누구나 빅데이터 처리 할 수 있는

APACHE PySpark™

Part 1

PySpark란?

- **Apache Spark**를 Python 언어에서 사용할 수 있도록 제공하는 라이브러리입니다. Apache Spark는 빅데이터 처리와 분석을 위한 분산 컴퓨팅 플랫폼이며, 매우 빠르고 대용량 데이터를 효율적으로 처리할 수 있습니다.
- Spark의 **Python API**로, Spark의 강력한 기능을 Python 코드로 쉽게 사용할 수 있게 해줍니다. 이로 인해 Spark의 분산 컴퓨팅 능력을 활용해 대규모 데이터를 병렬 처리하거나 클러스터에서 분석하는 작업을 Python으로 수행할 수 있습니다.
- Hadoop과 같은 분산 스토리지 시스템(HDFS, S3 등)과도 잘 통합되어 있어, 다양한 소스로부터 데이터를 읽고 처리하는데 적합합니다.

PySpark의 주요 특징

① **분산 처리**: Spark는 데이터를 여러 노드에 분산하여 병렬로 처리할 수 있습니다. PySpark는 이러한 Spark의 장점을 그대로 Python 코드에서 사용할 수 있게 해줍니다.

② **DataFrame API**: PySpark는 Pandas와 유사한 DataFrame API를 제공하여 대용량 데이터를 쉽게 처리하고 분석할 수 있습니다. SQL 쿼리처럼 데이터에 접근하거나 변환 작업을 수행할 수 있습니다.

③ **RDD (Resilient Distributed Dataset)**: PySpark는 Spark의 기본 데이터 구조인 RDD를 지원하며, 분산 처리와 복구 기능을 제공합니다.

④ **호환성**: PySpark는 Spark의 모든 주요 기능(스트리밍, 머신러닝, 그래프 처리 등)을 Python으로 사용할 수 있으며, 여러 클러스터 환경에서 동작합니다.

⑤ **머신러닝**: PySpark는 **MLlib**라는 머신러닝 라이브러리를 포함하고 있어, 대규모 데이터를 활용한 머신러닝 모델을 학습시킬 수 있습니다.

PySpark 설치

```
!pip install pyspark
! pip install findspark
```

PySpark 데이터셋 읽기

```
from pyspark.sql import SparkSession
#다양한 데이터 소스로부터 데이터를 읽는 메서드를 제공하는 DataFrameReader를 반환
spark = SparkSession.builder.appName("아이디").config("spark.driver.bindAddress", "127.0.0.1").getOrCreate()
df = spark.read.csv('main_dataset.csv')

#저장을 하고 싶을때
df.write.csv("cokerdataset.csv")
```

PySpark 데이터셋 읽기

Parquet

CSV

Json

PySpark 행 읽기

```
 from pyspark.sql import SparkSession
# 데이터셋 행 읽기
 df.show()

# 데이터셋 10 행 읽기
 df.show(10)
```

PySpark 데이터 선택 및 고르기

```
from pyspark.sql import SparkSession

# column1 그리고 column2 고르기

select_df = df.select("column1", "column2")


# 데이터 필터

filter_df = df.filter(df.column1>100)
```

PySpark 데이터 열 삭제

```
from pyspark.sql import SparkSession
```

```
df.drop = df.drop("column1")
```