards information in the process of dimensionality reduction. There's a trade-off between the number of features retained and the amount of information preserved.

Assumes linearity

PCA works best when the relationships between features are linear. It may not be effective if the data has complex non-linear relationships.

Interpretability of new features

The principal components (PCs) created by PCA can be difficult to interpret in the context of the original features.