ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

BÁO CÁO MÔN HỆ ĐIỀU HÀNH MỞ RỘNG (CO201D)

ĐỀ TÀI:

KHUNG PHẦN MỀM PHÂN TÍCH DỮ LIỆU DÙNG LÕI SPARK

Giảng viên hướng dẫn: Thầy Nguyễn Quang Hùng

Sinh viên thực hiện:

Họ và tên	**Mã số sinh viên**
Trần Công Minh Quân	2012528
Huỳnh Ngọc Như	2010495
Huỳnh Hữu Quyết Thắng	2014533

Thành phố Hồ Chí Minh, tháng 4 – 2022

Mục lục

Mô tả đề tài

Xu hướng hướng dữ liệu (data-driven) là giải pháp tất yếu, tiên tiến để giải quyết nhiều bài toán. Có nhiều nguồn dữ liệu để thu thập như: hệ thống cảm biến (sensor), mạng xã hội, camera, mobile phone, v.v.. và dữ liệu thu thập là dữ liệu lớn ở dạng có cấu trúc và phi cấu trúc. Dữ liệu được thu thập theo dạng dòng (stream). Đòi hỏi một khung phần mềm có khả năng phân tích dữ liệu với nhiều định dạng khác nhau cũng như dữ liệu dạng dòng (data streaming). Spark là một công cụ phân tích dữ liệu mạnh mẽ được sử dụng nhiều trong công nghiệp Google, Facebook, Amazone, v.v. với các module về: SQL, Streaming, Machine Learning, GraphX; hỗ trợ ngôn ngữ Scala, Java & Python cũng nhiều có nhiều dự án của đối tác khác (Third-party projects). Bài toán chính là phát triển một khung phần mềm có khả năng phân tích dữ liệu dựa trên Spark. Công việc trong nghiên cứu này là tiếp tục phát triển những tính năng mới nên có nhiều thuận lợi. Nhiệm vụ (yêu cầu về nội dung và số liệu ban đầu):

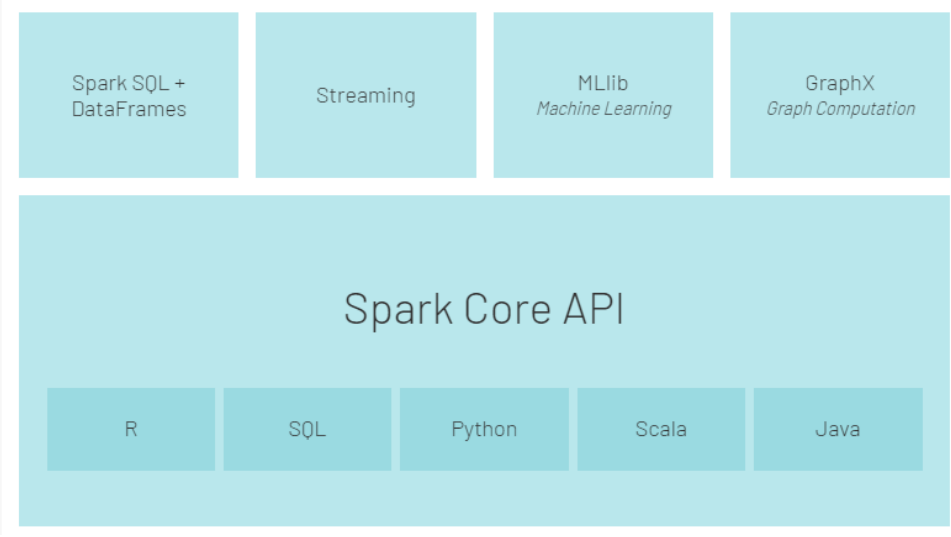
- Tìm hiểu Spark [1] và viết report kỹ thuật;
- Tìm hiểu Kafka [2] để nhận dữ liệu streaming từ Kafka về Saprk;
- Đọc tài liệu, triển khai lại giải pháp tích hợp một hàm mới về Fuzzy logic vào Spark [3]
- (môi trường thử nghiệp được triển khai trên hệ thống máy SuperNode-XP tại HPC Lab);
- Phát triển phát triển một khung phần mềm có khả năng phân tích dữ liệu dựa trên Spark;
- Triển khai thử nghiệm giải pháp: có điều kiện triển khai thực tế thu thập dữ liệu nông nghiệp, giao thông, y tế, giáo dục;
- Thử nghiệm, đánh giá và cải tiến;
- Viết bài báo khoa học.

Tìm hiểu về Apache Spark

1. Tổng quan về Apache Spark

Apache Spark là một khung(framework) xử lý song song với mã nguồn mở được phát triển dành cho việc xử lý dữ liệu quy mô lớn trên các máy tính phân cụm (clustered computers) [1]. Spark Core cung cấp các APIs cấp cao trong nhiều ngôn ngữ (Java, Scala, Python và R) và một loạt các công cụ liên quan dành cho việc quản lý và phân tích dữ liệu. Bên cạnh lõi xử lý Spark Core, môi trường Apache Spark API cũng gồm các thư viện, modules sau:

- Spark SQL dành cho SQL và việc xử lý dữ liệu ở dạng có cấu trúc (structured data)
- MLib thư viện danh cho lập trình học máy cho phép người dùng áp dụng các hoạt động thống kê nâng cao cho dữ liệu trong Spark và xây dựng các ứng dụng xứng quanh.
- GraphX dành cho xử lý đồ thị
- Structured Streaming để tính toán gia tăngvà xử lý dữ liệu dạng dòng (streaming)



Hình ảnh *"Hệ sinh thái" của Apache Spark. Nguồn ảnh: https://databricks.com/*

Về quá trình phát triển, Apache Spark được phát triển từ năm 2019 bởi phòng thí nghiệm AMPLab thuộc trường Đại học California, Berkeley. Sau này, Spark đã được trao cho Apache Software Foundation vào năm 2013 và được phát triển cho

đến nay. Trước khi Spark ra đời, Hadoop đang là một công cụ mạnh mẽ và phổ biến, tuy nhiên Hadoop có những hạn chế nhất định và Spark ra đời để cải thiện các hạn chế đó. Cụ thể, Spark cho phép xây dựng các mô hình dự đoán nhanh chóng với việc tính toán được thực hiện trên một nhóm các máy tính, có thể thính toán cũng lúc trên 1 nhóm máy tính(cluster computing); có thể tính toán cũng lúc trên toàn bộ tập dữ liệu mà không cần phải trích xuất mẫu tính toán thử nghiệm. Tốc độ xử lý của Spark có được do việc tính toán được thực hiện cùng lúc trên nhiều máy khác nhau. Đồng thời việc tính toán được thực hiện ở bộ nhớ trong (in-memories) hay thực hiện hoàn toàn trên RAM [3]. Đây cũng là điểm cải tiến nổi bật nhất cả Spark so với Haddop khi Haddop đòi hỏi các thao tác đều phải thực hiện trên ổ cứng. Nhờ tận dụng việc tính toán trong bộ nhớ trong và các tối ưu hoá khác, Spark có thể chạy nhanh hơn 10 lần so với Haddop ở trên đĩa cứng và 100 lần khi chạy trên bộ nhớ RAM [4.]

2. Điểm nổi bật của Spark

Tốc độ: Được thiết kế tối ưu cho hiệu suất, như đã đề cập Spark có thể nhanh hơn 100 lần so với Hadoop trong việc xử lý dữ liệu quy mô lớn bằng cách khai thác các tính toán bộ nhớ và các tối ưu hóa khác. Khi dữ liệu được lưu trữ trên đĩa cứng, Spark cũng cho tốc độ xử lý rất nhanh và đang nắm giữ kỉ lục thế giới cho sắp xếp quy mô lớn trong đĩa cứng (large-scale on-disk sorting)[5].

Đa hỗ trợ, thân thiện: Các API cho thao tác trên dữ liệu lớn của Spark đa phần dễ sử dụng. Spark cung cấp hơn 100 thao tác cho dữ liệu chuyển đổi (transforming data) và các khung APIs cho xử lý dữ liệu bán cấu trúc (semi structured data). Ngoài ra, Spark hỗ trợ đa ngôn ngữ.

Tích hợp: Spark cung cấp một gói tích hợp các thư viện cấp cao hỗ trợ SQL, streaming data, học máy (ML) và thuật toán đồ thị cho các phân tích nâng cao.

Dự án mã nguồn mở về xử lý dữ liệu lớn nhất: Kể từ khi được phát hành đến nay, Apache Spark đã được ứng dụng bởi nhiều doanh nghiệp lớn như Netflix, Yahoo, eBay.. bởi tốc độ và khả năng liên kết nhiều loại cơ sở dữ liệu và chạy các loại ứng dụng phân tích khác nhau của Spark. Nó nhanh chóng trở thành cộng đồng mã nguồn mở lớn nhất danh cho xử lý dữ liệu lớn với hơn 1000 người đóng góp tử hơn 250 tổ chức [6].

3. Chi tiết về Apache Spark Core

Apache Spark Core là nền tảng mà tất cả các chức năng khác của Spark dựa trên đó và phát triển lên. Spark Core hỗ trợ các tính toán trong bộ nhớ (in-memory computation) vì vậy cho tốc độ xử lý nhanh và là nền tảng của việc xử lý song song và phân phối dữ liệu lớn.

Các đặc điểm chính của Apache Spark core là:

- Phụ trách các chức năng I/O căn bản.
- Có vai trò quan trọng trong lập trình và quan sát các vai trò của các cụm Spark.
- Phân phối tác vụ (task).
- Phục hồi lỗi.
- Sử dụng tính toán trong bộ nhớ để cải tiến MapReduce.

Spark Core được nhúng với RDD (Bộ dữ liệu phân tán có khả năng phân phối – resilient distributed dataset). RDD là một trong những lớp trừu tượng của Spark. Spark RDD xử lý dữ liệu phân vùng trên tất cả các nodes của 1 cụm (cluster). Nó chứa chúng trong nhóm bộ nhớ của cụm dưới dạng 1 đơn vị (unit) [7].

4. Chi tiết về Apache Spark Streaming

Spark streaming là một chức năng dựa trên nền tảng Spark Core danh cho xử lý dữ liệu dạng dòng trực tiếp (live data stream) hỗ trợ khả năng mở rộng, thông lượng cao (high-througput) và khả năng chịu lỗi (fault tolerant). Spark có thể truy cập dữ liệu từ các nguồn như Kafka, Flume, Kinesis hoặc TCP socket. Nó có thể dùng nhiều thâutj toán khác nhau. Cuối cũng dữ liệu nhận được được ghi vào hệ thống tệp, cơ sở dữ liệu và bảng điều khiển trực tiếp (live dashboards).

Spark sử dụng Micro-batching để phát trực tuyến thời gian thực (real-time streaming). Micro-batching là một kỹ thuật cho phép 1 quá trình hoặc tác vụ có thể xử lý một luồng dưới dạng một chuỗi các lô dữ liệu nhỏ (stream as a sequence of small batches of data). Spark streaming nhóm dữ liệu trực tiếp thành các batches nhỏ sau đó cung cấp cho các hệ thống batch để xử lý.

Hình ảnh *cho quá trình xử lý dữ liệu dạng dòng (stream data). Nguồn: https://spark.apache.org/*

Có thể chia Spark Streaming thành 3 giai đoạn:

Một là thu thập (GATHERING). Spark cung cấp 2 loại nguồn cho dữ liệu dòng tích hợp (built-in streaming sources):

- Nguồn cơ bản: Những nguồn có sẵn trong StreamingContext API như hệ thống tập tin, kết nối ổ cắm (socket).
- Nguồn nâng cao: Những nguồn như Kafka, Flume, Kinesis thông qua các lớp tiện ích bổ sung.

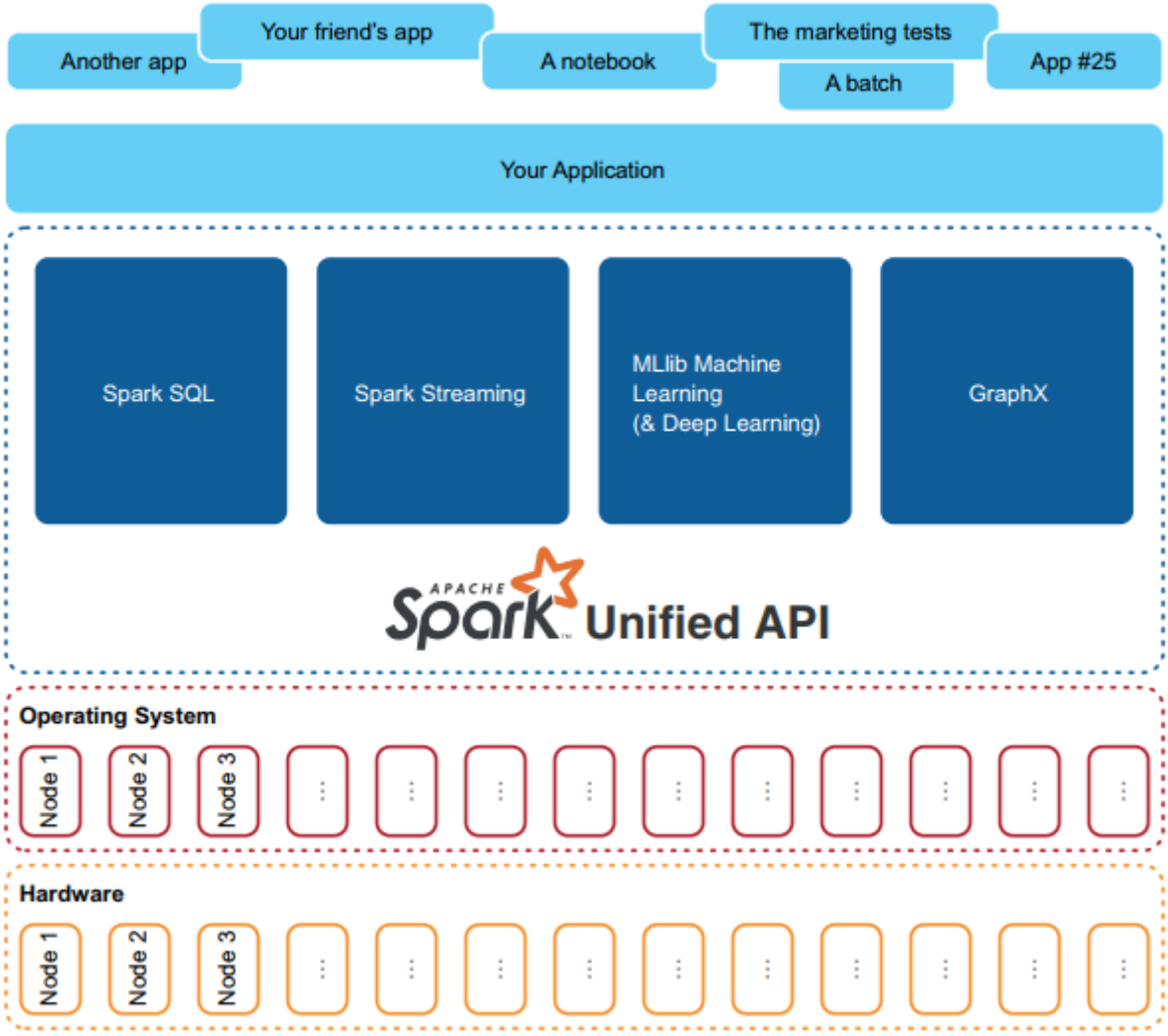
Hai là xử lý (PROCESSING). Những dữ liệu đã được thu thập sẽ được xử lý với một thuật toán phức tạp thể hiện bằng một hàm cấp cao ví dụ: ánh xạ, reduce, join và window.

Ba là lưu trữ (STRORAGE). Dữ liệu đã xử lý được đẩy ra hệ thống tệp, cơ sở dữ liệu và trang tổng quan trực tiếp. Spark streaming cũng cung cấp một lớp trừu tượng cấp cao gọi là luồng tuỳ ý (discretized stream) DStream. DStream đại diện cho một luồng dữ liệu liên tục và có thể được tạo ra từ các nguồn như Kafka và Kinesis. Bên trong DStream được biểu diễn như chuỗi các RDDs [8].

Hình ảnh *mô tả các nguồn dữ liệu dạng dòng (stream data) đầu ra, đầu vào mà Spark streaming hỗ trợ. Nguồn: https://spark.apache.org/*

5. Các thành phần của Apache Spark

Bốn thành phần chính mà người ta hay gọi là "trụ cột" mang lại sức mạnh của Apache Spark là Spark SQL, Spark Streaming, Spark MLlib (trong Machine Learning) và GraphX được xây dựng trên Spark Core.



Hình: Các thành phần của Spark

Spark Core: là engine thực thi chung cho nền tảng Spark, tất cả các chức năng khác được xây dựng trên Spark Core. Nó cung cấp khả năng tính toán trên bộ nhớ RAM và cả bộ dữ liệu tham chiếu trong các hệ thống external storage, cung cấp mô hình thực thi tổng quát để hỗ trợ nhiều ứng dụng đa dạng khác nhau cũng như cho phép nhiều lựa chọn về ngôn ngữ lập trình (Scala, Java, Python và R).

Spark SQL: để thực hiện các thao tác với dữ liệu có cấu trúc, giống với công việc của SQL trong một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS). Nó cung cấp DataFrames và cũng có thể hoạt động như một distributed SQL query engine. Spark SQL có sự tương thích mạnh mẽ với hệ sinh thái của Spark.

Các đặc điểm chính của Spark SQL:

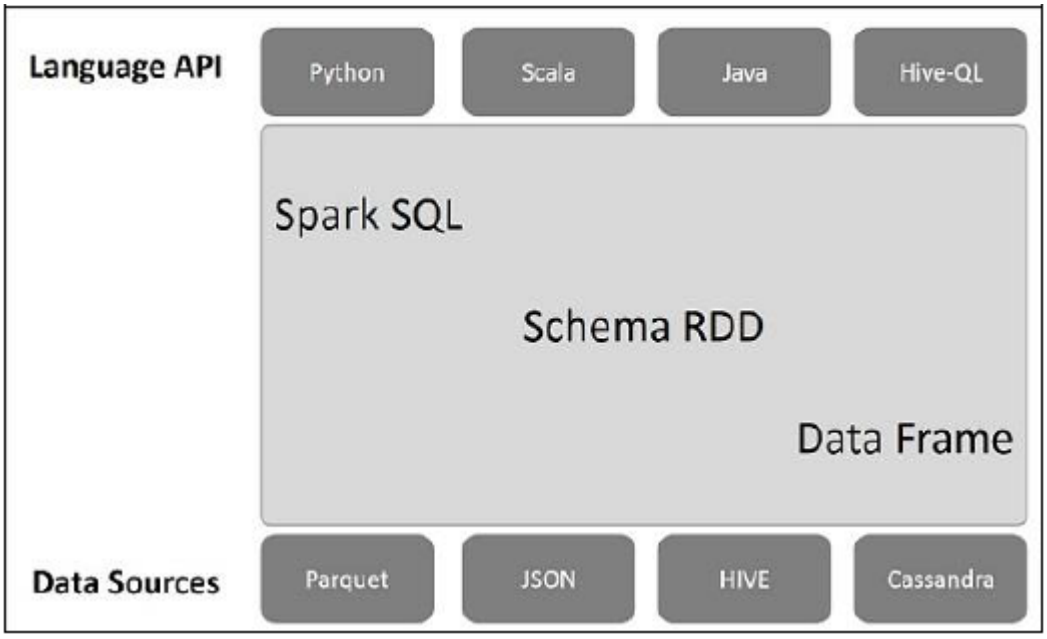
Integrated (tích hợp): có thể tích hợp Spark SQL với các chương trình Spark và truy vấn dữ liệu có cấu trúc trong các chương trình Spark.

Unified Data Access (truy cập dữ liệu nhất quán): tải và truy vấn dữ liệu từ nhiều nguồn khác nhau. Schema-RDDs cung cấp một giao diện duy nhất để làm việc hiệu quả với dữ liệu có cấu trúc, bao gồm Apache Hive tables, parquet files và JSON files.

Hive Compatibility (tương thích): you can run unmodified Hive queries on existing warehouses in Spark SQL. With existing Hive data, queries and UDFs, Spark SQL offer full compatibility.

Standard Connectivity: kết nối thông qua JDBC hoặc ODBC.

Scalability (khả năng mở rộng): Spark SQL tận dụng cấu trúc RDD giúp hỗ trợ công việc lớn và khả năng chịu lỗi truy vấn. Hơn nữa, nó dùng chung engine cho cả interactive và long queries.



Hình: Kiến trúc cơ bản của Spark SQL

Ba lớp chính trong kiến trúc của Spark SQL:

Language API: Spark tương thích với nhiều ngôn ngữ khác nhau và Spark SQL, nó cũng được hỗ trợ bởi các API như Python, HiveQL, Scala và Java.

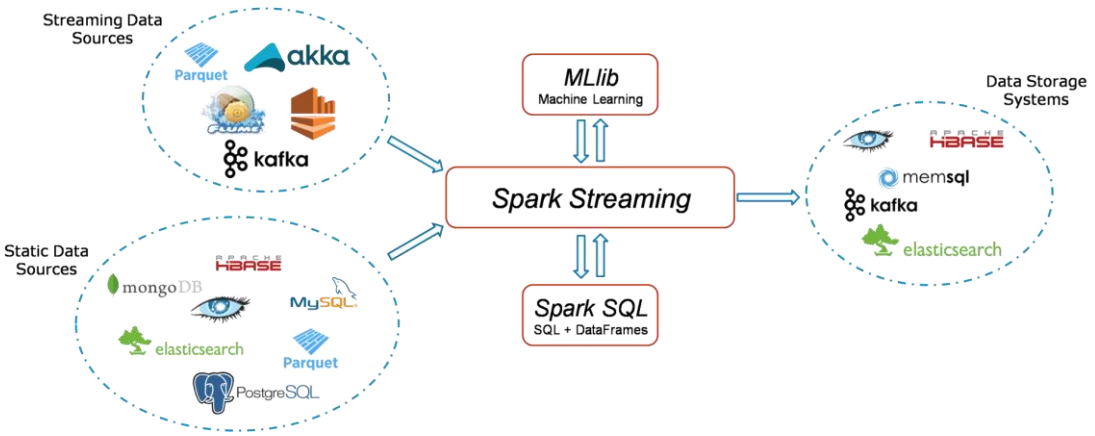
Schema RDD: vì Spark SQL làm việc trên schema, tables và records nên có thể dùng SchemaRDD hay dataframe như một table tạm thời.

Data Sources: nguồn dữ liệu cho Spark SQL như JSON document, HIVE tables, Parquet file và Cassandra database.

Spark SQL với DataFrame luôn có hiệu năng cao hơn rất nhiều so với sử dụng RDD. Còn so với các công nghệ tương tự khác như Impala hay Shark thì thời gian thực thi với Spark SQL cũng luôn rất ấn tượng.

Spark Streaming: nhiều ứng dụng cần khả năng xử lý và phân tích không chỉ với dữ liệu theo lô (batch data) mà còn với dữ liệu theo luồng (stream data) trong thời gian thực. Spark Streaming cho phép các ứng dụng tương tác và phân tích trên cả streaming và historical data, đồng thời kế thừa đặc tính dễ sử dụng và khả năng chịu lỗi của Spark. Nó dễ dàng tích hợp với nhiều hệ thống dữ liệu phổ biến khác (HDFS, Flume, Kafka và Twitter)

Dữ liệu có thể được nhận từ nhiều nguồn như Kafka, Kinesis hay TCP sockets và có thể được xử lý bằng các giải thuật phức tạp thông qua các hàm cấp cao như map, reduce, join và window. Dữ liệu sau khi được xử lý có thể được đưa vào filesystems, databases và live dashboards. Ngoài ra, có thể áp dụng các giải thuật MLlib và GraphX của Spark cho data streams.



Hình: Spark Streaming

Spark Streaming nhận live input data stream và chia dữ liệu đó thành các lô (batch), sau đó được xử lý bởi Spark Engine và cuối cùng đưa ra các lô dữ liệu đã được xử lý.

Hình: Quá trình xử lý dữ liệu với Spark Streaming

Spark Streaming đưa ra khái niệm *discretized stream* hay *Dstream* đại diện cho một luồng dữ liệu liên tục (continuous stream of data). DStream có thể được tạo từ các luồng dữ liệu đầu vào từ các nguồn như Kafka và Kinesis hoặc bằng cách áp dụng các high-level operations trên các DStreams khác. DStream được biểu diễn dưới dạng một chuỗi các RDDs.

Spark Mllib (Machine Learning library): dùng trong machine learning hay các tiện ích gần đây trong deep learning. Machine Learning đang dần trở thành một phần quan trọng trong khai thác dữ liệu lớn (big data) để có được những thông tin chi tiết hữu ích. Được xây dựng trên Spark, Mllib là một thư viện machine learning có thể cung cấp các tính năng, giải thuật nâng cao (multiple iterations to increase accuracy) và tính toán với tốc độ cao (nhanh gấp 100 lần so với MapReduce). Các hỗ trợ nâng cao như:

Các giải thuật ML: classification (phân loại), regression (hồi quy), clustering (gom cụm) và collaborative filtering (lọc cộng tác).

Featurization: feature extraction, transformation, dimensionality reduction, và selection.

Pipelines: tools for constructing, evaluating, và tuning ML Pipelines.

Persistence: saving and load algorithms, models, và Pipelines.

Utilities: linear algebra (đại số tuyến tính), statistics (thống kê), data handling (xử lý dữ liệu), …

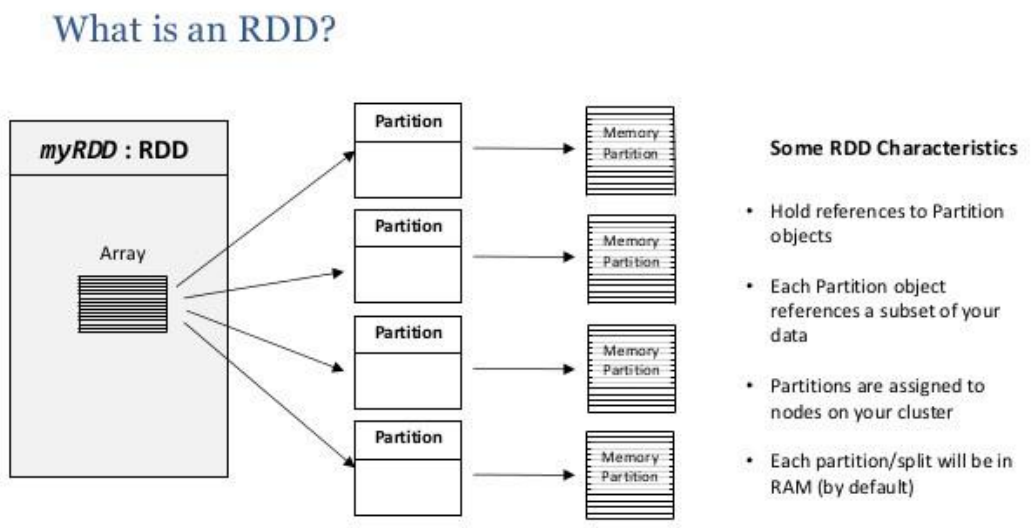
GraphX: giúp khai thác cấu trúc dữ liệu đồ thị như xây dựng, chuyển đổi và phân tích trên dữ liệu có cấu trúc đồ thị với quy mô lớn, tính toán đồ thị song song (graph-parallel computation). Cung cấp một công cụ thống nhất cho trích xuất – biến đổi – tải (ETL: extraction – transform – load). Các ứng dụng của GraphX: PageRank, Fraud Detection, Geographic information system, Disaster management.

6. RDD (Resilient Distributed Dataset)

Một trong những nhà phát triển Spark, Matei Zaharia đã đưa ra khái niệm RDD như sau: "*RDD is a distributed memory abstraction that lets programmers perform in-memory computations on large clusters in a fault-tolerant manner*."

RDD (Resilient Distributed Dataset), tạm dịch là tập dữ liệu phân tán linh hoạt, là một cấu trúc dữ liệu cơ bản của Spark đại diện cho tập dữ liệu phân tán cho phép tính toán trên các node khác nhau của một cụm máy chủ (cluster) với khả năng chịu lỗi (fault-tolerance).

RDD có thể được tạo từ một tập dữ liệu có sẵn trong ngôn ngữ sử dụng như Java, Python, Scala hoặc lấy từ các hệ thống lưu trữ như HDFS, Hbase hoặc các cơ sở dữ liệu quan hệ.

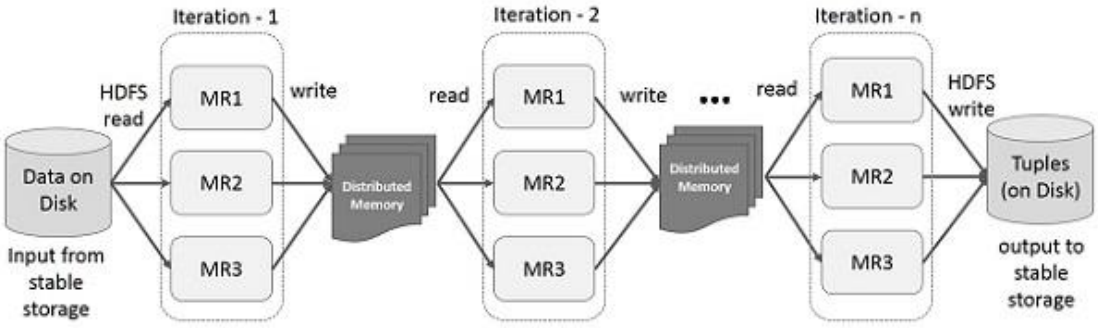


Hình RDD

Đặc điểm quan trọng của 1 RDD là số partitions. Một RDD bao gồm nhiều partition nhỏ, mỗi partition này đại diện cho 1 phần dữ liệu phân tán. Khái niệm partition là logical, tức là 1 node xử lý có thể chứa nhiều hơn 1 RDD partition. Theo mặc định, dữ liệu các partitions sẽ lưu trên memory. Với việc chia nhỏ dữ liệu thành các partition và cơ chế lazy evaluation của Spark có thể chỉ cần vài chục GB ram và 1 chương trình được thiết kế tốt để xử lý 1TB dữ liệu.

Iterative Operations on Spark RDD

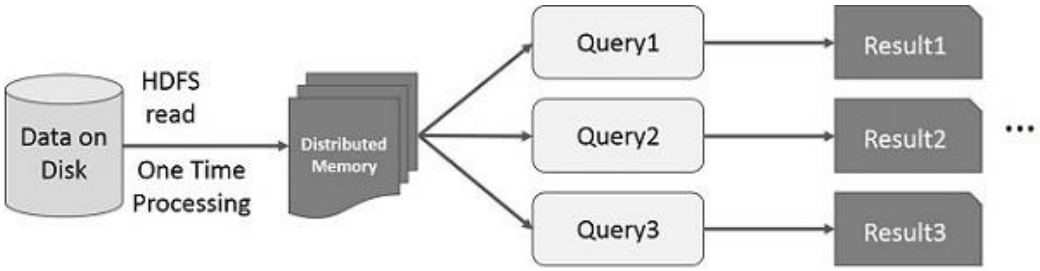
Hình dưới đây minh hoạ cho các hoạt động lặp lại trên Spark RDD. Nó lưu trữ các kết quả trung gian trong một bộ nhớ phân tán thay vì đĩa giúp tăng tốc độ. Chú ý rằng nếu bộ nhớ RAM không đủ thì nó sẽ lưu dữ liệu sang đĩa.



Hình: Iterative Operations on Spark RDD

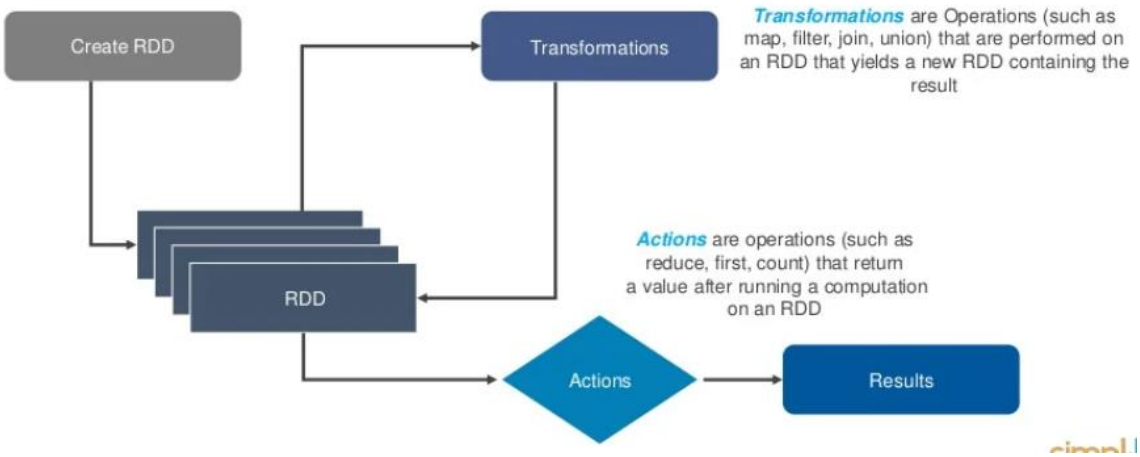
Interactive Operations on Spark RDD

Hình dưới đây minh hoạ cho các hoạt động tương tác trên Spark RDD. Nếu các truy vấn khác nhau được chạy lặp lại trên cùng một tập dữ liệu, thì dữ liệu cụ thể này có thể được lưu trong bộ nhớ để có thời gian thực thi tốt hơn.



Hình: Interactive Operations on Spark RDD

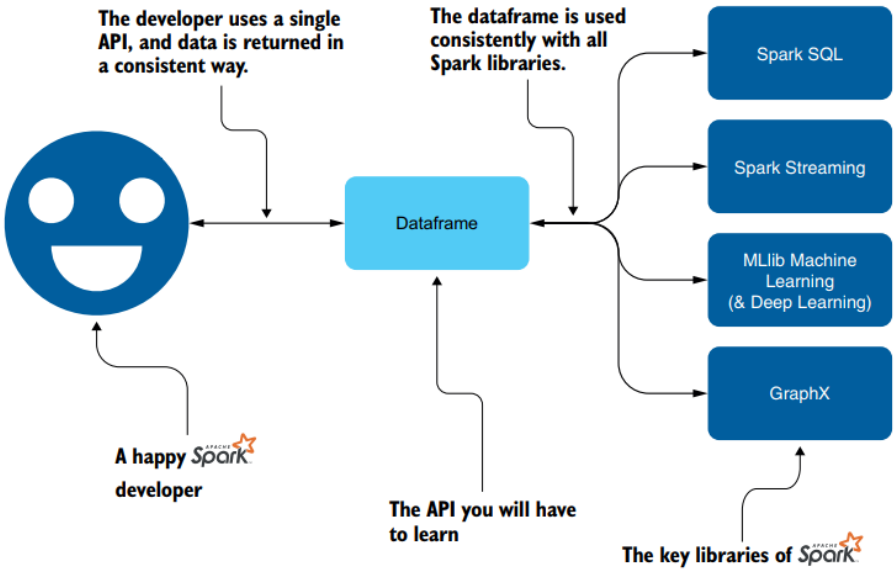
RDD cung cấp các transformation và action hoạt động như DataFrame hay DataSets. Transformation xử lý các thao tác lazily và Action xử lý thao tác cần xử lý tức thời.



Hình: Transformations, Actions

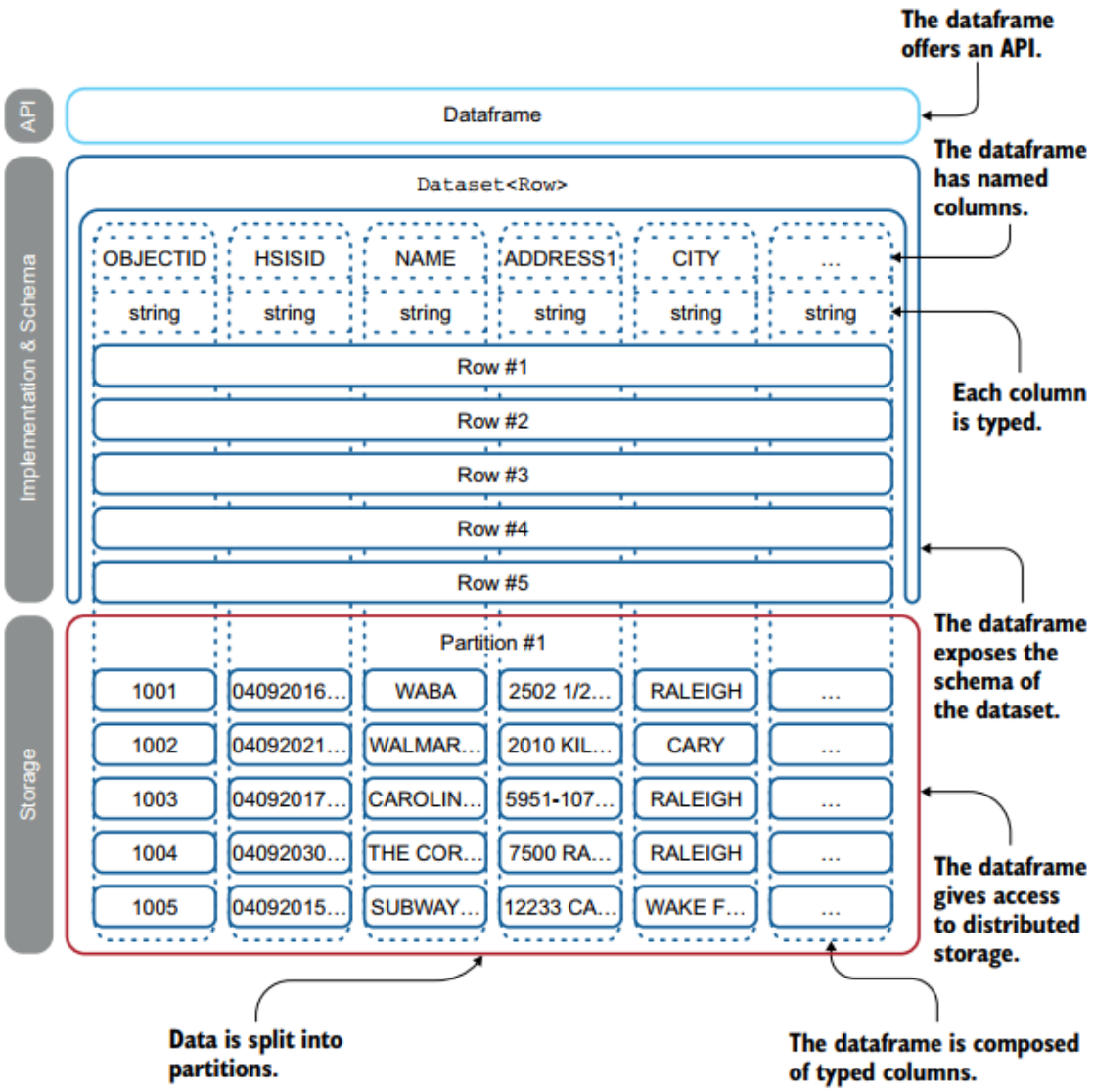
7. Dataframe

Dataframe đồng thời là một cấu trúc dữ liệu và là một API. Spark's dataframe API được sử dụng trong Spark SQL, Spark Streaming, MLlib (trong machine learning) và GraphX để thao tác với các cấu trúc dữ liệu đồ thị trong Spark. API này giúp việc truy cập các thư viện được đơn giản và nhất quán, thay vì phải dùng một API khác nhau cho mỗi thư viện. (xem hình 1)



Hình: Một API chung cho Spark SQL, ML, streaming, GraphX

Dataframe là một tập các bản ghi dữ liệu bất biến và phân tán được tổ chức thành các cột được đặt tên và kiểu tương ứng. Nó giống như một bảng trong hệ cơ sở dữ liệu quan hệ.

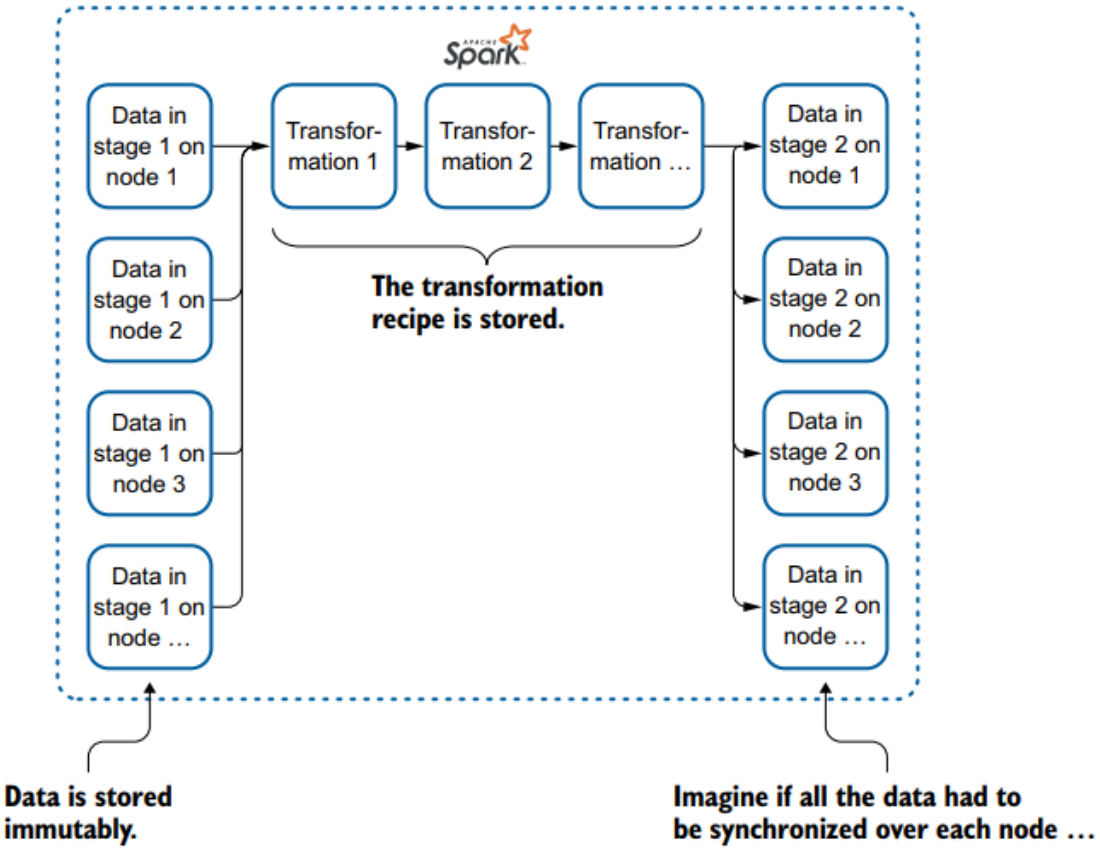


Hình: Tổ chức dữ liệu của Dataframe

Tính bất biến (immutability) được xem là một tính chất nền tảng và quan trọng được sử dụng của Spark để tối ưu quá trình xử lý dữ liệu.

Dataframes hay datasets và RDDs được xem là các bộ lưu trữ bất biến (có nghĩa là không thể thay đổi được). Dữ liệu của 1 DF sau khi tạo ra sẽ không thay đổi, nếu muốn chỉnh sửa ta cần tạo ra DF mới từ DF ban đầu, thông qua DF API.

Bằng cách chỉ lưu trữ transformation recipe, giúp giảm sự phụ thuộc vào việc lưu trữ dữ liệu và tăng độ tin cậy (khả năng phục hồi). Dữ liệu không được lưu trữ ở stage 2. (xem hình).



Hình: Chỉ giữ các transformation recipe.

Spark SQL hỗ trợ rất nhiều nguồn dữ liệu như file, database…. Với *DataFrameReader* có thể dễ dàng đọc dữ liệu từ nhiều nguồn, nhiều định dạng để rạo ra 1 DataFrame, từ đó có thể sử dụng các API của Spark SQL tương tác với chúng.

Code demo

1. Mô tả

Dữ liệu được lấy từ 2 file diem_f1_md.xlsx và diem_f2_md.xlsx, đối tượng khai thác là điểm các môn học của ngành khoa học máy tính của các sinh viên máy tính, từ năm 2015 đến năm 2017. Tập dữ liệu này được phân tích nhằm 2 mục đích:

- Khám phá số lượng sinh viên đã học đủ tất cả các môn cơ sở ngành/nhóm ngành bắt buộc của 3 năm đầu tiên, theo chương trình đào tạo,
- Tìm ra mối tương quan giữa tập điểm các môn học bắt buộc có ảnh hưởng đến việc lựa chọn môn tự chọn của sinh viên.

2. Data Processing

Đầu tiên, ta tiến hành cài pyspark và load dữ liệu của 2 file excel.

```
!pip install pyspark
```

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

```
spark = SparkSession.builder.master("local").appName("DataPoints").getO
rCreate()
```

```
spark
```

```
import pandas as pd
!pip install openpyxl
```

```
pandas_df = pd.read_excel('diem_f1_md.xlsx', engine = 'openpyxl')
```

```
pandas_df.to_csv("f1.csv")
```

```
df1 = spark.read.csv("./f1.csv", inferSchema=True, header=True)
```

```
pd.read_excel('diem_f2_md.xlsx', engine = 'openpyxl').to_csv("f2.csv")
```

```
df2 = spark.read.csv("./f2.csv", inferSchema=True, header=True)
```

```
df = df1.union(df2)
df.count()
```

```
1000000
```

```
df.printSchema()
```

```
root
 |-- _c0: integer (nullable = true)
 |-- NAMHOC: integer (nullable = true)
 |-- TENHK: integer (nullable = true)
 |-- NHHK: integer (nullable = true)
 |-- F_MAMH: string (nullable = true)
 |-- F_TENMHVN: string (nullable = true)
 |-- F_DVHT: double (nullable = true)
 |-- F_PHTRAMKT: integer (nullable = true)
 |-- F_PHTRAMTH: integer (nullable = true)
 |-- F_MANH: string (nullable = true)
 |-- F_TO: string (nullable = true)
 |-- STT: string (nullable = true)
 |-- F_MAKH: string (nullable = true)
 |-- F_TENLOP: string (nullable = true)
 |-- F_KHOI: string (nullable = true)
 |-- F_MANG: string (nullable = true)
 |-- F_TENNGVN: string (nullable = true)
 |-- MASV1: integer (nullable = true)
 |-- KT: double (nullable = true)
 |-- TILEKT: double (nullable = true)
 |-- BT: double (nullable = true)
 |-- TILEBT: double (nullable = true)
 |-- BTLDA: double (nullable = true)
 |-- TILEBTLDA: double (nullable = true)
 |-- TN: double (nullable = true)
 |-- TILETN: double (nullable = true)
 |-- THI: double (nullable = true)
 |-- TILETHI: double (nullable = true)
 |-- TKET: string (nullable = true)
 |-- F_DIEM1: double (nullable = true)
 |-- F_DIEM2: double (nullable = true)
 |-- F_DIEM10: string (nullable = true)
 |-- GHICHU: string (nullable = true)
 |-- T: string (nullable = true)
```

```
df_subject=df.filter((df['F_MAKH'] == 'MT') & (df['NAMHOC'] >= 2015)) \
      .groupBy('F_TENMHVN').count().orderBy('F_TENMHVN')
```

```
df_subject.count()
```

Ở đây, ta sẽ liệt kê ra các môn có trong 2 file dữ liệu, qua đó biết được tên môn được nhập liệu trong file là như thế nào, để tìm dữ liệu cho chính xác.

```
178
```

```
df_subject.show(df_subject.count(), truncate = False)
```

Xây dựng các hàm tạo dataframe chứa điểm của các sinh viên học môn học cụ thể, và làm sạch dữ liệu:

```
# loại bỏ các giá trị điểm 13, 21, RT
def cleanSubjectPointDF(dataframe):
    res = dataframe.filter(((dataframe['TKET'] >= 0) &(dataframe['TKET'
] <= 10)))
    return res;

def createSubjectPointDF(SubjectName):
    res = df.filter((df['F_TENMHVN'] == SubjectName) & (df['F_MAKH'] ==
 'MT') & (df['NAMHOC'] >= 2015)) \
    .select(df.columns[3:4] + df.columns[12:13] + df.columns[16:29])
    cleaned_res = cleanSubjectPointDF(res)
    return cleaned_res;
```

Từ đây, tiến hành xây dựng dataframe riêng cho từng môn, từ Hệ thống số cho đến Phân tích và thiết kế giải thuật:

```
df_HTS = createSubjectPointDF('Hệ thống số')
```

```
df_HTS.count()
```

```
1107
```

```
df_HTS.show()
```

```
+----+------+--------------------+--------+---+------+----+------+-----+---------+---+------+---+-------+----+
|NHHK|F_MAKH|          F_TENNGVN|   MASV1| KT|TILEKT|  BT|TILEBT|BTLDA|TILEBTLDA| TN|TILETN|THI|TILETHI|TKET|
+----+------+--------------------+--------+---+------+----+------+-----+---------+---+------+---+-------+----+
| 151|    MT|   Khoa học Máy tính|17348205|4.0|  30.0|null|  null| null|     null|6.5|  20.0|4.0|   50.0| 4.5|
| 151|    MT|   Khoa học Máy tính|63980467|6.5|  30.0|null|  null| null|     null|9.5|  20.0|9.0|   50.0| 8.5|
| 151|    MT|   Khoa học Máy tính|55919265|4.5|  30.0|null|  null| null|     null|8.0|  20.0|3.5|   50.0| 4.5|
| 151|    MT|   Khoa học Máy tính|39528837|5.5|  30.0|null|  null| null|     null|8.0|  20.0|5.0|   50.0|   6|
| 151|    MT|   Khoa học Máy tính|32004923|5.0|  30.0|null|  null| null|     null|6.5|  20.0|6.0|   50.0|   6|
| 151|    MT|   Khoa học Máy tính|75478642|3.0|  30.0|null|  null| null|     null|7.0|  20.0|4.5|   50.0| 4.5|
| 151|    MT|   Khoa học Máy tính|51303549|3.0|  30.0|null|  null| null|     null|9.0|  20.0|3.0|   50.0|   4|
| 151|    MT|MT&CôngNghệ ThôngTin|25323654|5.0|  30.0|null|  null| null|     null|9.5|  20.0|6.0|   50.0| 6.5|
| 151|    MT|   Khoa học Máy tính| 3661541|7.5|  30.0|null|  null| null|     null|7.0|  20.0|4.0|   50.0| 5.5|
| 151|    MT|   Khoa học Máy tính|60513944|2.0|  30.0|null|  null| null|     null|6.5|  20.0|3.0|   50.0| 3.5|
| 151|    MT|   Khoa học Máy tính|27879950|6.5|  30.0|null|  null| null|     null|6.0|  20.0|8.0|   50.0|   7|
| 151|    MT|   Khoa học Máy tính| 6796972|5.0|  30.0|null|  null| null|     null|6.5|  20.0|4.5|   50.0|   5|
| 151|    MT|MT&CôngNghệ ThôngTin| 6553341|4.0|  30.0|null|  null| null|     null|8.0|  20.0|4.5|   50.0|   5|
| 151|    MT|MT&CôngNghệ ThôngTin|74316373|5.0|  30.0|null|  null| null|     null|9.0|  20.0|5.5|   50.0|   6|
| 151|    MT|   Khoa học Máy tính|98307698|3.5|  30.0|null|  null| null|     null|6.5|  20.0|3.0|   50.0|   4|
| 151|    MT|   Khoa học Máy tính|17317141|6.5|  30.0|null|  null| null|     null|9.0|  20.0|4.0|   50.0|   6|
| 151|    MT|   Khoa học Máy tính|49370219|6.5|  30.0|null|  null| null|     null|9.5|  20.0|9.5|   50.0| 8.5|
| 151|    MT|   Khoa học Máy tính|20065532|7.0|  30.0|null|  null| null|     null|7.5|  20.0|6.0|   50.0| 6.5|
| 151|    MT|   Khoa học Máy tính|80888933|5.0|  30.0|null|  null| null|     null|8.0|  20.0|3.0|   50.0| 4.5|
| 151|    MT|   Khoa học Máy tính|87255890|6.0|  30.0|null|  null| null|     null|8.5|  20.0|8.5|   50.0|   8|
+----+------+--------------------+--------+---+------+----+------+-----+---------+---+------+---+-------+----+
only showing top 20 rows
```

```
df_NMDT = createSubjectPointDF('Nhập môn điện toán')
```

```
df_NMDT.count()
```

```
1082
```

```
-----------------------------------------------------------------
df_CTRR = createSubjectPointDF('Cấutrúc rời rạc cho KHMT')
-----------------------------------------------------------------
df_CTRR.count()
1100
-----------------------------------------------------------------
df_KTLT = createSubjectPointDF('Kỹ thuật lập trình')
-----------------------------------------------------------------
df_KTLT.count()
1306
-----------------------------------------------------------------

df_CTDLGT1 = createSubjectPointDF('Ctrúc dữliệu & giảithuật')
df_CTDLGT2 = createSubjectPointDF('Cấu trúc dữliệu & Gthuật')
df_CTDLGT = df_CTDLGT1.union(df_CTDLGT2)
-----------------------------------------------------------------
df_CTDLGT.count()
1261
-----------------------------------------------------------------
df_LTHDT1 = createSubjectPointDF('Lậptrình hướng đối tượng')
df_LTHDT2 = createSubjectPointDF('Lập trình hướng đốitượng')
df_LTHDT = df_LTHDT1.union(df_LTHDT2)
-----------------------------------------------------------------
df_LTHDT.count()
840
-----------------------------------------------------------------
df_KTMT = createSubjectPointDF('Kiến trúc máy tính')
-----------------------------------------------------------------
df_KTMT.count()
1070
-----------------------------------------------------------------
df_HTS.createOrReplaceTempView("HTS")
df_NMDT.createOrReplaceTempView("NMDT")
df_CTRR.createOrReplaceTempView("CTRR")
df_KTLT.createOrReplaceTempView("KTLT")
-----------------------------------------------------------------
df_CTDLGT.createOrReplaceTempView("CTDLGT")
df_KTMT.createOrReplaceTempView("KTMT")
df_LTHDT.createOrReplaceTempView("LTHDT")
-----------------------------------------------------------------
df_MHH = createSubjectPointDF('Mô hình hóa toán học')
df_HDH = createSubjectPointDF('Hệ điều hành')
```

```
df_HCSDL = createSubjectPointDF('Hệ cơ sở dữ liệu')

df_CNPM = createSubjectPointDF('Công nghệ phần mềm')
df_MMT = createSubjectPointDF('Mạng máy tính')

df_NLNNLT1 = createSubjectPointDF('Ng/lý ngôn ngữ lập trình')
df_NLNNLT2 = createSubjectPointDF('N/ly Ngon Ngu Lap Trinh')
df_NLNNLT = df_NLNNLT1.union(df_NLNNLT2)

df_PTTKGT = createSubjectPointDF('Phân tích và thiết kế gt').union(crea
teSubjectPointDF('PT & Thiết kế giải thuật'))
```

```
df_MHH.createOrReplaceTempView("MHH")
df_HDH.createOrReplaceTempView("HDH")
df_HCSDL.createOrReplaceTempView("HCSDL")

df_CNPM.createOrReplaceTempView("CNPM")
df_MMT.createOrReplaceTempView("MMT")
df_NLNNLT.createOrReplaceTempView("NLNNLT")

df_PTTKGT.createOrReplaceTempView("PTTKGT")
```

Để thuận tiện trong việc truy vấn dữ liệu, ta tạo view tương ứng với mỗi dataframe của môn học như trên. Chúng ta có thể sử dụng lệnh truy vấn như trong các DBMS thông qua spark.sql.

```
diemTK_nam3 = spark.sql("select \
      HTS.MASV1 as MSSV, NMDT.TKET as NMDT, CTRR.TKET as CTRR, HTS.TKET
 as HTS, KTLT.TKET as KTLT, \
      CTDLGT.TKET as CTDLGT, KTMT.TKET as KTMT, LTHDT.TKET as LTHDT, \
      MHH.TKET as MHH, HCSDL.TKET as HCSDL, HDH.TKET as HDH, \
      MMT.TKET as MMT, CNPM.TKET as CNPM, NLNNLT.TKET as NLNNLT, PTTKGT
.TKET as PTTKGT \
      from HTS, NMDT, KTLT, CTRR, CTDLGT, KTMT, LTHDT, MHH, HCSDL, HDH,
 MMT, CNPM, NLNNLT, PTTKGT \
      where HTS.MASV1 = NMDT.MASV1 and NMDT.MASV1 = CTRR.MASV1 and CTRR
.MASV1 = KTLT.MASV1 \
      and KTLT.MASV1 = CTDLGT.MASV1 and CTDLGT.MASV1 = KTMT.MASV1 and K
TMT.MASV1 = LTHDT.MASV1 \
      and LTHDT.MASV1 = MHH.MASV1 and MHH.MASV1 = HCSDL.MASV1 and HCSDL
.MASV1 = HDH.MASV1 \
      and HDH.MASV1 = MMT.MASV1 and MMT.MASV1 = CNPM.MASV1 and CNPM.MAS
V1 = NLNNLT.MASV1 \
      and NLNNLT.MASV1 = PTTKGT.MASV1")
```

```
diemTK_nam3.count()
440
```

```
diemTK_nam3.show()
```

```
+--------+----+----+---+----+------+----+-----+---+-----+---+---+----+------+------+
|    MSSV|NMDT|CTRR|HTS|KTLT|CTDLGT|KTMT|LTHDT|MHH|HCSDL|HDH|MMT|CNPM|NLNNLT|PTTKGT|
+--------+----+----+---+----+------+----+-----+---+-----+---+---+----+------+------+
| 4045426| 8.5| 8.5|  6|   8|   7.5| 8.5|    9|8.5|  7.5|8.5|8.5| 6.5|   7.5|   8.5|
|44827787| 8.5| 9.5|  9| 9.5|    10|   9|  9.5|9.5|    9|8.5|8.5| 8.0|   8.0|  10.0|
|   75321| 6.5| 6.5|  5|   6|   5.5| 6.5|  7.5|  7|    8|6.5|8.0| 7.0|   5.5|   9.0|
|   75321| 6.5| 6.5|  5|   1|   5.5| 6.5|  7.5|  7|    8|6.5|8.0| 7.0|   5.5|   9.0|
|94305108|   7| 7.5|7.5| 5.5|   7.5| 7.5|    8|  8|  8.5|8.5|7.5| 6.5|   7.0|   9.0|
| 6024723| 7.5| 7.5|7.5|   8|   8.5| 7.5|    8|  8|    7|  8|7.5| 7.0|   8.5|   9.0|
|76861612| 7.5|   8|  8|   8|   8.5| 8.5|    9|9.5|    8|8.5|8.5| 6.5|   5.5|   8.0|
|14495747|   7|   6|3.5|   5|   3.5| 3.5|    6|  7|  4.5|  5|6.5| 5.0|   1.5|   5.5|
|14495747|   7|   6|3.5| 2.5|   3.5| 3.5|    6|  7|  4.5|  5|6.5| 5.0|   1.5|   5.5|
|14495747|   7|   6|  1|   5|   3.5| 3.5|    6|  7|  4.5|  5|6.5| 5.0|   1.5|   5.5|
|14495747|   7|   6|  1| 2.5|   3.5| 3.5|    6|  7|  4.5|  5|6.5| 5.0|   1.5|   5.5|
|37734481|   7| 6.5|6.5| 5.5|   6.5| 7.5|  8.5|  7|  6.5|6.5|7.0| 4.5|   0.0|   8.0|
|37734481|   7| 6.5|6.5| 5.5|   6.5| 7.5|  8.5|  7|  6.5|6.5|7.0| 4.5|   0.0|   8.0|
|37734481|   7| 6.5|6.5| 5.5|   6.5| 7.5|  8.5|  7|  6.5|6.5|7.0| 4.5|   0.0|   8.0|
|37734481|   7| 6.5|6.5| 5.5|   6.5| 7.5|  8.5|  7|  6.5|6.5|7.0| 4.5|   0.0|   8.0|
|37734481|   7| 6.5|6.5| 0.5|   6.5| 7.5|  8.5|  7|  6.5|6.5|7.0| 4.5|   0.0|   8.0|
|37734481|   7| 6.5|6.5| 0.5|   6.5| 7.5|  8.5|  7|  6.5|6.5|7.0| 4.5|   0.0|   8.0|
|37734481|   7| 6.5|6.5| 0.5|   6.5| 7.5|  8.5|  7|  6.5|6.5|7.0| 4.5|   0.0|   8.0|
|37734481|   7| 6.5|6.5| 0.5|   6.5| 7.5|  8.5|  7|  6.5|6.5|7.0| 4.5|   0.0|   8.0|
|37734481|   7| 6.5|2.5| 5.5|   6.5| 7.5|  8.5|  7|  6.5|6.5|7.0| 4.5|   0.0|   8.0|
+--------+----+----+---+----+------+----+-----+---+-----+---+---+----+------+------+
only showing top 20 rows
```

Trên đây, có thể thấy nhiều MSSV bị lặp lại, điều này là do sinh viên có thể học một môn nhiều lần để cải thiện điểm. Do đó, theo cơ chế học cải thiện, ta lấy điểm tổng kết cao nhất qua các lần học:

```
diemTK_nam3.createOrReplaceTempView('diemTK_nam3')
```

```
maxdiemTK_nam3 = spark.sql("select MSSV, max(NMDT) as NMDT, max(CTRR) a
s CTRR, max(HTS) as HTS, max(KTLT) as KTLT, \
      max(CTDLGT) as CTDLGT, max(KTMT) as KTMT, max(LTHDT) as LTHDT, \
      max(MHH) as MHH, max(HCSDL) as HCSDL, max(HDH) as HDH, \
      max(MMT) as MMT, max(CNPM) as CNPM, max(NLNNLT) as NLNNLT, max(PT
TKGT) as PTTKGT \
      from diemTK_nam3 group by MSSV")
```

```
maxdiemTK_nam3.count()
122
```

```
maxdiemTK_nam3.show()
```

```
+--------+----+----+---+----+------+----+-----+---+-----+---+---+----+------+------+
|    MSSV|NMDT|CTRR|HTS|KTLT|CTDLGT|KTMT|LTHDT|MHH|HCSDL|HDH|MMT|CNPM|NLNNLT|PTTKGT|
+--------+----+----+---+----+------+----+-----+---+-----+---+---+----+------+------+
|   75321| 6.5| 6.5|  5|   6|   5.5| 6.5|  7.5|  7|    8|6.5|8.0| 7.0|   5.5|   9.0|
| 2683782| 7.5|   8|6.5|   9|   8.5| 7.5|    7|8.5|    8|  7|8.5| 8.5|   7.0|   7.0|
| 4045426| 8.5| 8.5|  6|   8|   7.5| 8.5|    9|8.5|  7.5|8.5|8.5| 6.5|   7.5|   8.5|
| 4536867|   8|   8|  9| 7.5|     9| 9.5|  9.5|9.5|    9|9.5|9.5| 8.0|   9.5|  10.0|
| 5763862|   7|   6|  8|   6|   7.5|   7|  6.5|8.5|  7.5|8.5|7.5| 6.5|   5.0|   7.0|
| 6024723| 7.5| 7.5|7.5|   8|   8.5| 7.5|    8|  8|    7|  8|7.5| 7.0|   8.5|   9.0|
| 6136739| 8.5|   7|  7| 8.5|   5.0| 5.5|  8.5|6.5|  7.5|8.0|6.5| 6.0|   5.0|   9.0|
| 7675525| 8.5| 7.5|  9|   8|   8.5| 8.5|    9|9.5|    9|8.5|9.0| 8.0|   7.0|   8.5|
|10372040| 6.5|   6|4.5| 2.5|   4.5|   6|  6.5|  6|  6.0|5.0|6.5| 5.5|   3.5|   6.5|
|10520631|   5| 7.5|4.5|   5|   4.5| 3.5|  4.5|  6|  4.5|6.5|6.5| 5.5|   5.0|   8.0|
|11968298| 8.5| 8.5|7.5|   8|   8.5| 8.5|    7|  8|    8|7.5|8.0| 7.5|   5.5|   7.5|
|12118103| 6.5|   6|6.5| 7.5|     8| 7.5|  8.5|  7|  8.5|  9|9.0| 8.0|   7.5|  10.0|
|12201042| 7.5|   7|7.5| 7.5|     8|   8|    9|  9|  7.5|6.5|8.5| 9.0|   6.5|   8.5|
|12515604|   7|   7|  6| 5.5|   6.5|   5|    7|  6|    6|6.5|7.0| 7.0|   1.0|   8.5|
|12540524| 6.5| 7.5|7.5|   7|     8| 7.5|  9.5| 10|  8.5|8.5|9.5| 8.5|   8.5|   9.5|
|12967654| 7.5|   7|6.5|   8|   6.5| 7.5|  8.5|8.5|    8|  7|7.5| 8.0|   5.0|   8.5|
|13000115|   7| 5.5|6.5|   9|   5.5| 6.5|    8|  6|    6|7.5|7.0| 7.0|   5.5|   8.0|
|13793340|   7| 7.5|  7| 7.5|     5|   5|  6.5|6.5|  5.5|  7|7.5| 6.5|   2.5|   8.5|
|14495747|   7|   6|3.5|   5|   3.5| 3.5|    6|  7|  4.5|  5|6.5| 5.0|   1.5|   5.5|
|15401017| 8.5|   8|  6|   8|     8|   8|  7.5|7.5|  8.5|7.5|8.5| 8.0|   8.5|   8.5|
+--------+----+----+---+----+------+----+-----+---+-----+---+---+----+------+------+
only showing top 20 rows
```

Từ năm 2015 đến 2017, có tổng cộng 122 sinh viên đã học đầy đủ tất cả các môn cơ sở ngành/nhóm ngành bắt buộc theo chương trình khoa học máy tính. Tập dữ liệu này sẽ được xuất ra để phân tích tiếp trong phần Data Visualization.

```
----------------------------------------------------------------
maxdiemTK_nam3.write.csv('diemTK_nam3.csv', header = True)
----------------------------------------------------------------
```

Để ý rằng, nhiều sinh viên đã chọn học các môn tự chọn ngay từ năm 2, chứ không cần đến năm 3. Do đó, chúng ta sẽ xây dựng một tập dữ liệu chứa điểm của các sinh viên đã học qua đầy đủ các môn cơ sở ngành/nhóm ngành bắt buộc của 2 năm học đầu tiên.

```
diemTK_nam2 = spark.sql("select \
     HTS.MASV1 as MSSV, NMDT.TKET as NMDT, CTRR.TKET as CTRR, HTS.TKET
as HTS, KTLT.TKET as KTLT, \
     CTDLGT.TKET as CTDLGT, KTMT.TKET as KTMT, LTHDT.TKET as LTHDT, \
     MHH.TKET as MHH, HCSDL.TKET as HCSDL, HDH.TKET as HDH \
     from HTS, NMDT, KTLT, CTRR, CTDLGT, KTMT, LTHDT, MHH, HCSDL, HDH \
     where HTS.MASV1 = NMDT.MASV1 and NMDT.MASV1 = CTRR.MASV1 and CTRR.
MASV1 = KTLT.MASV1 \
     and KTLT.MASV1 = CTDLGT.MASV1 and CTDLGT.MASV1 = KTMT.MASV1 and KT
MT.MASV1 = LTHDT.MASV1 \
     and LTHDT.MASV1 = MHH.MASV1 and MHH.MASV1 = HCSDL.MASV1 and HCSDL.
MASV1 = HDH.MASV1")
----------------------------------------------------------------
```

```
diemTK_nam2.createOrReplaceTempView('diemTK_nam2')
```

```
maxdiemTK_nam2 = spark.sql("select MSSV, max(NMDT) as NMDT, max(CTRR) a
s CTRR, max(HTS) as HTS, max(KTLT) as KTLT, \
      max(CTDLGT) as CTDLGT, max(KTMT) as KTMT, max(LTHDT) as LTHDT, \
      max(MHH) as MHH, max(HCSDL) as HCSDL, max(HDH) as HDH \
      from diemTK_nam2 group by MSSV")
```

```
maxdiemTK_nam2.count()
```

```
359
```

Từ năm 2015 đến 2017, có tổng cộng 359 sinh viên đã học đầy đủ các môn bắt buộc của 2 năm học đầu. Tiếp theo chúng ta sẽ xem trong số các sinh viên này, họ học những môn tự chọn nào. Các môn tự chọn được xét đến ở đây gồm Nhập môn trí tuệ nhân tạo, Đồ họa máy tính, Mật mã và an ninh mạng.

```
maxdiemTK_nam2.show()
```

```
+-------+----+----+---+----+------+----+-----+----+-----+---+
|   MSSV|NMDT|CTRR|HTS|KTLT|CTDLGT|KTMT|LTHDT| MHH|HCSDL|HDH|
+-------+----+----+---+----+------+----+-----+----+-----+---+
|  75321| 6.5| 6.5|  5|   6|   5.5| 6.5|  7.5|   7|    8|6.5|
|  81657| 7.5|   7|5.5| 7.0|   7.0| 7.5|  7.5| 6.0|  7.5|7.5|
| 209312|   7| 6.5|5.5| 7.5|   4.5| 7.0|  0.0|   7|    6|  8|
| 583833| 6.5|   7|  6| 5.5|   6.5|   7|    8|   7|    6|  7|
|1509861| 7.5|   5|  5| 9.0|   3.5| 6.5|  6.0| 7.0|  7.5|6.5|
|2047807| 6.5| 6.5|7.5| 9.5|   8.5| 6.5|  8.5| 7.5|  8.5|8.5|
|2223555|   9|   8|9.5|10.0|   9.0| 8.0|  8.0|10.0|  8.0|8.5|
|2240989| 8.5|   7|  7| 7.0|   6.0| 8.0|  7.0| 6.5|  7.5|8.0|
|2260878| 7.5| 8.5|8.5|10.0|   8.5| 8.0|  8.0| 8.5|  0.0|9.0|
|2274996| 5.5|   7|  7| 7.5|   7.5|   6|    8|   7|  7.5|7.5|
|2343848|   8|   7|  6| 7.5|   5.0| 7.0|  6.5| 6.5|  0.0|5.5|
|2683782| 7.5|   8|6.5|   9|   8.5| 7.5|    7| 8.5|    8|  7|
|3095962|   7| 6.5|  6| 6.5|   7.5|   6|  7.5| 5.5|  7.5|6.5|
|3419145| 7.5| 6.5|  5|   6|     1|   5|    7|   6|  6.0|  7|
|3546806| 8.5| 7.5|  7| 7.0|   6.5| 7.0|  7.5| 7.0|  7.0|7.0|
|4045426| 8.5| 8.5|  6|   8|   7.5| 8.5|    9| 8.5|  7.5|8.5|
|4536867|   8|   8|  9| 7.5|     9| 9.5|  9.5| 9.5|    9|9.5|
|4650139| 8.5|   8|8.5|10.0|   9.5| 8.5|  8.5| 8.5|  8.5|9.0|
|5665575| 5.5| 5.5|5.0|   6|   6.5|   6|  6.5| 6.5|    7|  8|
|5763862|   7|   6|  8|   6|   7.5|   7|  6.5| 8.5|  7.5|8.5|
+-------+----+----+---+----+------+----+-----+----+-----+---+
only showing top 20 rows
```

```
maxdiemTK_nam2.createOrReplaceTempView('maxdiemTK_nam2')
```

```
df_NMAI = createSubjectPointDF('Nhập môn trítuệ nhân tạo')
df_DHMT = createSubjectPointDF('Đồ họa máy tính')
df_MMANM = createSubjectPointDF('Mật mã & an ninh mạng').union(createSu
bjectPointDF('Mật mã và an ninh mạng'))
```

```
df_NMAI.createOrReplaceTempView("NMAI")
df_DHMT.createOrReplaceTempView("DHMT")
df_MMANM.createOrReplaceTempView("MMANM")
```

```
res_NMAI = spark.sql("select NMAI.MASV1 as MSSV, NMAI.TKET as NMTTNT fr
om NMAI, maxdiemTK_nam2 where NMAI.MASV1 = maxdiemTK_nam2.MSSV")
res_NMAI.count()
```

```
159
```

```
res_DHMT = spark.sql("select DHMT.MASV1 as MSSV, DHMT.TKET as DHMT from
 DHMT, maxdiemTK_nam2 where DHMT.MASV1 = maxdiemTK_nam2.MSSV")
res_DHMT.count()
```

```
120
```

```
res_MMANM = spark.sql("select MMANM.MASV1 as MSSV, MMANM.TKET as MMANM
from MMANM, maxdiemTK_nam2 where MMANM.MASV1 = maxdiemTK_nam2.MSSV")
res_MMANM.count()
```

```
92
```

```
res_NMAI.createOrReplaceTempView('res_NMAI')
res_DHMT.createOrReplaceTempView('res_DHMT')
res_MMANM.createOrReplaceTempView('res_MMANM')
```

```
diemTK_BB_TC = spark.sql("select maxdiemTK_nam2.MSSV, NMDT,CTRR,HTS,KTL
T,CTDLGT,KTMT,LTHDT,MHH,HCSDL,HDH, NMTTNT, DHMT, MMANM \
          from maxdiemTK_nam2 left join res_NMAI on maxdiemTK_nam2.MSSV
 = res_NMAI.MSSV \
          left join res_DHMT on maxdiemTK_nam2.MSSV = res_DHMT.MSSV \
          left join res_MMANM on maxdiemTK_nam2.MSSV = res_MMANM.MSSV")
```

```
diemTK_BB_TC.show()
```

```
+-------+----+----+---+----+------+----+-----+----+-----+---+------+----+-----+
|   MSSV|NMDT|CTRR|HTS|KTLT|CTDLGT|KTMT|LTHDT| MHH|HCSDL|HDH|NMTTNT|DHMT|MMANM|
+-------+----+----+---+----+------+----+-----+----+-----+---+------+----+-----+
|  75321| 6.5| 6.5|  5|   6|   5.5| 6.5|  7.5|   7|    8|6.5|   5.0|null|  7.0|
|  81657| 7.5|   7|5.5| 7.0|   7.0| 7.5|  7.5| 6.0|  7.5|7.5|   5.0|null| null|
| 209312|   7| 6.5|5.5| 7.5|   4.5| 7.0|  0.0|   7|    6|  8|   4.0| 0.0| null|
| 583833| 6.5|   7|  6| 5.5|   6.5|   7|    8|   7|    6|  7|   8.5| 6.0|  7.5|
|1509861| 7.5|   5|  5| 9.0|   3.5| 6.5|  6.0| 7.0|  7.5|6.5|  null|null| null|
|2047807| 6.5| 6.5|7.5| 9.5|   8.5| 6.5|  8.5| 7.5|  8.5|8.5|   9.0| 8.5| null|
|2223555|   9|   8|9.5|10.0|   9.0| 8.0|  8.0|10.0|  8.0|8.5|  null|null| null|
|2240989| 8.5|   7|  7| 7.0|   6.0| 8.0|  7.0| 6.5|  7.5|8.0|  null|null| null|
|2260878| 7.5| 8.5|8.5|10.0|   8.5| 8.0|  8.0| 8.5|  0.0|9.0|  null|null| null|
|2274996| 5.5|   7|  7| 7.5|   7.5|   6|    8|   7|  7.5|7.5|  10.0| 8.0|  7.5|
|2343848|   8|   7|  6| 7.5|   5.0| 7.0|  6.5| 6.5|  0.0|5.5|  null|null| null|
|2683782| 7.5|   8|6.5|   9|   8.5| 7.5|    7| 8.5|    8|  7|  null| 7.0| null|
|3095962|   7| 6.5|  6| 6.5|   7.5|   6|  7.5| 5.5|  7.5|6.5|   7.0| 6.0| null|
|3419145| 7.5| 6.5|  5|   6|     1|   5|    7|   6|  6.0|  7|  null|null| null|
|3546806| 8.5| 7.5|  7| 7.0|   6.5| 7.0|  7.5| 7.0|  7.0|7.0|  null|null| null|
|4045426| 8.5| 8.5|  6|   8|   7.5| 8.5|    9| 8.5|  7.5|8.5|  null| 8.5| null|
|4536867|   8|   8|  9| 7.5|     9| 9.5|  9.5| 9.5|    9|9.5|  null|10.0| 10.0|
|4650139| 8.5|   8|8.5|10.0|   9.5| 8.5|  8.5| 8.5|  8.5|9.0|  null|null| null|
|5665575| 5.5| 5.5|5.0|   6|   6.5|   6|  6.5| 6.5|    7|  8|   6.0| 6.5| null|
|5763862|   7|   6|  8|   6|   7.5|   7|  6.5| 8.5|  7.5|8.5|  null|null|  7.5|
+-------+----+----+---+----+------+----+-----+----+-----+---+------+----+-----+
only showing top 20 rows
```

Sau khi gộp điểm của 3 môn tự chọn vào tập dữ liệu chung, ta sẽ xử lí một chút. Ở cột của 3 môn tự chọn, những giá trị là điểm số sẽ được chuyển thành 1, những giá trị null được chuyển thành 0. Tức là, những sinh viên có điểm sẽ là những người đã quyết định học môn đó. Tập dữ này sẽ được xuất ra và để dành cho phần Data Model.

```
-----------------------------------------------------------------
from pyspark.sql.functions import when

ndf = diemTK_BB_TC.withColumn("NMTTNT", \
              when(diemTK_BB_TC["NMTTNT"] >= 0, 1).otherwise(0))
ndf = ndf.withColumn("DHMT", when(ndf["DHMT"] >= 0, 1).otherwise(0))
ndf = ndf.withColumn("MMANM", when(ndf["MMANM"] >= 0, 1).otherwise(0))
-----------------------------------------------------------------
ndf.count()
359
-----------------------------------------------------------------
ndf.show(359)
-----------------------------------------------------------------
ndf.write.csv('input.csv', header = True)
```

3. Data visualization

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data = pd.read_csv("diemTK_nam3.csv")
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 122 entries, 0 to 121
Data columns (total 15 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   MSSV    122 non-null    int64
 1   NMDT    122 non-null    float64
 2   CTRR    122 non-null    float64
 3   HTS     122 non-null    float64
 4   KTLT    122 non-null    float64
 5   CTDLGT  122 non-null    float64
 6   KTMT    122 non-null    float64
 7   LTHDT   122 non-null    float64
 8   MHH     122 non-null    float64
 9   HCSDL   122 non-null    float64
 10  HDH     122 non-null    float64
 11  MMT     122 non-null    float64
 12  CNPM    122 non-null    float64
 13  NLNNLT  122 non-null    float64
 14  PTTKGT  122 non-null    float64
dtypes: float64(14), int64(1)
memory usage: 14.4 KB
```

```
data.head()
```

	MSSV	NMDT	CTRR	HTS	KTLT	CTDLGT	KTMT	LTHDT	MHH	HCSDL	HDH	MMT	CNPM	NLNNLT	PTTKGT
0	75321	6.5	6.5	5.0	6.0	5.5	6.5	7.5	7.0	8.0	6.5	8.0	7.0	5.5	9.0
1	2683782	7.5	8.0	6.5	9.0	8.5	7.5	7.0	8.5	8.0	7.0	8.5	8.5	7.0	7.0
2	4045426	8.5	8.5	6.0	8.0	7.5	8.5	9.0	8.5	7.5	8.5	8.5	6.5	7.5	8.5
3	4536867	8.0	8.0	9.0	7.5	9.0	9.5	9.5	9.5	9.0	9.5	9.5	8.0	9.5	10.0
4	5763862	7.0	6.0	8.0	6.0	7.5	7.0	6.5	8.5	7.5	8.5	7.5	6.5	5.0	7.0

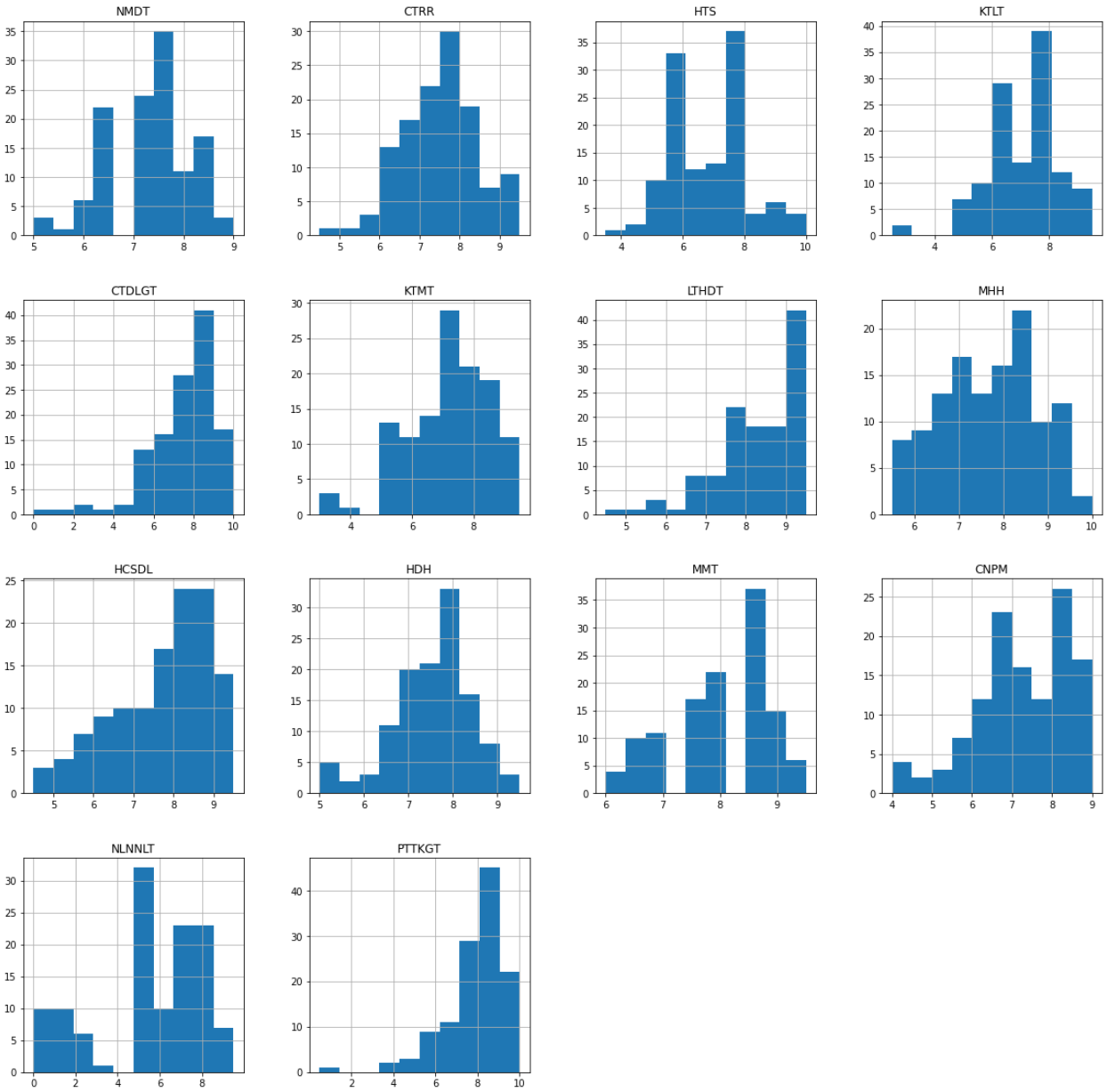
Các giá trị thống kê về điểm số của các môn học.

```
data[['NMDT','CTRR','HTS','KTLT','CTDLGT','KTMT','LTHDT','MHH','HCSDL',
'HDH','MMT','CNPM','NLNNLT','PTTKGT']].describe()
```

	NMDT	CTRR	HTS	KTLT	CTDLGT	KTMT	LTHDT	MHH	HCSDL	HDH	MMT	CNPM	NLNNLT	PTTKGT
count	122.000000	122.000000	122.000000	122.000000	122.000000	122.000000	122.000000	122.000000	122.000000	122.000000	122.000000	122.000000	122.000000	122.000000
mean	7.290984	7.274590	6.844262	7.069672	7.282787	7.221311	8.098361	7.688525	7.471311	7.569672	8.000000	7.024590	5.569672	8.065574
std	0.840209	0.953685	1.272860	1.278098	1.700738	1.337707	1.078795	1.199738	1.211684	0.977676	0.869597	1.185052	2.687064	1.484007
min	5.000000	4.500000	3.500000	2.500000	0.000000	3.000000	4.500000	5.500000	4.500000	5.000000	6.000000	4.000000	0.000000	0.500000
25%	6.500000	6.500000	6.000000	6.000000	6.500000	6.500000	7.500000	7.000000	6.500000	7.000000	7.500000	6.500000	5.000000	7.625000
50%	7.500000	7.500000	7.000000	7.000000	7.500000	7.500000	8.000000	8.000000	8.000000	7.500000	8.000000	7.000000	6.000000	8.500000
75%	7.875000	8.000000	7.500000	8.000000	8.500000	8.000000	9.000000	8.500000	8.500000	8.000000	8.500000	8.000000	7.500000	9.000000
max	9.000000	9.500000	10.000000	9.500000	10.000000	9.500000	9.500000	10.000000	9.500000	9.500000	9.500000	9.000000	9.500000	10.000000

Phổ điểm của từng môn học.

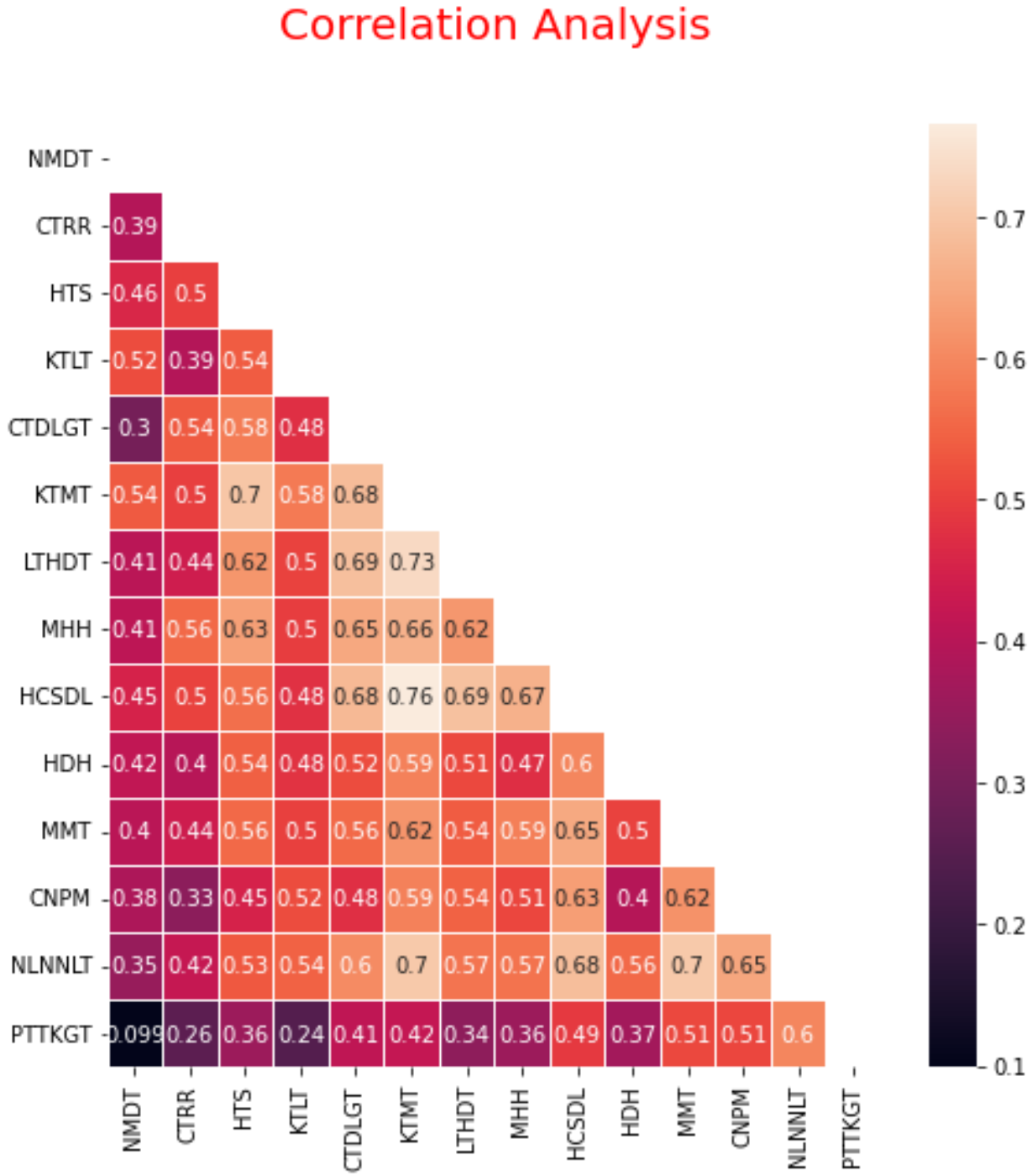
```
data[['NMDT','CTRR','HTS','KTLT','CTDLGT','KTMT','LTHDT','MHH','HCSDL',
'HDH','MMT','CNPM','NLNNLT','PTTKGT']].hist(figsize=(20,20),bins=10)
```



Nhìn vào biểu đồ trên ta có thể thấy phân bố điểm của từng môn học, ví dụ môn CTRR có phân bố điểm tập trung ở mức 7.5 – 8, môn Hệ điều hành có mức điểm tập trung ở trên dưới 8.

Tương quan điểm số giữa các môn học.

```
plt.figure(figsize=(8,8))
plt.title('Correlation Analysis',color='Red',fontsize=20,pad=40)

df = data.loc[:, data.columns!='MSSV']
corr = df.corr()
mask = np.triu(np.ones_like(corr,dtype = bool))
sns.heatmap(df.corr(),mask=mask,annot=True,linewidths=.5);
plt.show()
```

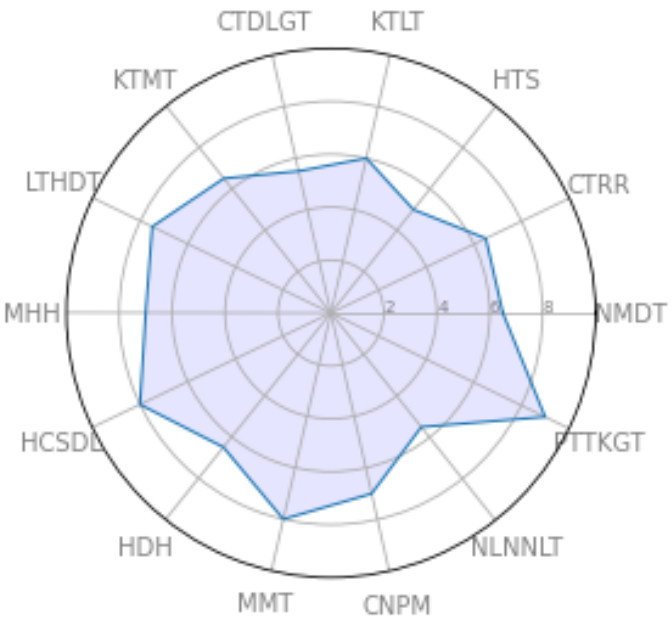


Ở hình trên, các ô có màu càng nhạt thì có mối tương quan các mạnh, ngược lại có mối tương quan không đáng kể. Ví dụ môn KTMT và môn HCSDL tương quan khá mạnh (0.76), vì vậy có thể nói môn KTMT có nhiều ảnh hưởng đến kết quả học tập của môn HCSDL hay một sinh viên có điểm KTMT cao thường cũng sẽ có điểm HCSDL cao.

Radar

```
categories=list(data)[1:]
N = len(categories)

# We are going to plot the first line of the data frame.
```

```
# But we need to repeat the first value to close the circular graph:
values=data.loc[0].drop('MSSV').values.flatten().tolist()
values += values[:1]
values

# What will be the angle of each axis in the plot? (we divide the plot
/ number of variable)
angles = [n / float(N) * 2 * 3.14 for n in range(N)]
angles += angles[:1]

# Initialise the spider plot
ax = plt.subplot(111, polar=True)

# Draw one axe per variable + add labels
plt.xticks(angles[:-1], categories, color='grey', size=10)

# Draw ylabels
ax.set_rlabel_position(0)
plt.yticks([2,4,6,8], ["2","4","6","8"], color="grey", size=7)
plt.ylim(0,10)

# Plot data
ax.plot(angles, values, linewidth=1, linestyle='solid')

# Fill area
ax.fill(angles, values, 'b', alpha=0.1)

# Show the graph
plt.show()
```

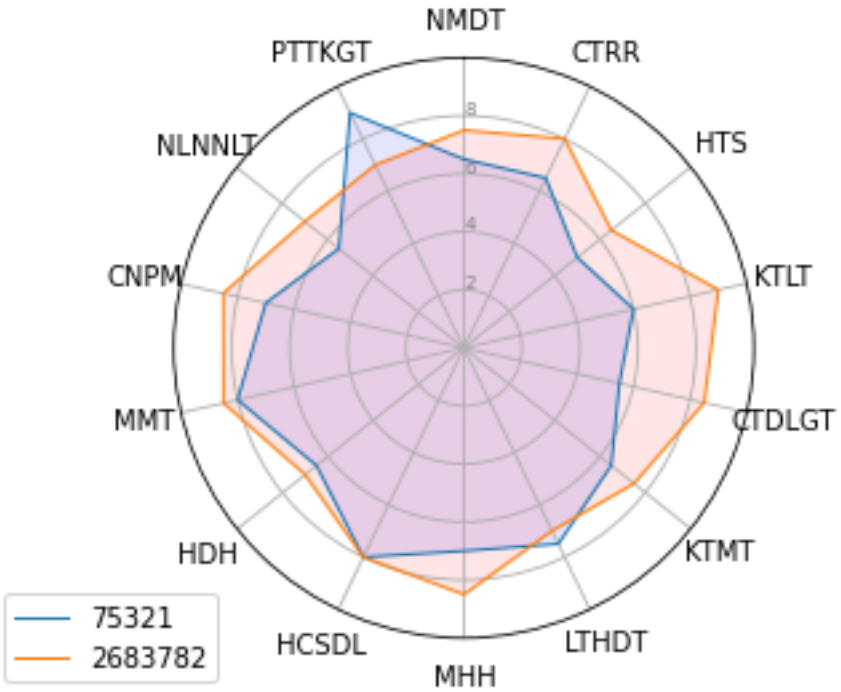


Biểu đồ trên thể hiện điểm số của sinh viên có MSSV là 75321 ứng với 14 môn học. Nhìn vào biểu đồ này có thể nhận thấy sinh viên này thường học tốt ở những môn nào và chưa tốt ở các môn nào. Ví dụ ở trên, sinh viên này có vẻ học các môn ở năm 2 và năm 3 tốt hơn so với các môn ở năm đầu đại học.

Ngoài ra biểu đồ radar cũng giúp ta có thể dễ dàng so sánh kết quả học tập của hai hay một số sinh viên với nhau.

```
# ------- PART 1: Create background

# number of variable
categories=list(data)[1:]
N = len(categories)

# What will be the angle of each axis in the plot? (we divide the plot
/ number of variable)
angles = [n / float(N) * 2 * 3.14 for n in range(N)]
angles += angles[:1]

# Initialise the spider plot
ax = plt.subplot(111, polar=True)

# If you want the first axis to be on top:
ax.set_theta_offset(3.14 / 2)
ax.set_theta_direction(-1)

# Draw one axe per variable + add labels
plt.xticks(angles[:-1], categories)

# Draw ylabels
ax.set_rlabel_position(0)
plt.yticks([2,4,6,8], ["2","4","6","8"], color="grey", size=7)
plt.ylim(0,10)


# ------- PART 2: Add plots

# Plot each individual = each line of the data
# I don't make a loop, because plotting more than 3 groups makes the
chart unreadable

# Ind1
values=data.loc[0].drop('MSSV').values.flatten().tolist()
values += values[:1]
ax.plot(angles, values, linewidth=1, linestyle='solid', label="75321")
ax.fill(angles, values, 'b', alpha=0.1)

# Ind2
values=data.loc[1].drop('MSSV').values.flatten().tolist()
values += values[:1]
ax.plot(angles, values, linewidth=1, linestyle='solid',
label="2683782")
ax.fill(angles, values, 'r', alpha=0.1)

# Add legend
plt.legend(loc='upper right', bbox_to_anchor=(0.1, 0.1))
```

```
# Show the graph
plt.show()
```



Ví dụ, quan sát biểu đồ trên có thể thấy, sinh viên thứ hai (màu cam) có kết quả học tập cao và đồng đều giữa các môn hơn so với sinh viên thứ nhất (màu xanh)

4. Data model

```
!pip install pyspark
-----------------------------------------------------------------
import pyspark
from pyspark.sql import SparkSession

-----------------------------------------------------------------
spark = SparkSession.builder.master("local").appName("Practices").getOr
Create()
-----------------------------------------------------------------
spark
-----------------------------------------------------------------
df = spark.read.csv('input.csv', inferSchema = True, header = True)
-----------------------------------------------------------------
df.printSchema()
```

```
root
 |-- MSSV: integer (nullable = true)
 |-- NMDT: double (nullable = true)
 |-- CTRR: double (nullable = true)
 |-- HTS: double (nullable = true)
 |-- KTLT: double (nullable = true)
 |-- CTDLGT: double (nullable = true)
 |-- KTMT: double (nullable = true)
 |-- LTHDT: double (nullable = true)
 |-- MHH: double (nullable = true)
 |-- HCSDL: double (nullable = true)
 |-- HDH: double (nullable = true)
 |-- NMTTNT: integer (nullable = true)
 |-- DHMT: integer (nullable = true)
 |-- MMANM: integer (nullable = true)
```

```
from pyspark.ml.feature import VectorAssembler
```

```
df.columns
```

```
['MSSV',
 'NMDT',
 'CTRR',
 'HTS',
 'KTLT',
 'CTDLGT',
 'KTMT',
 'LTHDT',
 'MHH',
 'HCSDL',
 'HDH',
 'NMTTNT',
 'DHMT',
 'MMANM']
```

Tạo vector gom các biến độc lập:

```
assembler = VectorAssembler(inputCols = ['NMDT','CTRR','HTS','KTLT','CT
DLGT','KTMT','LTHDT','MHH','HCSDL','HDH'],outputCol = 'features')
```

```
output = assembler.transform(df)
```

```
output.show(truncate = False)
```

```
+-------+----+----+---+----+------+----+-----+----+-----+---+------+----+-----+--------------------------------------+
|MSSV   |NMDT|CTRR|HTS|KTLT|CTDLGT|KTMT|LTHDT|MHH |HCSDL|HDH|NMTTNT|DHMT|MMANM|features                              |
+-------+----+----+---+----+------+----+-----+----+-----+---+------+----+-----+--------------------------------------+
|75321  |6.5 |6.5 |5.0|6.0 |5.5   |6.5 |7.5  |7.0 |8.0  |6.5|1     |0   |1    |[6.5,6.5,5.0,6.0,5.5,6.5,7.5,7.0,8.0,6.5]  |
|81657  |7.5 |7.0 |5.5|7.0 |7.0   |7.5 |7.5  |6.0 |7.5  |7.5|1     |0   |0    |[7.5,7.0,5.5,7.0,7.0,7.5,7.5,6.0,7.5,7.5]  |
|209312 |7.0 |6.5 |5.5|7.5 |4.5   |7.0 |0.0  |7.0 |6.0  |8.0|1     |1   |0    |[7.0,6.5,5.5,7.5,4.5,7.0,0.0,7.0,6.0,8.0]  |
|583833 |6.5 |7.0 |6.0|5.5 |6.5   |7.0 |8.0  |7.0 |6.0  |7.0|1     |1   |1    |[6.5,7.0,6.0,5.5,6.5,7.0,8.0,7.0,6.0,7.0]  |
|1509861|7.5 |5.0 |5.0|9.0 |3.5   |6.5 |6.0  |7.0 |7.5  |6.5|0     |0   |0    |[7.5,5.0,5.0,9.0,3.5,6.5,6.0,7.0,7.5,6.5]  |
|2047807|6.5 |6.5 |7.5|9.5 |8.5   |6.5 |8.5  |7.5 |8.5  |8.5|1     |1   |0    |[6.5,6.5,7.5,9.5,8.5,6.5,8.5,7.5,8.5,8.5]  |
|2223555|9.0 |8.0 |9.5|10.0|9.0   |8.0 |8.0  |10.0|8.0  |8.5|0     |0   |0    |[9.0,8.0,9.5,10.0,9.0,8.0,8.0,10.0,8.0,8.5]|
|2240989|8.5 |7.0 |7.0|7.0 |6.0   |8.0 |7.0  |6.5 |7.5  |8.0|0     |0   |0    |[8.5,7.0,7.0,7.0,6.0,8.0,7.0,6.5,7.5,8.0]  |
|2260878|7.5 |8.5 |8.5|10.0|8.5   |8.0 |8.0  |8.5 |0.0  |9.0|0     |0   |0    |[7.5,8.5,8.5,10.0,8.5,8.0,8.0,8.5,0.0,9.0] |
|2274996|5.5 |7.0 |7.0|7.5 |7.5   |6.0 |8.0  |7.0 |7.5  |7.5|1     |1   |1    |[5.5,7.0,7.0,7.5,7.5,6.0,8.0,7.0,7.5,7.5]  |
|2343848|8.0 |7.0 |6.0|7.5 |5.0   |7.0 |6.5  |6.5 |0.0  |5.5|0     |0   |0    |[8.0,7.0,6.0,7.5,5.0,7.0,6.5,6.5,0.0,5.5]  |
|2683782|7.5 |8.0 |6.5|9.0 |8.5   |7.5 |7.0  |8.5 |8.0  |7.0|0     |1   |0    |[7.5,8.0,6.5,9.0,8.5,7.5,7.0,8.5,8.0,7.0]  |
|3095962|7.0 |6.5 |6.0|6.5 |7.5   |6.0 |7.5  |5.5 |7.5  |6.5|1     |1   |0    |[7.0,6.5,6.0,6.5,7.5,6.0,7.5,5.5,7.5,6.5]  |
|3419145|7.5 |6.5 |5.0|6.0 |1.0   |5.0 |7.0  |6.0 |6.0  |7.0|0     |0   |0    |[7.5,6.5,5.0,6.0,1.0,5.0,7.0,6.0,6.0,7.0]  |
|3546806|8.5 |7.5 |7.0|7.0 |6.5   |7.0 |7.5  |7.0 |7.0  |7.0|0     |0   |0    |[8.5,7.5,7.0,7.0,6.5,7.0,7.5,7.0,7.0,7.0]  |
|4045426|8.5 |8.5 |6.0|8.0 |7.5   |8.5 |9.0  |8.5 |7.5  |8.5|0     |1   |0    |[8.5,8.5,6.0,8.0,7.5,8.5,9.0,8.5,7.5,8.5]  |
|4536867|8.0 |8.0 |9.0|7.5 |9.0   |9.5 |9.5  |9.5 |9.0  |9.5|0     |1   |1    |[8.0,8.0,9.0,7.5,9.0,9.5,9.5,9.5,9.0,9.5]  |
|4650139|8.5 |8.0 |8.5|10.0|9.5   |8.5 |8.5  |8.5 |8.5  |9.0|0     |0   |0    |[8.5,8.0,8.5,10.0,9.5,8.5,8.5,8.5,8.5,9.0] |
|5665575|5.5 |5.5 |5.0|6.0 |6.5   |6.0 |6.5  |6.5 |7.0  |8.0|1     |1   |0    |[5.5,5.5,5.0,6.0,6.5,6.0,6.5,6.5,7.0,8.0]  |
|5763862|7.0 |6.0 |8.0|6.0 |7.5   |7.0 |6.5  |8.5 |7.5  |8.5|0     |0   |1    |[7.0,6.0,8.0,6.0,7.5,7.0,6.5,8.5,7.5,8.5]  |
+-------+----+----+---+----+------+----+-----+----+-----+---+------+----+-----+--------------------------------------+
only showing top 20 rows
```

```
output.select('features','NMTTNT','DHMT','MMANM').show(truncate = False
)
```

```
+---------------------------------------------+------+----+-----+
|features                                     |NMTTNT|DHMT|MMANM|
+---------------------------------------------+------+----+-----+
|[6.5,6.5,5.0,6.0,5.5,6.5,7.5,7.0,8.0,6.5]    |1     |0   |1    |
|[7.5,7.0,5.5,7.0,7.0,7.5,7.5,6.0,7.5,7.5]    |1     |0   |0    |
|[7.0,6.5,5.5,7.5,4.5,7.0,0.0,7.0,6.0,8.0]    |1     |1   |0    |
|[6.5,7.0,6.0,5.5,6.5,7.0,8.0,7.0,6.0,7.0]    |1     |1   |1    |
|[7.5,5.0,5.0,9.0,3.5,6.5,6.0,7.0,7.5,6.5]    |0     |0   |0    |
|[6.5,6.5,7.5,9.5,8.5,6.5,8.5,7.5,8.5,8.5]    |1     |1   |0    |
|[9.0,8.0,9.5,10.0,9.0,8.0,8.0,10.0,8.0,8.5]  |0     |0   |0    |
|[8.5,7.0,7.0,7.0,6.0,8.0,7.0,6.5,7.5,8.0]    |0     |0   |0    |
|[7.5,8.5,8.5,10.0,8.5,8.0,8.0,8.5,0.0,9.0]   |0     |0   |0    |
|[5.5,7.0,7.0,7.5,7.5,6.0,8.0,7.0,7.5,7.5]    |1     |1   |1    |
|[8.0,7.0,6.0,7.5,5.0,7.0,6.5,6.5,0.0,5.5]    |0     |0   |0    |
|[7.5,8.0,6.5,9.0,8.5,7.5,7.0,8.5,8.0,7.0]    |0     |1   |0    |
|[7.0,6.5,6.0,6.5,7.5,6.0,7.5,5.5,7.5,6.5]    |1     |1   |0    |
|[7.5,6.5,5.0,6.0,1.0,5.0,7.0,6.0,6.0,7.0]    |0     |0   |0    |
|[8.5,7.5,7.0,7.0,6.5,7.0,7.5,7.0,7.0,7.0]    |0     |0   |0    |
|[8.5,8.5,6.0,8.0,7.5,8.5,9.0,8.5,7.5,8.5]    |0     |1   |0    |
|[8.0,8.0,9.0,7.5,9.0,9.5,9.5,9.5,9.0,9.5]    |0     |1   |1    |
|[8.5,8.0,8.5,10.0,9.5,8.5,8.5,8.5,8.5,9.0]   |0     |0   |0    |
|[5.5,5.5,5.0,6.0,6.5,6.0,6.5,6.5,7.0,8.0]    |1     |1   |0    |
|[7.0,6.0,8.0,6.0,7.5,7.0,6.5,8.5,7.5,8.5]    |0     |0   |1    |
+---------------------------------------------+------+----+-----+
only showing top 20 rows
```

Ta sẽ xây dựng 3 cây quyết định cho 3 môn tự chọn dựa trên vector các cột điểm bắt buộc:

```
model_df = output.select('features', 'NMTTNT', 'DHMT', 'MMANM')
```

```
-----------------------------------------------------------------------
train_df, test_df = model_df.randomSplit([0.8,0.2])

-----------------------------------------------------------------------
train_df.count()

287

-----------------------------------------------------------------------
test_df.count()

72
-----------------------------------------------------------------------
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

-----------------------------------------------------------------------
def DeciTree(tenmon):
    df_classifier = DecisionTreeClassifier(labelCol=tenmon).fit(train_d
f)
    df_predict = df_classifier.transform(test_df)
    df_predict.show()
    df_accuracy = MulticlassClassificationEvaluator(labelCol=tenmon,
                                metricName='accuracy').evaluate(df_predict)
    print("Accuracy: ", df_accuracy)
    df_precision = MulticlassClassificationEvaluator(labelCol='NMTTNT',
                 metricName='weightedPrecision').evaluate(df_predict)
    print("Precision: ", df_precision)
    return df_classifier


-----------------------------------------------------------------------
DeciTree('NMTTNT').featureImportances
```

```
+--------------------+------+----+-----+-------------+--------------------+----------+
|            features|NMTTNT|DHMT|MMANM|rawPrediction|         probability|prediction|
+--------------------+------+----+-----+-------------+--------------------+----------+
|[5.0,7.5,5.5,6.0,...|     1|   1|    1|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[5.5,5.5,5.0,6.0,...|     1|   1|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[5.5,6.0,7.0,7.0,...|     1|   1|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.0,6.0,5.5,6.0,...|     0|   0|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.0,6.0,5.5,8.0,...|     1|   1|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.0,6.5,1.0,5.5,...|     0|   0|    0|    [3.0,0.0]|           [1.0,0.0]|       0.0|
|[6.0,7.0,5.5,5.5,...|     1|   0|    1|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.0,7.0,6.0,6.5,...|     0|   0|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.0,7.5,5.0,6.0,...|     1|   1|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.5,6.0,5.0,6.5,...|     1|   1|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.5,6.0,6.0,7.0,...|     1|   1|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.5,6.0,6.5,7.5,...|     1|   1|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.5,6.0,7.5,7.5,...|     1|   1|    1|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.5,6.5,5.0,6.0,...|     1|   0|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.5,7.5,6.5,6.5,...|     1|   1|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[6.5,8.0,6.0,7.5,...|     1|   1|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[7.0,5.0,7.0,7.5,...|     0|   0|    0|    [5.0,1.0]|[0.83333333333333...|       0.0|
|[7.0,5.5,6.5,9.0,...|     0|   0|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
|[7.0,6.5,2.5,8.0,...|     0|   0|    0|    [0.0,2.0]|           [0.0,1.0]|       1.0|
|[7.0,6.5,5.0,5.0,...|     1|   1|    0|  [10.0,69.0]|[0.12658227848101...|       1.0|
+--------------------+------+----+-----+-------------+--------------------+----------+
only showing top 20 rows

Accuracy:  0.75
Precision:  0.7515873015873017
SparseVector(10, {0: 0.4082, 1: 0.14, 2: 0.043, 3: 0.0351, 4: 0.0563, 5: 0.0872, 6: 0.0139, 7: 0.0858, 9: 0.1305})
```

```
DeciTree('DHMT').featureImportances
```

```
+--------------------+------+----+-----+-------------+--------------------+----------+
|            features|NMTTNT|DHMT|MMANM|rawPrediction|         probability|prediction|
+--------------------+------+----+-----+-------------+--------------------+----------+
|[5.0,7.5,5.5,6.0,...|     1|   1|    1|    [5.0,0.0]|           [1.0,0.0]|       0.0|
|[5.5,5.5,5.0,6.0,...|     1|   1|    0|    [0.0,3.0]|           [0.0,1.0]|       1.0|
|[5.5,6.0,7.0,7.0,...|     1|   1|    0|   [9.0,24.0]|[0.27272727272727...|       1.0|
|[6.0,6.0,5.5,6.0,...|     0|   0|    0|   [9.0,24.0]|[0.27272727272727...|       1.0|
|[6.0,6.0,5.5,8.0,...|     1|   1|    0|    [0.0,1.0]|           [0.0,1.0]|       1.0|
|[6.0,6.5,1.0,5.5,...|     0|   0|    0|   [11.0,3.0]|[0.78571428571428...|       0.0|
|[6.0,7.0,5.5,5.5,...|     1|   0|    1|   [9.0,24.0]|[0.27272727272727...|       1.0|
|[6.0,7.0,6.0,6.5,...|     0|   0|    0|   [0.0,16.0]|           [0.0,1.0]|       1.0|
|[6.0,7.5,5.0,6.0,...|     1|   1|    0|   [11.0,3.0]|[0.78571428571428...|       0.0|
|[6.5,6.0,5.0,6.5,...|     1|   1|    0|   [11.0,3.0]|[0.78571428571428...|       0.0|
|[6.5,6.0,6.0,7.0,...|     1|   1|    0|   [9.0,24.0]|[0.27272727272727...|       1.0|
|[6.5,6.0,6.5,7.5,...|     1|   1|    0|    [1.0,0.0]|           [1.0,0.0]|       0.0|
|[6.5,6.0,7.5,7.5,...|     1|   1|    1|   [9.0,24.0]|[0.27272727272727...|       1.0|
|[6.5,6.5,5.0,6.0,...|     1|   0|    0|   [11.0,3.0]|[0.78571428571428...|       0.0|
|[6.5,7.5,6.5,6.5,...|     1|   1|    0|   [0.0,16.0]|           [0.0,1.0]|       1.0|
|[6.5,8.0,6.0,7.5,...|     1|   1|    0|   [0.0,16.0]|           [0.0,1.0]|       1.0|
|[7.0,5.0,7.0,7.5,...|     0|   0|    0|    [1.0,0.0]|           [1.0,0.0]|       0.0|
|[7.0,5.5,6.5,9.0,...|     0|   0|    0|    [1.0,0.0]|           [1.0,0.0]|       0.0|
|[7.0,6.5,2.5,8.0,...|     0|   0|    0|   [11.0,0.0]|           [1.0,0.0]|       0.0|
|[7.0,6.5,5.0,5.0,...|     1|   1|    0|   [11.0,3.0]|[0.78571428571428...|       0.0|
+--------------------+------+----+-----+-------------+--------------------+----------+
only showing top 20 rows

Accuracy:  0.7222222222222222
Precision:  0.7247474747474747
SparseVector(10, {0: 0.281, 1: 0.0563, 2: 0.1813, 3: 0.0956, 5: 0.1424, 6: 0.1706, 9: 0.0728})
```

```
DeciTree('MMANM').featureImportances
```

```
+--------------------+------+----+-----+-------------+--------------------+----------+
|            features|NMTTNT|DHMT|MMANM|rawPrediction|         probability|prediction|
+--------------------+------+----+-----+-------------+--------------------+----------+
|[5.0,7.5,5.5,6.0,...|     1|   1|    1|    [0.0,2.0]|           [0.0,1.0]|       1.0|
|[5.5,5.5,5.0,6.0,...|     1|   1|    0|    [0.0,2.0]|           [0.0,1.0]|       1.0|
|[5.5,6.0,7.0,7.0,...|     1|   1|    0|    [0.0,2.0]|           [0.0,1.0]|       1.0|
|[6.0,6.0,5.5,6.0,...|     0|   0|    0|    [2.0,4.0]|[0.33333333333333...|       1.0|
|[6.0,6.0,5.5,8.0,...|     1|   1|    0|    [8.0,0.0]|           [1.0,0.0]|       0.0|
|[6.0,6.5,1.0,5.5,...|     0|   0|    0|   [11.0,0.0]|           [1.0,0.0]|       0.0|
|[6.0,7.0,5.5,5.5,...|     1|   0|    1|    [8.0,0.0]|           [1.0,0.0]|       0.0|
|[6.0,7.0,6.0,6.5,...|     0|   0|    0|    [2.0,0.0]|           [1.0,0.0]|       0.0|
|[6.0,7.5,5.0,6.0,...|     1|   1|    0|    [8.0,0.0]|           [1.0,0.0]|       0.0|
|[6.5,6.0,5.0,6.5,...|     1|   1|    0|  [26.0,21.0]|[0.55319148936170...|       0.0|
|[6.5,6.0,6.0,7.0,...|     1|   1|    0|  [26.0,21.0]|[0.55319148936170...|       0.0|
|[6.5,6.0,6.5,7.5,...|     1|   1|    0|  [26.0,21.0]|[0.55319148936170...|       0.0|
|[6.5,6.0,7.5,7.5,...|     1|   1|    1|  [26.0,21.0]|[0.55319148936170...|       0.0|
|[6.5,6.5,5.0,6.0,...|     1|   0|    0|  [26.0,21.0]|[0.55319148936170...|       0.0|
|[6.5,7.5,6.5,6.5,...|     1|   1|    0|  [26.0,21.0]|[0.55319148936170...|       0.0|
|[6.5,8.0,6.0,7.5,...|     1|   1|    0|  [26.0,21.0]|[0.55319148936170...|       0.0|
|[7.0,5.0,7.0,7.5,...|     0|   0|    0|  [26.0,21.0]|[0.55319148936170...|       0.0|
|[7.0,5.5,6.5,9.0,...|     0|   0|    0|  [26.0,21.0]|[0.55319148936170...|       0.0|
|[7.0,6.5,2.5,8.0,...|     0|   0|    0|   [11.0,0.0]|           [1.0,0.0]|       0.0|
|[7.0,6.5,5.0,5.0,...|     1|   1|    0|    [3.0,3.0]|           [0.5,0.5]|       0.0|
+--------------------+------+----+-----+-------------+--------------------+----------+
only showing top 20 rows

Accuracy:  0.7638888888888888
Precision:  0.5669154228855722
SparseVector(10, {0: 0.2868, 1: 0.04, 2: 0.0135, 4: 0.3274, 5: 0.2102, 7: 0.1222})
```

Độ hiệu quả của cây quyết định được đánh giá dựa trên 2 chỉ số accuracy và precision. Nhìn chung, 3 cây quyết định vừa xây ở trên có độ chính xác ở mức tạm được, một phần là vì tập dữ liệu nhỏ, chưa đủ tổng quát. Dưới đây là hàm phụ để trợ giúp vẽ cây dữ liệu. Hình ảnh các cây dữ liệu sẽ được đính kèm báo cáo.

```
def parse_debug_string_lines(lines):

    block = []
    while lines:


        if lines[0].startswith('If'):
            bl = ' '.join(lines.pop(0).split()[1:]).replace('(', '').re
place(')', '')
            block.append({'name': bl, 'children': parse_debug_string_li
nes(lines)})


            if lines[0].startswith('Else'):
                be = ' '.join(lines.pop(0).split()[1:]).replace('(', ''
).replace(')', '')
                block.append({'name': be, 'children': parse_debug_strin
g_lines(lines)})
        elif not lines[0].startswith(('If', 'Else')):
```

```
            block2 = lines.pop(0)
            block.append({'name': block2})
        else:
            break

    return block


def debug_str_to_json(debug_string):
    data = []
    for line in debug_string.splitlines():
        if line.strip():
            line = line.strip()
            data.append(line)
        else:
            break
        if not line: break

    json = {'name': 'Root', 'children': parse_debug_string_lines(data[1
:])}


    return json
```

Tài liệu tham khảo

[1] “What is Apache Spark?” SearchDataManagement. [Online]. Available: https://www.techtarget.com/searchdatamanagement/definition/Apache-Spark. [Accessed: 09-April-2022].

[2] “Tìm hiểu về Apache Spark” Phu Ngoc Nguyen. [Online]. Available: https://viblo.asia/p/tim-hieu-ve-apache-spark-ByEZkQQW5Q0. [Accessed: 09-April-2022].

[3] “Tổng quan về Apache Spark cho hệ thông Big Data” Hoang Trong Hieu. [Online]. Available: https://viblo.asia/p/tong-quan-ve-apache-spark-cho-he-thong-big-data-RQqKLxR6K7z. [Accessed: 09-April-2022].

[4] “Apache Spark Officially Sets a New Record in Large-Scale Sorting” Reynold Xin. [Online]. Available https://databricks.com/blog/2014/11/05/spark-officially-sets-a-new-record-in-large-scale-sorting.html. [Accessed: 09-April-2022].

[5], [6] “Apache Spark” Databricks.com. [Online]. Available: https://databricks.com/spark/about. [Accessed: 09-April-2022].

[7] “Apache Spark Ecosystem – Complete Spark Components Guide” data-flair.training blog. [Online]. Available: https://data-flair.training/blogs/apache-spark-ecosystem-components/ [Accessed: 09-April-2022].

[8] “Spark Streaming Programming Guide” Apache Spark. [Online]. Available: https://spark.apache.org/docs/latest/streaming-programming-guide.html#spark-streaming-programming-guide [Accessed: 09-April-2022].

[9] Apache Spark Fundamentals. [Online]. Available: https://techmaster.vn/posts?keyword=Apache+Spark+Fundamentals&category=&author [Accessed: 09-April-2022].

[10] Apache Spark Key Terms, Explained. [Online]. Available: https://databricks.com/blog/2016/06/22/apache-spark-key-terms-explained.html [Accessed: 09-April-2022].

[11] Spark – RDD. [Online]. Available: https://www.tutorialspoint.com/spark_sql/spark_rdd.htm [Accessed: 09-April-2022].

[12] Spark Overview. [Online]. Available: https://spark.apache.org/docs/2.3.0/index.html [Accessed: 09-April-2022].

[13] Jean-Georges Perrin. Spark in Action 2 edition. Manning Publication 2020.