Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features.

.

| 500 Features | After Feature Selection → | Less than 500 Features |
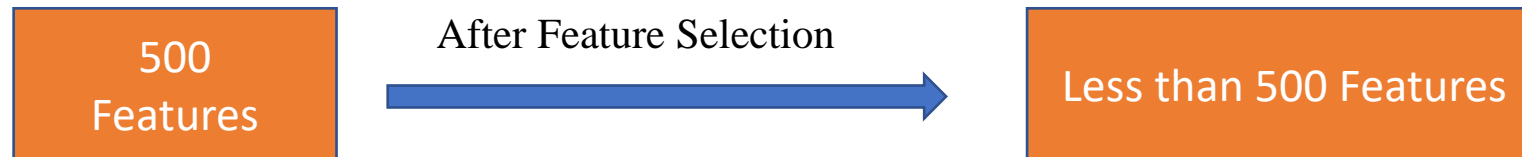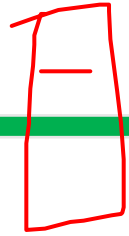
Dimensionality reduction is a technique used in machine learning to reduce the number of input features or variables while preserving the important information contained in the data. It is particularly useful when working with high-dimensional datasets where the number of features is large. The goal of dimensionality reduction is to simplify the data representation by transforming it into a lower-dimensional space. This can lead to several benefits, such as:

1. Reduced computational complexity: High-dimensional data can be computationally expensive to process and analyze. Dimensionality reduction helps in reducing the computational requirements by working with a smaller set of features.

2. Elimination of irrelevant features: Not all features in a dataset contribute equally to the overall information. Removing some features may be redundant or irrelevant, which can improve model performance by focusing on the most important information.

3. Visualization: Visualizing high-dimensional data is challenging. Dimensionality reduction techniques can project the data onto a lower-dimensional space, typically 2D or 3D, allowing easier visualization and understanding of the underlying patterns.
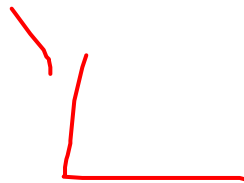
Feature selection techniques can be classified into three main categories:

1. Filter Methods: Filter methods rank features based on statistical measures or heuristic measures that assess the relevance of each feature independently of the machine learning model. Common statistical measures include correlation coefficients, chi-square test, information gain, and mutual information. Filter methods are computationally efficient and can be applied as a preprocessing step before training the model.

2. Wrapper Methods: Wrapper methods evaluate feature subsets by training and evaluating a machine learning model on different subsets of features. They use a specific machine learning algorithm to assess the performance of different feature subsets. Wrapper methods are computationally more expensive compared to filter methods, but they can capture feature interactions and provide better feature subsets tailored to the specific learning algorithm.

3. Embedded Methods: Embedded methods incorporate the feature selection process as part of the model training process. These methods typically use regularization techniques, such as L1 regularization (Lasso) or L2 regularization (Ridge), which impose penalties on the model coefficients. The regularization process encourages the model to automatically select the most relevant features during training.

Transpose Matrix:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}_{2 \times 3} \qquad A^T = \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}_{3 \times 2}$$

Is the column vector a linear combination of anything that comes before it?

Nothing comes before it.

No scalar multiplication of [1,1] that can reach [1,2].

1([1,2,1]) + 2(1,1,3) can reach [3,4,7]].

2([1,1,3]) can reach [2,2,6].

✗ ✗ ✓ ✓

$$\begin{bmatrix} 1 & 1 & 3 & 2 \\ 2 & 1 & 4 & 2 \\ 1 & 3 & 7 & 6 \end{bmatrix} = rank\ 2$$

Addition:

3+4=7

$$\begin{bmatrix} 3 & 8 \\ 4 & 6 \end{bmatrix} + \begin{bmatrix} 4 & 0 \\ 1 & -9 \end{bmatrix} = \begin{bmatrix} 7 & 8 \\ 5 & -3 \end{bmatrix}$$

Multiplication:

$$\begin{bmatrix} A & B \\ C & D \\ E & F \end{bmatrix} \times \begin{bmatrix} G \\ H \end{bmatrix} = \begin{bmatrix} A \times G + B \times H \\ C \times G + D \times H \\ E \times G + F \times H \end{bmatrix}$$

The homogenous system of linear equations is :

$$a_{11} x_1 + a_{12} x_2 + \ldots\ldots + a_{1n} x_n = 0$$
$$a_{21} x_1 + a_{22} x_2 + \ldots\ldots + a_{2n} x_n = 0$$

Khan academy: https://www.youtube.com/watch?v=xyAuNHPsq-g&list=PLFD0EB975BA0CC1E0

The (linear) PCA algorithm

1. Compute mean value of data $\overline{X} = \frac{1}{N}\sum_{i=1}^{N} x^{(i)}$

2. Center data via $y^{(i)} = x^{(i)} - \overline{X}$

3. Compute covariance matrix $C = \frac{1}{N}\sum_{i=1}^{N} y^{(i)} y^{(i)^T}$

4. Determine the M eigenvalues and eigenvectors of C numerically

5. Select $1 \leq k \leq M$ respective eigenvectors $v^{(1)}, \ldots, v^{(k)}$ of the k largest non-vanishing eigenvalues

6. Assemble selected eigenvectors $v^{(1)}, \ldots, v^{(k)}$ columnwise to matrix $T \in \mathbb{R}^{M \times k}$

7. Compute principal components for each centred input point $y^{(i)} \in \mathbb{R}^{M}$ via:
$$T^T y^{(i)} = z^{(i)} \in \mathbb{R}^{k}$$

Question

Let input data $x^1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$, $x^2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$, $x^3 = \begin{pmatrix} 1 \\ -2 \end{pmatrix}$, $x^4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ be given. Compute for all data points the first principal component (i.e. dimension $k = 1$).

Solution: PCA

Step:01 Compute mean value

$$\overline{X} = \frac{1}{4} \sum_{i=1}^{4} x^i = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Step:02 Center data $y^i = x^i - \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

Step:03 Compute covariance matrix

$$C = \frac{1}{4} \sum_{i=1}^{4} y^i (y^i)^T = \frac{1}{4} \left(\begin{pmatrix} 4 & 4 \\ 4 & 4 \end{pmatrix} + \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 4 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \right) = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

(Alternative: $Y = (y^1\ y^2\ y^3\ y^4)$, $C = \frac{1}{4} Y Y^T$)

Step:04 Compute eigenvalues and eigenvectors. First, compute the roots of the characteristic polynomial of C:

$$\chi_C(\lambda) = (\lambda - 2)(\lambda - 2) - 1 = \lambda^2 - 4\lambda + 3.$$

Using the quadratic formula we can calculate the eigenvalues $\lambda = 3$ and $\lambda = 1$. Therefore, the largest eigenvalue is $\lambda = 3$ and since we want to compute one principal component it is enough to calculate the eigenvector for $\lambda = 3$.

Step:05 This can be done with Gaussian elimination:

$$C - 3 \cdot \mathbb{1} = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \Leftrightarrow \begin{pmatrix} -1 & 1 \\ 0 & 0 \end{pmatrix}.$$

This implies that the eigenvector is given by $v = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Step:06 $T = (v) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Step:07 Compute the first principal components for each data point:

$$z^1 = T^T y^1 = (1\ \ 1) \begin{pmatrix} 2 \\ 2 \end{pmatrix} = 4,$$

$$z^2 = T^T y^2 = (1\ \ 1) \begin{pmatrix} -2 \\ 0 \end{pmatrix} = -2,$$

$$z^1 = T^T y^1 = (1\ \ 1) \begin{pmatrix} 0 \\ -2 \end{pmatrix} = -2,$$

$$z^1 = T^T y^1 = (1\ \ 1) \begin{pmatrix} 0 \\ 0 \end{pmatrix} = 0.$$

(Solved)

$$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = 0 \Rightarrow -\begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$$

$$\begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = 0$$

$$(2-\lambda)(2-\lambda) - 1 = 0$$
$$= 4 - 2\lambda - 2\lambda + \lambda^2 - 1 = 0$$
$$\lambda^2 - 4\lambda + 3 = 0$$
$$\lambda^2 - 3\lambda - \lambda + 3 = 0$$
$$\lambda(\lambda - 3) - 1(\lambda - 3) = 0$$
$$(\lambda - 3)(\lambda - 1) = 0$$
$$\lambda = 3, \lambda = 1$$

$$x + y - z = -2$$
$$2x - y + z = 5$$
$$-x + 2y + 2z = 1$$

$$
\begin{array}{ccc|c}
x & y & z & \\
\hline
1 & 1 & -1 & -2 \\
2 & -1 & 1 & 5 \\
-1 & 2 & 2 & 1
\end{array}
$$

$$\rightarrow R_3 = 2R_1 + R_3$$

$$
\begin{array}{ccc|c}
1 & 1 & -1 & -2 \\
2 & -1 & 1 & 5 \\
0 & 3 & 1 & -1
\end{array}
$$

$$-2R_1 + R_2$$

$$
\begin{array}{ccc|c}
1 & 1 & -1 & -2 \\
0 & -3 & 3 & 9 \\
0 & 3 & 1 & -1
\end{array}
$$

$$R_2 + R_3$$

$$
\begin{array}{ccc|c}
1 & 1 & -1 & -2 \\
0 & -3 & 3 & 9 \\
0 & 0 & 4 & 8
\end{array}
$$

$$\Rightarrow$$

$$x + y - z = -2$$
$$-3y + 3z = 9$$
$$4z = 8$$

$$\curvearrowright z = 2$$

$$-3y + 6 = 9$$
$$y = -1$$

$$x = -2 - 2$$
$$x = 1$$