Introduction to Linear transformation and application in Data science

Functions :

A function is a mathematical relationship that uniquely associates element of one set (called domain) with element of the of another set (called codomain) in simpler terms the function maps input to output in a specific way.

Notation: The function f mapping elements from set X (domain) to set Y (co-domain) is denoted by f : X->Y

Linear Transformation:

A linear transformation is a function between two vector spaces that preserves the operation of vector addition and scalar multiplication. This mean that if T is a linear transformation from a vector V to vector space W , then for any vectors satisfies 2 conditions

 T : V->W

Additivity T(u+v)=T(u)+T(v)

Homogeneity T(cu)=c T(u)

Where u and v ∈ Vectors and c is the scalar value

Application of the Linear transformation:

1. Principal Component Analysis (PCA)

Definition:

Principal Component Analysis (PCA) is a dimensionality reduction method that projects a high-dimensional dataset onto a lower-dimensional space while retaining the most significant patterns in the data. It achieves this by identifying new orthogonal axes (principal components) that explain the maximum variance in the dataset.

Function: $f(X)=(W^T)X$ ---> T is the transpose

Example:

Take an example of a dataset of student scores in Math, Science, and English. These three variables are correlated, i.e., we may not require all three to determine student performance. PCA has the capability to convert them into a single principal component (PC1) that captures overall performance without losing much of the original data.

How Linear Transformation is Applied:

PCA uses a linear transformation by computing the eigenvectors and eigenvalues of the covariance matrix of the data. This feature space rotation aligns the new principal components along the directions of largest variance, efficiently dimensionality-reducing the data while preserving vital information.

2. Feature Engineering

Definition:

Feature engineering is the action of constructing new features from raw data to enhance model performance. This may include scaling, merging, or converting current features into more informative representations.

Function: $f(Weight, Height) = (Weight)/(Height) * (Height)$

Example:

In the dataset with weight (kg) and height (cm), instead of using the raw values, we can construct a new feature:

$BMI = (Weight)/(Height) * (Height)$

Weight in kg and Height in meter

This new feature depicts body composition more accurately than height and weight separately.

How Linear Transformation is Used:

Linear transformations are used by taking mathematical operations such as addition, multiplication, or linear combinations in order to generate new features. For example, if we aim to generate an interaction term of two features, we multiply those two features with each other and obtain a new feature that shows their combined effect on the target variable.

3. Data Preprocessing (Normalization & Standardization)

Definition:

Data preprocessing involves scaling, transforming, and preparing data for machine learning models. Two common techniques—Normalization and Standardization—help ensure that different features contribute equally to model training.

(A) Normalization (Min-Max Scaling)

This method normalizes data into a constant range, typically [0,1]:

Function : $f(X) = X - Xmin/Xmax - Xmin$

Use Case: Ideal for models that are sensitive to the range of input features, like k-NN, neural networks, and clustering algorithms.

Example: If income ranges from $10,000 to $100,000, normalizing it will keep all values between 0 and 1.

(B) Standardization (Z-Score Normalization)

This method transforms data to have zero mean and unit variance:

Function: $f(X) = X - \mu/\sigma$

Use Case: Critical for models based on Gaussian (normal) distribution, e.g., linear regression, logistic regression, PCA.

For a mean salary of $60,000 and a standard deviation of $15,000, a salary of $75,000 standardizes to 1.0, meaning it is 1 standard deviation above the mean.

How Linear Transformation is Used:

Linear transformations are used in normalization and standardization through scaling and shifting the data. In normalization, we use a linear transformation to transform the values to a particular range. In standardization, we divide by the standard deviation and subtract the mean to transform the data to a standard normal distribution.

4. Neural Networks (Linear Layers)

Definition:

Neural networks are computational models based on the human brain. They are composed of layers of neurons, where each neuron performs a transformation on input data. Example: A basic fully connected layer in a neural network performs the transformation:

Function $f(X)=WX+b$

Where,

X = Input features (e.g., image pixels, numerical data)

W = Weight matrix (learned during training)

b = Bias term (adjusts output)

Z= Transformed output

How Linear Transformation is Applied:

In neural networks, linear transformations are applied by performing matrix multiplication between the weight matrix W and the input features X. This operation produces a weighted sum of the inputs, and adding the bias term b shifts the output, allowing the model to learn complex relationships in the data.

Inverse Transformation

Inverse transformations are essential in data processing, especially when we need to convert standardized or normalized data back to its original scale. This step is crucial when interpreting the results of machine learning models, as it allows us to understand predictions in the context of the original data.

Example: Inverse of Standardization

Consider a dataset with a mean salary of $60,000 and a standard deviation of $15,000. After standardizing a salary of $75,000, we obtain a standardized score of 1.0. To retrieve the original salary from the standardized score, we apply the inverse transformation:

$$\text{Original Salary}=(\text{Standardized Score}\times\sigma)+\mu=(1.0\times15{,}000)+60{,}000=75{,}000$$

This inverse transformation confirms that the standardized score of 1.0 corresponds to the original salary of $75,000.

Conclusion:

In summary, linear transformations are central to many applications in data science and machine learning. From Dimensionality Reduction using Principal Component Analysis (PCA) to feature engineering that maximizes model performance, linear transformations enable the efficient representation of data. Normalization and standardization techniques see to it that input features are properly scaled so as to make balanced contributions to training the model. In addition, neural networks use linear transformations in their layers to learn intricate relationships in data.

The role of inverse transformations is crucial, as they enable us to reverse normalized or standardized data back to the original scale so that model predictions can be meaningfully and interpretably understood. Having an understanding of these concepts gives data scientists the tools required to preprocess data well, to optimize model performance, and to interpret results correctly, ultimately informing better decision-making.

As the world of data science keeps advancing, becoming a master of linear transformations and their uses will always be fundamental in gaining meaningful insights from large sets of data.