Data Emgginering

PENGANTAR PYTHON UNTUK DATA ENGINEERING

PERTEMUAN

3

HANSEN LOUISTHER, S.KOM, M.M.S.I

1. Pengenalan Python dan Pustaka Terkait

Python adalah bahasa pemrograman tingkat tinggi yang terkenal karena kesederhanaan dan kemudahan penggunaannya. Dalam dunia data engineering, Python sangat populer karena memiliki banyak pustaka yang kuat dan fleksibel untuk pengolahan data. Berikut adalah pengenalan beberapa pustaka utama yang sering digunakan dalam data engineering.

1. Pengenalan Python dan Pustaka Terkait

Pandas

Pandas adalah pustaka Python yang menyediakan struktur data dan alat analisis data yang mudah digunakan. Pustaka ini sangat kuat untuk manipulasi data dan analisis data tabular.

- Instalasi: pip install pandas
- Struktur Data Utama: DataFrame dan Series
- Fitur Utama:
  - Membaca dan menulis data dari/ke berbagai format seperti CSV, Excel, SQL, dan JSON.
  - Pemrosesan data yang efisien: filtering, merging, reshaping, pivoting, dll.
  - Analisis data statistik dan deskriptif.

2. Operasi Dasar dengan Pandas

Contoh Penggunaan Pandas:

```
import pandas as pd

# Membaca data dari file CSV
data = pd.read_csv('data.csv')

# Menampilkan beberapa baris pertama dari data
print(data.head())

# Memilih kolom tertentu
selected_columns = data[['kolom1', 'kolom2']]

# Menambahkan kolom baru
data['kolom_baru'] = data['kolom1'] + data['kolom2']
```

2. Operasi Dasar dengan Pandas

NumPy
adalah pustaka utama untuk komputasi ilmiah dengan Python. Ini mendukung array multidimensi besar dan berbagai fungsi matematika untuk operasi pada array tersebut.

- Instalasi: pip install numpy
- Struktur Data Utama: ndarray (N-dimensional array)
- Fitur Utama:
  - Operasi aritmatika pada array.
  - Fungsi matematis dasar dan lanjutan: linear algebra, statistik, Fourier transform.
  - Dukungan untuk array besar dan operasi vektor.

2. Operasi Dasar dengan Pandas

Contoh Penggunaan NumPy:

```
import numpy as np

# Membuat array NumPy
arr = np.array([1, 2, 3, 4, 5])

# Operasi aritmatika
arr = arr * 2

# Statistik dasar
mean = np.mean(arr)
std_dev = np.std(arr)

# Mengubah bentuk array
reshaped_arr = arr.reshape(5, 1)
```

3. SQLAlchemy

SQLAlchemy
adalah pustaka SQL Toolkit dan Object Relational Mapper (ORM) untuk Python. Ini memberikan cara yang efisien dan fleksibel untuk bekerja dengan basis data.

- Instalasi: pip install sqlalchemy
- Fitur Utama:
  - Koneksi ke berbagai jenis basis data (MySQL, PostgreSQL, SQLite, dll.)
  - Pemodelan objek basis data menggunakan ORM.
  - Eksekusi perintah SQL langsung.

3. SQLAlchemy

Contoh Penggunaan SQLAlchemy:

```
from sqlalchemy import create_engine, MetaData, Table, Column, Integer, String

# Membuat engine untuk basis data SQLite
engine = create_engine('sqlite:///my_database.db')

# Definisi tabel
metadata = MetaData()
users = Table('users', metadata,
              Column('id', Integer, primary_key=True),
              Column('name', String),
              Column('age', Integer))

# Membuat tabel di basis data
metadata.create_all(engine)
```

3. SQLAlchemy

Contoh Penggunaan SQLAlchemy:

```
# Menyisipkan data
with engine.connect() as conn:
    conn.execute(users.insert(), [
        {'name': 'Alice', 'age': 25},
        {'name': 'Bob', 'age': 30}
    ])

# Membaca data
with engine.connect() as conn:
    result = conn.execute(users.select())
    for row in result:
        print(row)
```

3. SQLAlchemy

PySpark
adalah antarmuka Python untuk Apache Spark, sistem komputasi cluster yang cepat dan umum. PySpark memungkinkan pemrosesan data besar dalam skala besar menggunakan Python.

- Instalasi: pip install pyspark
- Fitur Utama:
  - Pemrosesan data terdistribusi.
  - Analisis data besar menggunakan RDD dan DataFrame.
  - Dukungan untuk SQL, machine learning, dan streaming data.

3. SQLAlchemy

Contoh Penggunaan PySpark:

```
from pyspark.sql import SparkSession

# Membuat SparkSession
spark = SparkSession.builder.appName('DataEngineering').getOrCreate()

# Membaca data dari file CSV
df = spark.read.csv('data.csv', header=True, inferSchema=True)

# Menampilkan skema DataFrame
df.printSchema()

# Menampilkan beberapa baris data
df.show()
```

3. SQLAlchemy

Contoh Penggunaan PySpark:

```
# Pemrosesan data
df_filtered = df.filter(df['age'] > 30)

# Menyimpan data ke file CSV
df_filtered.write.csv('filtered_data.csv')
```

Terima kasih!