General Data Science Terminologies

Common Terms and Terminologies that are used in Data Science

Contents

Data Science: 4
Machine Learning (ML): 4
Supervised Learning: 4
Unsupervised Learning: 4
Regression: 4
Classification: 4
Clustering: 4
Dimensionality Reduction: 4
Feature Engineering: 4
Train/Test Split: 4
Cross-Validation: 4
Overfitting: 5
Underfitting: 5
Python: 5
List Comprehension: 5
Lambda Function: 5
NumPy Terminologies: 5
NumPy: 5
Array: 5
ndarray: 5
Shape: 6
Axis: 6
Broadcasting: 6
Slicing: 6
Element-wise Operation: 6
Reshape: 6
Dot Product: 6
Linspace: 6
arange: 7
Pandas Terminologies: 7
Pandas: 7
DataFrame: 7
Series: 7
Index: 7

GroupBy: 7

Aggregation: 8

Filtering: 8

Pivot Table: 8

Merging: 8

Concatenation: 8

Missing Data (NaN): 8

Apply: 8

Data Science:

- A multidisciplinary field focused on extracting knowledge and insights from data using scientific methods, algorithms, and systems.

Machine Learning (ML):

- A subset of artificial intelligence where systems learn from data to make predictions or decisions without explicit programming.

Supervised Learning:

- A type of machine learning where the model is trained on labeled data, meaning both input and output are provided.

Unsupervised Learning:

- A type of machine learning where the model is trained on unlabeled data, and it identifies patterns and relationships without explicit output labels.

Regression:

- A statistical method used to model the relationship between a dependent variable and one or more independent variables.

Classification:

- A type of supervised learning where the goal is to assign labels to inputs from predefined categories.

Clustering:

- An unsupervised learning technique used to group similar data points together.

Dimensionality Reduction:

- The process of reducing the number of features or variables in a dataset while maintaining the essential information.

Feature Engineering:

- The process of transforming raw data into features that better represent the underlying problem to the machine learning model.

Train/Test Split:

- Dividing data into training and test sets to evaluate model performance. The training set is used to train the model, while the test set is used for evaluation.

Cross-Validation:

- A technique to assess the performance of a model by partitioning the data into subsets and training the model on some subsets while validating it on others.

Overfitting:

- When a model learns the training data too well, including noise, making it perform poorly on unseen data.

Underfitting:

- When a model is too simple and fails to capture the underlying patterns in the data.

Python-Specific Data Science Terminologies:

Python:

- A high-level programming language widely used in data science for its simplicity, versatility, and extensive libraries.

List Comprehension:

- A concise way to create lists in Python. Example: [x**2 for x in range(10)] generates a list of squares from 0 to 9.

Lambda Function:

- A small, anonymous function defined using the keyword lambda. Example: lambda x: x**2 defines a function that squares the input.

NumPy Terminologies:

NumPy:

- A fundamental Python library for numerical computing that provides support for arrays, matrices, and a wide variety of mathematical operations.

Array:

- A data structure provided by NumPy that is used to store elements of the same type, typically numbers. Arrays are more efficient than Python lists for numerical operations.

```
#Python Code

import numpy as np

arr = np.array([1, 2, 3])
```

ndarray:

- The primary NumPy object for N-dimensional arrays, which can have one or more dimensions (1D, 2D, or more).

Shape:

- A tuple indicating the size of each dimension of the array. Example: A 3x4 matrix will have a shape of (3, 4).

Axis:

- The dimensions along which operations like sum, mean, or any other mathematical computation are performed. Axis 0 typically refers to rows, while axis 1 refers to columns.

Broadcasting:

- A feature in NumPy that allows operations between arrays of different shapes by "stretching" smaller arrays across larger ones where appropriate.

```
#python

arr = np.array([1, 2, 3])

result = arr + 5  # Adds 5 to each element of the array
```

Slicing:

- A method to extract a portion of an array. Example: arr[1:4] extracts elements from index 1 to 3.

Element-wise Operation:

- Operations applied to corresponding elements in arrays. Example: arr1 + arr2 adds elements of two arrays element by element.

Reshape:

- A function that changes the shape of an array without changing its data. Example: arr.reshape(2, 3) changes a 1D array into a 2x3 matrix.

Dot Product:

- A mathematical operation that takes two equal-length sequences of numbers and returns a single number, often used in vector and matrix multiplication.

```
python

np.dot(arr1, arr2)
```

Linspace:

- A function that generates linearly spaced values over a specified range. Example: np.linspace(0, 10, 5) generates 5 numbers between 0 and 10.

arange:

- Similar to Python's range() but returns a NumPy array. Example: np.arange(0, 10, 2) generates an array [0, 2, 4, 6, 8].

Pandas Terminologies:

Pandas:

- A Python library used for data manipulation and analysis, providing data structures like Series and DataFrame for efficient data handling.

DataFrame:

- A two-dimensional, size-mutable, and potentially heterogeneous tabular data structure in Pandas. It is similar to a spreadsheet or SQL table.

```
python


df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
```

Series:

- A one-dimensional labeled array capable of holding any data type. It is like a column in a DataFrame.

```
python


s = pd.Series([1, 2, 3], index=['a', 'b', 'c'])
```

Index:

- Labels that identify rows or columns in a DataFrame or Series. Pandas automatically creates an index for rows or columns unless specified.

GroupBy:

- A method to split the data into groups based on some criteria, apply a function to each group, and then combine the results.

```
python


df.groupby('column_name').sum()
```

Aggregation:

- Operations like sum, mean, or count that are applied to grouped data. Example: df.groupby('column').agg({'value': 'sum'}).

Filtering:

- The process of selecting rows or columns based on specific conditions. Example: df[df['A'] > 10] filters rows where column A is greater than 10.

Pivot Table:

- A table that summarizes data by grouping and applying aggregate functions, similar to pivot tables in Excel.

  python

  df.pivot_table(values='column_name', index='row_name', aggfunc='mean')

Merging:

- The process of combining two DataFrames based on a common column or index, similar to SQL joins.

  python

  pd.merge(df1, df2, on='column_name')

Concatenation:

- The process of combining two or more DataFrames or Series either vertically or horizontally.

  python

  pd.concat([df1, df2], axis=0) # Concatenates vertically

Missing Data (NaN):

- A placeholder for missing data in Pandas. Methods like fillna() or dropna() are used to handle missing values.

Apply:

- A function that applies a custom function or lambda expression to each row or column.

  python

*df['column'].apply(lambda x: x**2)*

- **Join**:

    1. A method to combine DataFrames using their index, similar to SQL joins (left, right, inner, outer).

python

df1.join(df2, how='inner')