19th Mar_AssQ on Feature Engineering -3

July 12, 2023

Q1. What is Min-Max scaling, and how is it used in data preprocessing? Provide an example to illustrate its application.

Min-Max scaling, also known as normalization, is a common data preprocessing technique used to rescale numeric features to a specific range. It transforms the data so that it falls within a predefined interval, typically between 0 and 1. This scaling method is useful when the features have different scales and ranges, and we want to bring them to a common scale to avoid dominance of certain features over others.

The formula to perform Min-Max scaling on a feature x is as follows:

x_scaled = (x - min(x)) / (max(x) - min(x))

In this formula, min(x) represents the minimum value of the feature x, and max(x) represents the maximum value of the feature x.

Here's an example to illustrate the application of Min-Max scaling:

Let's say we have a dataset of housing prices with two features: "Area" (measured in square feet) and "Price" (measured in dollars). The "Area" feature ranges from 500 to 3000 square feet, while the "Price" feature ranges from $50,000 to $500,000.

Before applying Min-Max scaling, the data might look like this:

Area | Price 1500 | 200,000 1000 | 150,000 2500 | 300,000

To scale the features using Min-Max scaling, we calculate the minimum and maximum values for each feature:

min(Area) = 500 max(Area) = 3000

min(Price) = 50,000 max(Price) = 500,000

Next, we apply the Min-Max scaling formula to each value:

Scaled_Area = (Area - min(Area)) / (max(Area) - min(Area)) Scaled_Price = (Price - min(Price)) / (max(Price) - min(Price))

After performing the calculations, the scaled data will look like this:

Scaled_Area | Scaled_Price 0.333 | 0.3 0.000 | 0.0 1.000 | 1.0

As you can see, the "Area" values are now scaled between 0 and 1, and the "Price" values are also scaled between 0 and 1. This scaling ensures that both features have the same impact on any subsequent analysis or machine learning models that use the data

Q2. What is the Unit Vector technique in feature scaling, and how does it differ from Min-Max scaling? Provide an example to illustrate its application.

The Unit Vector technique, also known as vector normalization, is another data preprocessing method used for feature scaling. Unlike Min-Max scaling, which scales the features to a specific range, the Unit Vector technique scales the features to have a unit norm or length.

In the Unit Vector technique, each feature vector is divided by its Euclidean norm, resulting in a vector with a length of 1. This normalization ensures that all the feature vectors have the same scale, but it does not preserve the original range of the data.

The formula to perform Unit Vector scaling on a feature vector x is as follows:

x_scaled = x / ||x||

In this formula, ||x|| represents the Euclidean norm or length of the feature vector x.

Here's an example to illustrate the application of the Unit Vector technique:

Let's consider a dataset of customer reviews, where each review is represented by a feature vector indicating the frequency of occurrence of certain words. Suppose we have two features: "love" and "hate," representing the number of times the words "love" and "hate" appear in each review.

Before applying the Unit Vector technique, the data might look like this:

love | hate 2 | 1 3 | 4 1 | 2

To scale the features using the Unit Vector technique, we calculate the Euclidean norm for each feature vector:

||[2, 1]|| = sqrt(2^2 + 1^2) = sqrt(5)   2.236 ||[3, 4]|| = sqrt(3^2 + 4^2) = sqrt(25) = 5 ||[1, 2]|| = sqrt(1^2 + 2^2) = sqrt(5)   2.236

Next, we divide each feature vector by its Euclidean norm:

Scaled_love = [2, 1] / 2.236   [0.894, 0.447] Scaled_hate = [1, 2] / 2.236   [0.447, 0.894]

After performing the calculations, the scaled data will look like this:

Scaled_love | Scaled_hate 0.894 | 0.447 0.6 | 0.8 0.447 | 0.894

Q3. What is PCA (Principle Component Analysis), and how is it used in dimensionality reduction? Provide an example to illustrate its application.

PCA, which stands for Principal Component Analysis, is a dimensionality reduction technique used to transform a high-dimensional dataset into a lower-dimensional space while preserving most of the important information. It achieves this by identifying the principal components, which are linear combinations of the original variables, capturing the maximum amount of variance in the data.

The process of PCA involves the following steps:

Standardize the data: If the features of the dataset have different scales, it is necessary to standardize them so that each feature has a mean of zero and a standard deviation of one. This step ensures that all variables contribute equally to the analysis.

Compute the covariance matrix: The covariance matrix is calculated based on the standardized data, representing the relationships and variances between the different variables.

Compute the eigenvectors and eigenvalues: The eigenvectors and eigenvalues are derived from the covariance matrix. The eigenvectors determine the directions of the principal components, and the eigenvalues represent the amount of variance explained by each component.

Select the principal components: The eigenvectors are ranked according to their corresponding eigenvalues, with the highest eigenvalue indicating the most important component. By selecting a subset of these components, we can reduce the dimensionality of the data.

Q4. What is the relationship between PCA and Feature Extraction, and how can PCA be used for Feature Extraction? Provide an example to illustrate this concept.

PCA and feature extraction are closely related concepts, with PCA being a specific technique used for feature extraction. Feature extraction refers to the process of transforming the original set of features into a new set of features, typically with lower dimensionality, while preserving or maximizing the relevant information in the data.

PCA can be used as a feature extraction method by identifying the principal components, which are linear combinations of the original features. These principal components are ordered based on their corresponding eigenvalues, indicating the amount of variance explained by each component. By selecting a subset of the principal components, we can effectively reduce the dimensionality of the data while retaining the most important information.

Here's an example to illustrate the concept of using PCA for feature extraction:

Suppose we have a dataset with 100 images, each represented by a vector of 1,000 pixels. Each pixel represents a feature in the dataset, resulting in a high-dimensional input space. We want to extract a smaller set of features that captures the most important information in the images.

We can apply PCA to this dataset, where each image is considered as a data point. By performing PCA, we obtain the principal components, which are essentially patterns or directions in the pixel space that explain the maximum variance.

Let's say after performing PCA, we find that the first principal component is strongly related to the overall brightness of the images, while the second principal component captures the variation in the orientation of edges. These principal components can be seen as new features that summarize the information contained in the original pixels.

We can choose to retain only a subset of the principal components, such as the first 50 components, which would result in a lower-dimensional representation of the images. Each image is now represented by a vector of 50 values instead of the original 1,000 pixels.

This reduced-dimensional feature representation can be used for various tasks such as image classification, clustering, or visualization. By extracting the most relevant features using PCA, we can simplify the data, remove noise or redundancy, and improve computational efficiency while still retaining the essential information needed

Q5. You are working on a project to build a recommendation system for a food delivery service. The dataset contains features such as price, rating, and delivery time. Explain how you would use Min-Max scaling to preprocess the data.

To preprocess the data for building a recommendation system for a food delivery service, you can use Min-Max scaling on certain features such as price, rating, and delivery time. Min-Max scaling, also known as normalization, is a technique used to transform the data to a specific range, typically between 0 and 1.

Here's how you would use Min-Max scaling to preprocess the data:

Identify the features: In this case, the features are price, rating, and delivery time.

Determine the minimum and maximum values: Calculate the minimum and maximum values for each feature in the dataset. For example, find the minimum and maximum prices, ratings, and delivery times in the dataset.

Apply the Min-Max scaling formula: For each feature, use the following formula to scale the values:

scaled_value = (original_value - min_value) / (max_value - min_value)

This formula scales the original value to a range between 0 and 1 based on the minimum and maximum values of that feature.

Q6. You are working on a project to build a model to predict stock prices. The dataset contains many features, such as company financial data and market trends. Explain how you would use PCA to reduce the dimensionality of the dataset.

When building a model to predict stock prices with a dataset containing numerous features like company financial data and market trends, PCA can be used to reduce the dimensionality of the dataset. By reducing the number of features, PCA can help simplify the model and improve its computational efficiency, while still retaining most of the important information.

Here's how you can use PCA for dimensionality reduction in the context of predicting stock prices:

Data preprocessing: Start by preprocessing the dataset, which may involve cleaning the data, handling missing values, and normalizing the features to ensure they are on a similar scale. This step is crucial before applying PCA.

Select the features: Identify the relevant features from the dataset that you want to include in the dimensionality reduction process. These features should be related to company financial data and market trends, such as stock price history, earnings per share, market indices, trading volumes, or other financial indicators.

Apply PCA: Once you have selected the relevant features, apply PCA to the dataset. PCA will transform the original features into a set of linearly uncorrelated components called principal components.

Q7. For a dataset containing the following values: [1, 5, 10, 15, 20], perform Min-Max scaling to transform the values to a range of -1 to 1.

[18]:
```
import pandas as p
from sklearn.preprocessing import MinMaxScaler
data = [1,5,10,15,20]

scaler = MinMaxScaler(feature_range=(-1,1))
reshaped_data = [[x] for x in data]
scaled_data = scaler.fit_transform(reshaped_data)
scaled_data = [x[0] for x in scaled_data]
```

[19]:
```
scaled_data
```

[19]:
```
[-0.9999999999999999,
 -0.5789473684210525,
 -0.05263157894736836,
 0.47368421052631593,
 1.0]
```

Q8. For a dataset containing the following features: [height, weight, age, gender, blood pressure], perform Feature Extraction using PCA. How many principal components would you choose to retain, and why?

[20]:
```python
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

data = {
    'height': [170, 165, 180, 160],
    'weight': [65, 60, 75, 55],
    'age': [30, 35, 45, 28],
    'gender': [1, 0, 1, 0],
    'blood pressure': [120, 110, 130, 115]
}

# Convert the data dictionary to a numpy array
X = np.array(list(data.values())).T

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Apply PCA
pca = PCA()
pca.fit(X_scaled)

# Determine the number of principal components to retain
explained_variance_ratio = pca.explained_variance_ratio_
cumulative_variance_ratio = np.cumsum(explained_variance_ratio)

# Find the number of components that explain at least 95% of the variance
num_components = np.argmax(cumulative_variance_ratio >= 0.95) + 1

print("Number of principal components to retain:", num_components)
```

```
Number of principal components to retain: 2
```

[ ]:

[ ]: