**IPL Data Analysis | Apache Spark**

We will proceed as per the below steps, one by one:

**1. Prepare your Data:**

* **Acquire the Data:** The project will use the IPL (Indian Premier League) cricket data set.
* **Upload the Data to Cloud Storage:** The project will fetch the data from Amazon S3, a cloud storage service. There are other cloud storage options as well, depending on preference.

**2. Set Up the Development Environment:**

* **Use Apache Spark:** The project will utilize Apache Spark for data processing. We can configure Apache Spark on the development machine or use a cloud-based Spark environment.

**3. Data Processing with Apache Spark:**

* **Write PySpark Code:** The project will highlight using PySpark, a Python API for Apache Spark. Code is written to manipulate and transform data according to the project's objectives.

**4. Data Analysis:**

* **Use SQL for Analysis:** The project demonstrates using SQL queries to analyze the transformed data.
* **Data Visualization:** The project demonstrates creating data visualizations to understand the data better. Tools like Tableau or Power BI can be used to create charts and graphs as well.

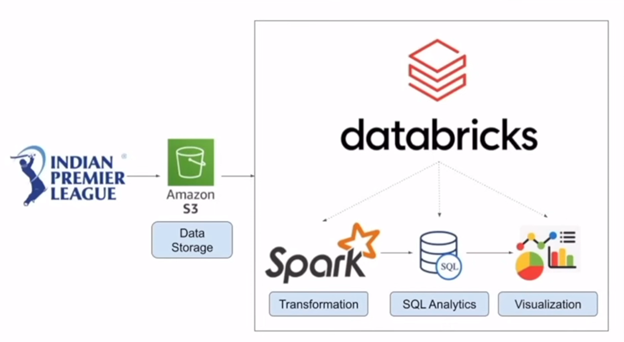
**5. Find Insights and Make Decisions:**

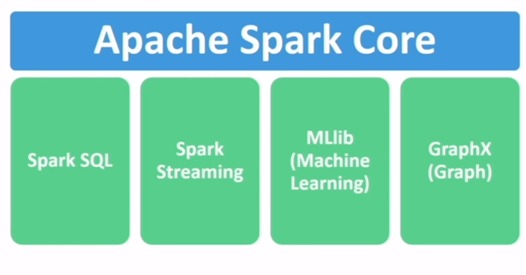
* **Answer Questions:** Ultimately, the goal is to use the processed and analyzed data to answer specific questions and gain insights relevant to the project.
* **Make Decisions:** Based on the extracted knowledge from the data, we can make informed decisions related to the project goals.

**Additional Notes:**

* The project provides a high-level overview. Each step has the potential to involve further research or breakdown into more detailed sub-sateps depending on the specific project requirements and chosen tools.
* The projects portrays deeper understanding of Apache Spark, PySpark coding, and data visualization tools.

**Project Architecture Diagram**





Apache Spark Core: Heart of Apache. Responsible for executing every code submitted to Spark Application.

Spark SQL : Supports SQL queries inside Spark application.

Spark Streaming: to Handle and process the real time data. Like the ones we see on google maps or uber.

MLlib (Machine Learning) : supports running machine learning applications on large data.

GraphX (Graph) : Supports processing the graphical data space inside Spark Application.

**Apache Spark** is a unified computing engine and a set of libraries for parallel data processing on computer clusters. It’s a framework that divides your data in multiple partitions and processes that data on to the large clusters of machines. It will combine the output at the end again.

**APACHE SPARK**

**A diagram of a software development process

Description automatically generated with medium confidence**

**SPARK ARCHITECTURE**

Diagram of a computer process

Description automatically generated

**Driver Process**

It’s the manager. The driver process runs your main() function, sits on a node in the cluster, and is responsible for three things:

* Maintaining information about the Spark Application
* Responding to a user's program or input
* Analyzing, distributing, and scheduling work across the executors

**Executor Processes**

The executors are responsible for carrying out the work that the driver assigns them.

* Executing code assigned to it by the driver.
* Reporting the state of the computation on that executor back to the driver node.

Spark, in addition to its cluster mode, also has a local mode. Which means that they can live on the same machine or different machines. In local mode, the driver and executors run (as threads) on your individual computer instead of a cluster.

**TheSparkSession:**  You control your Spark Application through a driver process called the SparkSession.

The SparkSession instance is the way Spark executes user-defined manipulations across the cluster. There is a one-to-one correspondence between a SparkSession and a Spark Application.

In Scala and Python, the variable is available as **spark** when you start the console.

**Spark DataFrame:**

Its similar to spreadsheets. A DataFrame is the most common Structured API and simply represents a table of data with rows and column.

In Spark all this data is stored distributed fashion. Fragments of data can be on different machines.

**Tranformations :** These are instructions used to specify the modifications to Spark about how we want to change the ‘DataFrame’.

In Spark, core data structures are immutable. They cannot be changed after they are created.

We will have to instruct.

* Spark keeps on stacking all the transformation logic(like filter, group.. etc) until there is an action (eg : show) demanded. Once we put an action to it, it executes all the transformations at once.

**DATABRICKS** is a software that support Apache Spark. It takes care of all the infrastructural component and just lets us work on the business side.