

**Student Academic Success: A Predictive Analytics Approach**

**Using Kafka, Hadoop, Spark, and Jupyter Notebooks**

Big Data

2023-2024

Master of Engineering in Data Science

Institute of Technology of Cambodia

Group members:

1. POV Phannet (ID: M080201 )
2. SAM Lyheng (ID: M080203 )
3. LUK Chamnan (ID: M080213)

Submission Date: 4 February, 2024

Lectures: Mr. CHAN Sophal

Big Data Project Report

Title: Student Academic Success: A Predictive Analytics Approach Using Kafka, Hadoop, Spark, and Jupyter Notebooks.

# **Introduction**

This transformative project in the realm of education is dedicated to advancing student academic success through the strategic application of predictive analytics. Integrating Kafka, Hadoop, and Spark, the initiative seeks to revolutionize educational practices by identifying patterns, anticipating trends, and fostering a personalized learning environment. The significance lies in the potential of predictive analytics to offer early identification of at-risk students, personalize learning experiences, and optimize the allocation of academic support resources. Kafka ensures real-time data ingestion, Hadoop provides a scalable storage infrastructure, and Spark, as the dynamic computing engine, leads advanced machine learning efforts. The project anticipates outcomes such as improved student success rates, empowered educators, and the establishment of a data-driven culture within the educational institution. Ultimately, this endeavor aspires to redefine the educational narrative, creating a more personalized, responsive, and successful academic experience for all students.

# **Project Scope**

## **Overview**

The project, titled "Student Academic Success: A Predictive Analytics Approach Using Kafka, Hadoop, Spark, and Jupyter Notebooks," represents a pioneering initiative aimed at transforming the support system for students through the application of predictive analytics. By seamlessly integrating cutting-edge technologies such as Apache Kafka, Hadoop, Spark, and Jupyter Notebooks, the project seeks to redefine conventional approaches, providing actionable insights and optimizing user experiences within the context of student academic success. Rooted in a commitment to enhancing student success, the project leverages Apache Kafka for real-time data streaming, ensuring a continuous flow of critical information that is integral to understanding student behaviors, trends, and challenges. This comprehensive approach reflects a dedication to harnessing the wealth of data within the academic environment to drive positive and impactful outcomes for student academic success.

## **Objectives:**

## Real-Time Academic Data Ingestion Using Kafka:

## Implement Kafka as the distributed streaming platform for the real-time ingestion of diverse student academic data.

## Ensure a continuous and efficient data flow, laying the foundation for up-to-the-minute analytics on student performance, engagement, and success factors.

## Specify the types of data you plan to ingest in real-time

## Clarify the desired frequency of data ingestion for a more precise understanding of real-time requirements.

## Comprehensive Academic Analysis within Hadoop Ecosystem:

## Leverage the Hadoop Distributed File System (HDFS) for scalable and reliable storage of academic datasets.

## Conduct thorough analysis within the Hadoop ecosystem, exploring patterns, correlations, and concealed insights in student academic records.

## Define specific academic datasets that will be analyzed

* + Highlight the types of patterns and correlations you aim to uncover in student academic records.
* Predictive Modeling for Student Success Analysis:
  + Utilize Apache Spark's advanced machine learning capabilities to craft predictive models.
  + Analyze historical academic data to discern and forecast trends in student performance, identifying key factors influencing academic success.
  + Provide details on the specific machine learning algorithms or techniques you plan to employ with Apache Spark.
  + Clarify the scope of forecasting trends, such as predicting dropout rates or identifying factors influencing academic success.
* Interactive Data Exploration and Visualization using Jupyter Notebooks:
  + Integrate Jupyter Notebooks for collaborative and interactive data exploration and visualization.
  + Enhance the interpretability of key academic metrics, trends, and patterns, fostering a collaborative approach to data-driven decision-making in the context of student academic success.
  + Specify the key academic metrics you intend to explore interactively.
  + Provide examples of the types of visualizations that will be created to enhance interpretability.

# **Framework Development**

## 

## 1. Tools and Technologies:

1. **Apache Kafka:**

* Real-time Data Ingestion: Apache Kafka serves as a pivotal tool for the real-time ingestion of a diverse range of student academic data. Its robust architecture ensures an uninterrupted and efficient flow of information, laying the foundation for up-to-the-minute analytics on student performance, engagement, and success factors. Kafka's fault-tolerant mechanisms further enhance the reliability and seamlessness of data streaming.

1. **Hadoop Distributed File System (HDFS):**

* Scalable Storage Solution: HDFS is employed as the storage layer to accommodate the vast and varied nature of academic datasets. Its scalable architecture allows for the reliable storage of extensive records related to student performance and academic activities. The use of HDFS ensures data integrity and supports efficient analysis within the broader Hadoop ecosystem, enabling a comprehensive understanding of academic patterns and trends.

1. **Apache Spark:**

* Advanced Machine Learning Capabilities: Apache Spark is harnessed for its advanced machine learning capabilities, enabling the development of predictive models for student success analysis. As a dynamic and high-performance computing engine, Spark plays a crucial role in conducting in-depth analyses of academic data. It identifies patterns and influential factors contributing to student outcomes, providing actionable insights based on historical academic data.

1. **Jupyter Notebooks:**

* Interactive Exploration and Visualization: Jupyter Notebooks are seamlessly integrated into the workflow for interactive exploration and visualization of student academic data. This tool enhances the collaborative interpretation of academic metrics, trends, and patterns. By providing a platform for stakeholders to interactively explore key data points, Jupyter Notebooks foster a collaborative and informed decision-making process, empowering stakeholders in their efforts to improve student academic success.

## 2. Components

1. **Data Ingestion Layer:**

* **Tool:** Apache Kafka
* **Functionality:**
  + Real-Time Academic Data Ingestion: Apache Kafka is employed for the real-time ingestion of diverse student academic data, ensuring a continuous flow of information.
  + Fault-Tolerant Configuration: Kafka is configured to establish fault-tolerant mechanisms, ensuring the reliability and continuous streaming of critical academic data.

1. **Storage Layer:**

* **Tool:** Hadoop Distributed File System (HDFS)
* **Functionality:**
  + Scalable Storage Solution: HDFS serves as the storage layer, providing scalable and reliable storage for large academic datasets.
  + Efficient Storage Mechanisms: HDFS efficiently stores data across multiple nodes, enhancing scalability for the storage of extensive student academic records.

1. **Interactive Analysis and Visualization:**

* **Tool:** Apache Spark
* **Functionality:**
  + Advanced Machine Learning Capabilities: Apache Spark leverages advanced machine learning capabilities for predictive modeling, enabling the exploration of patterns and influential factors in student academic data.
  + In-Depth Analysis: Spark acts as a dynamic and high-performance computing engine, facilitating in-depth analysis of historical academic data to derive actionable insights for improving student outcomes.

1. **Analytics and Predictive Modeling Layer:**

* **Tool:** Jupyter Notebooks
* **Functionality:**
  + Interactive Exploration and Analysis: Jupyter Notebooks are utilized for interactive exploration and analysis of student academic data, providing a collaborative environment for stakeholders.
  + Visualization for Interpretability: Jupyter Notebooks enable the visualization of key academic metrics, trends, and patterns, enhancing interpretability and supporting data-driven decision-making for student academic success initiatives.

# **IV. Implementation**

In the pursuit of advancing student academic success, this implementation leverages state-of-the-art technologies to create a dynamic and efficient data processing ecosystem. The integration of Apache Kafka, Docker, Hadoop, and Spark represents a groundbreaking approach, emphasizing real-time data streaming, containerized deployment, and scalable analytics. This concise introduction sets the stage for a comprehensive exploration of how these technologies seamlessly converge to lay the groundwork for transformative insights and data-driven decision-making within the educational domain.

1. **Local Kafka Setup:**
   * Start by setting up a local Kafka instance for real-time data streaming. This involves downloading and configuring Kafka locally.
   * Create a specific Kafka topic where the real-time data will be ingested.
   * Implement a dedicated Python script (producer.py) that serves as a Kafka producer to inject data into the specified Kafka topic.
2. **Dockerized Hadoop and Spark:**
   * Containerize Hadoop and Spark services using Docker to ensure portability and reproducibility.
   * Create a docker-compose.yml file that orchestrates the deployment of Hadoop and Spark services in distinct containers. Define the necessary configurations for each service, including ports, volumes, and dependencies.
3. **Separation for Scalability and Ease of Management:**
   * Utilize Docker's containerization to separate Hadoop and Spark components, ensuring a modular and organized deployment.
   * Leverage separate containers for Hadoop and Spark to enhance scalability and ease of management. This separation allows for efficient resource allocation and individual scaling of components.
4. **Flume as Data Transfer Intermediary:**
   * Containerize Flume operates within its own Docker container, serving as the intermediary for data transfer between Kafka and HDFS.
   * Configure Flume to facilitate the seamless transfer of data from Kafka to HDFS, ensuring a reliable and efficient data pipeline.
5. **Modular Setup for Flexibility and Scalability:**
   * Emphasize the modular setup of each component within its containerized environment. This approach enhances flexibility, making it easier to update, replace, or scale individual components without affecting the entire system.
   * Ensure that each containerized service can operate independently, promoting encapsulation and reducing potential conflicts between different components.

# **V. Result**

Upon implementing the Spark framework to load data from the Hadoop Distributed File System (HDFS), the project entered a transformative phase in data processing. The data preprocessing stage, executed within the Spark environment, played a pivotal role in cleansing and structuring academic data for further analysis. Subsequent examinations of correlations within the dataset unearthed valuable insights into relationships between various academic variables. The strategic removal of unnecessary tables optimized the dataset, streamlining subsequent analyses.

The pivotal phase involved training a machine learning model, specifically employing the Random Forest algorithm. This process revealed the predictive potential of the model in forecasting student academic outcomes. The results of the Random Forest training provided a nuanced understanding of the factors influencing academic success, aligning with the overarching goal of the project—to enhance student success through predictive analytics.

The culmination of these results underscores the efficacy of employing Spark within a Hadoop ecosystem for intricate data processing, paving the way for informed decision-making and proactive interventions to support student academic success. The achieved accuracy of **0.7391** in the Random Forest model training phase signifies commendable predictive performance, reflecting the model's ability to make correct predictions. While accuracy serves as a foundational indicator, ongoing analysis and model refinement will consider additional metrics for a more comprehensive evaluation, reinforcing the positive stride made in leveraging machine learning within the academic context. This showcases the potential of predictive analytics to meaningfully contribute to student success initiatives.

# **Conclusion**

# In conclusion, the project aims to reshape e-commerce by leveraging predictive analytics with Kafka, Hadoop, Spark, and Jupyter Notebooks. Anticipated outcomes include a scalable big data framework for efficient processing, improved data quality, and readiness for sophisticated analyses. The integration of predictive models and real-time data ingestion ensures agile responses to market changes, while collaborative exploration and visualization tools foster a culture of informed decision-making. Overall, the project positions the organization for data-driven excellence, promising actionable insights, and sustained innovation in the dynamic e-commerce landscape.

## **Limitation**

* **Dependency on Confluent for Kafka Connect:**
  + A notable limitation arises from the dependency on Confluent for Kafka Connect, a tool that facilitates data integration between Kafka and external systems. The use of Kafka Connect typically requires the adoption of Confluent, which is a third-party tool with associated costs. This introduces a financial consideration that may impact the project's scalability and accessibility, potentially limiting its widespread adoption.
* **Limited Variety in Machine Learning Models:**
  + The choice to exclusively train the model with Random Forest in the SparkML format represents a limitation in terms of model variety. While Random Forest is a robust algorithm, the decision not to explore and incorporate additional machine learning models might restrict the ability to capture nuances in the data that could be better addressed by different algorithms. The limitation in model variety might impact the project's adaptability to diverse academic datasets and scenarios, potentially missing out on the strengths of other machine learning approaches.
* **Distributed Environment Setup**
  + In the current project setup, there is a limitation arising from the use of Docker containers for Spark, Hadoop, and Kafka. Due to the isolated nature of these containers and the dependency hierarchy, we are required to execute commands in a sequential manner. The workflow mandates initiating the Kafka container locally, followed by the Hadoop container, and finally, the Spark container. This step-by-step execution is necessary to ensure the proper initialization and coordination between the different components. While this sequential approach facilitates a successful deployment, it introduces a constraint on the overall efficiency of the workflow. Future optimizations or adjustments to the deployment strategy could potentially enhance the parallelization of these processes, minimizing the need for strict sequential execution.

# 

# 

# 

# 