

PYTHON_LIBRARIES

NumPy

Purpose

Provides support for large multi-dimensional arrays and matrices.

Key Features

- Powerful N-dimensional array object.
- Functions for performing mathematical operations on arrays.
- Tools for integrating C/C++ and Fortran code.
- Useful for linear algebra, Fourier transform, and random number capabilities.

Example

```
import numpy as np

# Creating an array
arr = np.array([1, 2, 3, 4, 5])
print(arr)

# Performing arithmetic operations
arr2 = arr * 2
print(arr2)

# Creating a 2D array (matrix)
matrix = np.array([[1, 2, 3], [4, 5, 6]])
print(matrix)

# Matrix multiplication
matrix2 = np.dot(matrix, matrix.T)
print(matrix2)
```

Pandas

Purpose

Offers data manipulation and analysis tools.

Key Features

- Data structures like Series (1D) and DataFrame (2D) for handling structured data.
- Functions for reading/writing data between in-memory data structures and different file formats (e.g., CSV, Excel).
- Tools for data alignment, merging, and reshaping.
- Time series-specific functionality.

Example

```
import pandas as pd

# Creating a DataFrame
data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35]}
df = pd.DataFrame(data)
print(df)

# Reading data from a CSV file
df = pd.read_csv('data.csv')
print(df.head())

# Data manipulation
df['Age'] = df['Age'] + 1
print(df)

# Filtering data
filtered_df = df[df['Age'] > 30]
print(filtered_df)
```

Matplotlib

Purpose

Plotting and visualization library.

Key Features

- Comprehensive library for creating static, animated, and interactive visualizations.
- Wide range of plot types including line, bar, scatter, histogram, and 3D plots.
- Highly customizable plots with fine-grained control over plot elements.

Example

```
import matplotlib.pyplot as plt
```

```
# Line plot
```

```
x = [1, 2, 3, 4, 5]
```

```
y = [2, 3, 5, 7, 11]
```

```
plt.plot(x, y)
```

```
plt.xlabel('X-axis')
```

```
plt.ylabel('Y-axis')
```

```
plt.title('Line Plot')
```

```
plt.show()
```

```
# Bar plot
```

```
plt.bar(x, y)
```

```
plt.xlabel('X-axis')
```

```
plt.ylabel('Y-axis')
```

```
plt.title('Bar Plot')
```

```
plt.show()
```

Scikit-learn

Purpose

Machine learning library.

Key Features

- Simple and efficient tools for data mining and data analysis.
- Includes classification, regression, clustering algorithms.
- Tools for model selection, preprocessing, and evaluation.
- Built on NumPy, SciPy, and matplotlib.

Example

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Load dataset
iris = load_iris()
X, y = iris.data, iris.target

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a model
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)
```

```
# Evaluate the model

accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy}')
```

Seaborn

Purpose

Statistical data visualization based on Matplotlib.

Key Features

- High-level interface for drawing attractive and informative statistical graphics.
- Built-in themes for styling Matplotlib graphics.
- Functions for visualizing distributions, categorical data, and fitting/regression relationships.
- Capabilities for multi-plot grids and visualizing linear regression models.

Example

```
import seaborn as sns
import pandas as pd

# Load dataset
df = sns.load_dataset('iris')

# Basic scatter plot
sns.scatterplot(x='sepal_length', y='sepal_width', data=df)
plt.title('Scatter Plot with Seaborn')
plt.show()

# Histogram
sns.histplot(df['sepal_length'], kde=True)
```

```
plt.title('Histogram with KDE')
```

```
plt.show()
```

```
# Pair plot
```

```
sns.pairplot(df, hue='species')
```

```
plt.title('Pair Plot')
```

```
plt.show()
```

