Machine Learning

Video 46:

Curse of Dimensionality

The Curse of Dimensionality refers to the challenges faced when working with high-dimensional data in machine learning. As the number of features increases, data becomes sparse, making it harder to find meaningful patterns. This leads to overfitting, increased computational cost, and reduced model performance in high-dimensional spaces.

Thus, it says that adding columns (or called as features or dimensionality) will be helpful only to a certain point, after that there is no use of it.

Why high dimensionality is not good?

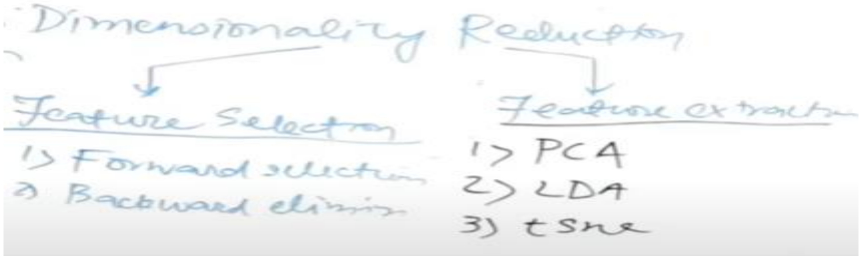
High dimensionality can be problematic because:

1. **Sparsity**: Data becomes sparse, meaning there are fewer data points relative to the number of dimensions. This makes it harder to generalize and find patterns.
2. **Overfitting**: Models may memorize data instead of learning general trends, leading to poor performance on unseen data.
3. **Increased computational cost**: More features require more calculations, slowing down model training and making it harder to interpret results.
4. **Distance metrics lose meaning**: In high dimensions, the distance between data points becomes less informative, affecting algorithms like k-nearest neighbours.

How to reduce this effect?

To mitigate the Curse of Dimensionality, several techniques can be applied:

1. **Dimensionality Reduction**: Methods like PCA (Principal Component Analysis) or t-SNE reduce the number of features while preserving important patterns, making the data more manageable.
2. **Feature Selection**: Selecting only the most relevant features through techniques like recursive feature elimination (RFE) or feature importance (from models like Random Forests) helps reduce dimensionality.
3. **Regularization**: Regularization methods such as L1 (Lasso) or L2 (Ridge) penalize overly complex models, preventing overfitting in high-dimensional spaces.
4. **Increasing Data**: Collecting more data can help alleviate sparsity by providing a more complete representation of the high-dimensional space.
5. **Nonlinear Models**: Some models, like neural networks, can naturally handle high-dimensional data by learning complex representations and reducing the impact of irrelevant features.



Video 47:

Principle Component Analysis (PCA)

We will cover the point Feature extraction.

What is PCA?

Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms data into a new coordinate system. It identifies the directions (principal components) with the most variance in the data and projects the data onto these components, reducing the number of features while retaining important information.

It is unsupervised.

Its main goal is to reduce the features, while keeping the essence of the data.

Benefits:

The benefits of PCA include:

1. **Dimensionality Reduction**: Reduces the number of features, making models faster and less complex.
2. **Noise Reduction**: Helps eliminate irrelevant or noisy features, improving model performance.
3. **Improved Visualization**: Reduces high-dimensional data to 2D or 3D, making it easier to visualize patterns.
4. **Enhances Performance**: Increases model accuracy by focusing on the most important features.
5. **Uncorrelated Features**: PCA produces new features that are uncorrelated, which can help certain algorithms perform better.

What is feature selection?

Feature selection is the process of selecting a subset of the most relevant features (variables) from the original dataset to improve model performance. It involves removing irrelevant, redundant, or noisy features, which can reduce overfitting, improve model accuracy, and decrease computational costs. Methods include filtering, wrapping, and embedding techniques.

What is Feature extraction?

Feature extraction is the process of transforming raw data into a set of meaningful, compact features that capture the essential information for machine learning models. Unlike feature selection, which selects from existing features, feature extraction creates new features by combining or transforming the original data, often using techniques like PCA or domain-specific methods.

Use Feature Selection when:

1. **You have a large number of features**: Reducing the feature set can improve model performance and reduce computational complexity.
2. **Your features are already meaningful**: If the original features are already informative, selecting the most relevant ones can enhance accuracy.
3. **You want to simplify the model**: It helps in creating more interpretable models by keeping only the most important features.

Use Feature Extraction when:

1. **The data is high-dimensional**: When there are too many features that don't directly represent the underlying patterns, feature extraction can help create a compact, meaningful representation (e.g., using PCA or deep learning).
2. **Features are highly correlated**: When features are redundant, extraction can combine them into new, uncorrelated features, capturing more variance in fewer dimensions.
3. **You need to transform data for specific models**: In cases where models require transformed data (like text or image data), feature extraction can help convert raw data into more useful features.

In summary, feature selection is used when you want to choose the most relevant features from existing ones, while feature extraction is used when you need to create new features that better represent the data.

How does this PCA works?

PCA (Principal Component Analysis) works by transforming the data into a new set of axes, where the first axis (the first principal component) captures the most variance, the second axis captures the second most variance, and so on.

Here's how PCA works in terms of a graph:

1. **Original Data**: Imagine a 2D scatter plot where each point represents a sample with two features (x and y axes). The points are spread out in different directions.
2. **Find the Directions of Maximum Variance**: PCA identifies the directions in which the data varies the most. These are the principal components. The first principal component (PC1) is the direction that has the maximum variance (the longest spread), and the second principal component (PC2) is orthogonal (perpendicular) to the first, capturing the next highest variance.
3. **Projection onto New Axes**: The data is projected onto the new axes (PC1 and PC2). This transforms the original data into a new coordinate system, where the points are aligned along the principal components.
4. **Dimensionality Reduction**: By keeping only the first few principal components (which capture the most variance), we can reduce the dimensionality of the data while preserving most of its information.

In terms of a 2D graph:

- **Step 1**: Imagine the scatter of data points along the x and y axes.
- **Step 2**: PCA finds the "line" or axis (principal component) that best fits the data, which is the direction of greatest variance.
- **Step 3**: Data points are projected onto this new line (PC1), and if needed, the second line (PC2) captures the remaining variance.

Why Variance in important?

Variance is important in PCA and other machine learning techniques because it measures the spread or dispersion of data points. Here's why it matters:

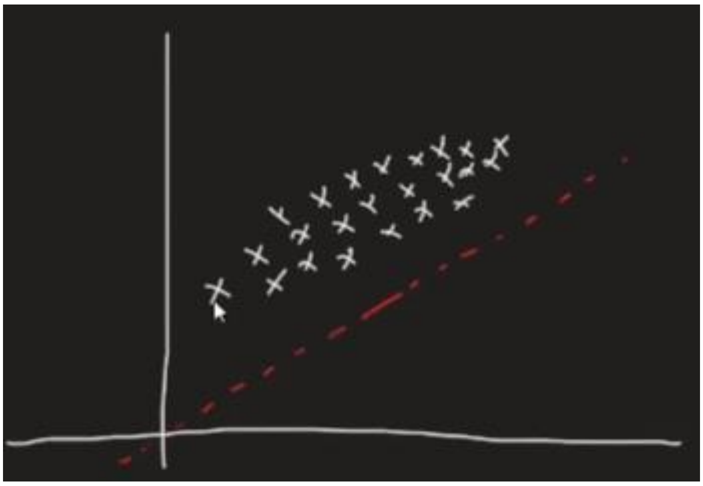
1. **Capturing Information**: The more variance a feature has, the more information it contains about the dataset. By focusing on the directions of maximum variance, PCA helps to preserve the most significant aspects of the data when reducing dimensionality.

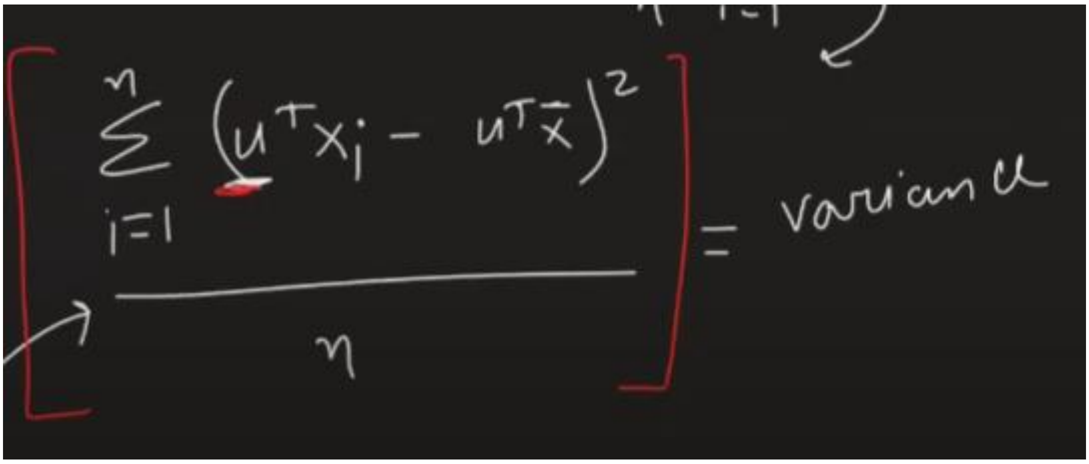
2. **Dimensionality Reduction**: In PCA, we prioritize components that capture the highest variance. The assumption is that these components represent the most important patterns, and less-variant components are often less informative and can be discarded, reducing data complexity.
3. **Data Representation**: Variance indicates how much a feature or set of features contributes to the diversity of the data. Larger variance means more diversity, which can lead to more meaningful insights and better model predictions.
4. **Noise vs. Signal**: Variance helps separate noise (low variance) from signal (high variance). Features with high variance likely represent meaningful patterns, while low-variance features might be just noise, which can negatively affect the performance of a model.

In summary, variance plays a key role in understanding the structure of data, guiding which aspects to focus on for better performance and more effective data analysis.

Video 48:

Principle Component Analysis (PCA) | Problem Formulation and Step by Step Solution





$$\left[\frac{\sum_{i=1}^{n}\left(u^T x_i - u^T \bar{x}\right)^2}{n}\right] = \text{variance}$$

#Steps

#1. Mean centring (not mandatory)

#2. Find covariance matrix

#3. Find Eigen value and eigen vector

Example:

Code link:

https://colab.research.google.com/drive/1feultBseSRfhEpApk3CmVnd33wjIlcft?usp=sharing

Video 49:

Principle Component Analysis (PCA) – code explained

Example:
Code link:

https://www.kaggle.com/code/rememberful/notebook121f3a79d6

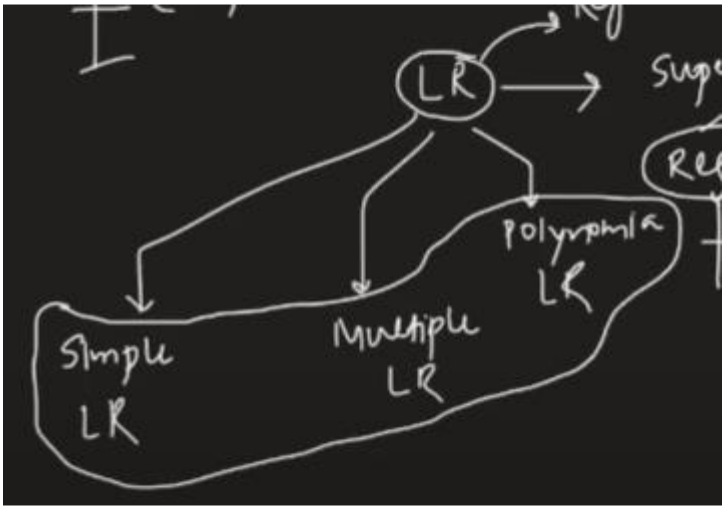
Video 50:

Simple Linear Regression

Simple Linear Regression is a statistical method used in machine learning to model the relationship between two variables by fitting a linear equation (y = mx + b) to the data. It predicts a dependent variable (y) based on an independent variable (x), aiming to minimize prediction errors.

It is supervised.

Types:



In Simple Linear Regression, there is only **one input column** (or independent variable). It models the relationship between a single predictor (input) and the target (output) variable. If there are multiple input columns, it becomes **Multiple Linear Regression** instead.

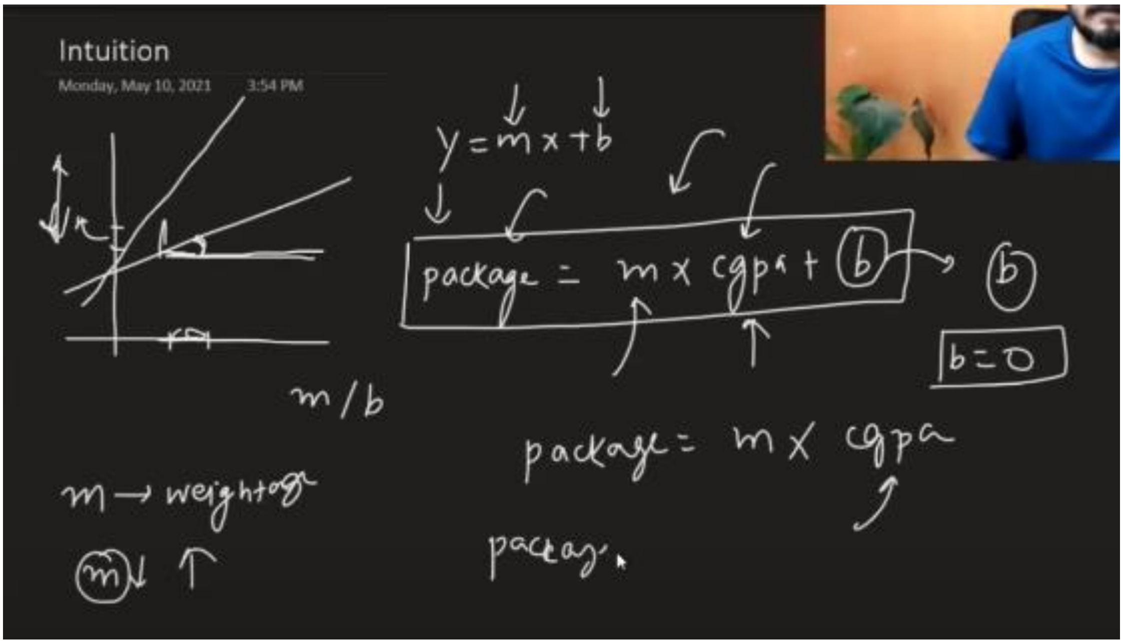
Example:

Code link:

https://colab.research.google.com/drive/1dwfdtMale-0wb8ygBOJftm-ZbiVRyQKo?usp=sharing

Data link:

https://github.com/campusx-official/100-days-of-machine-learning/tree/main/day48-simple-linear-regression



Intuition

Monday, May 10, 2021 3:54 PM