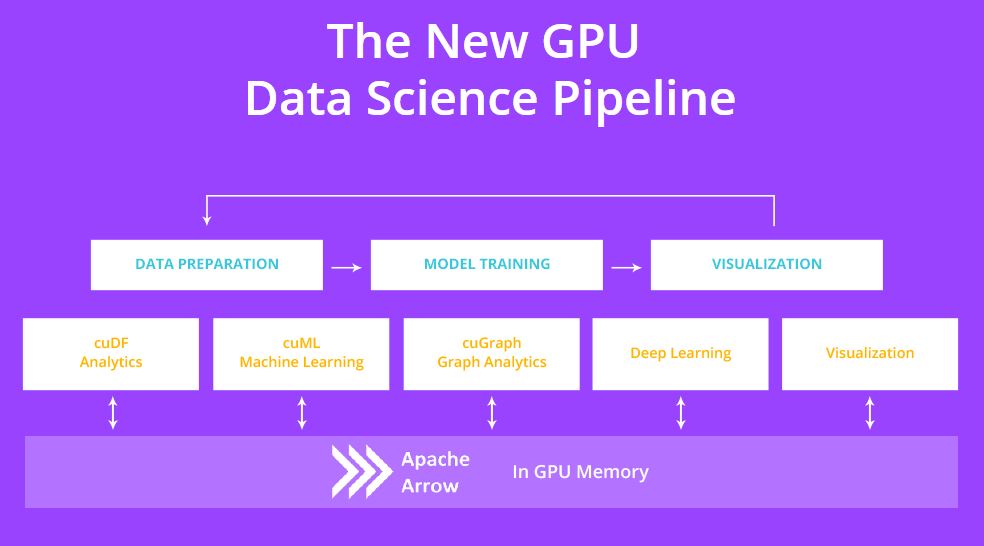
**Open Data Science**

**Cuda Kernals:-** In **CUDA**, the host refers to the CPU and its memory, while the device refers to the GPU and its memory. Code run on the host can manage memory on both the host and device, and also launches **kernels** which are functions executed on the device. These **kernels** are executed by many GPU threads in parallel.

**TPU:-** A tensor processing unit is an AI accelerator application-specific integrated circuit developed by Google specifically for neural network machine learning.

**GPU:-** A graphics processing unit is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. GPUs are used in embedded systems, mobile phones, personal computers, workstations, and game consoles.

**MKL:-** Intel Math Kernel Library is a library of optimized math routines for science, engineering, and financial applications. Core math functions include BLAS, LAPACK, ScaLAPACK, sparse solvers, fast Fourier transforms, and vector math. The routines in MKL are hand-optimized specifically for Intel processors.

**HPC:- High-performance computing** (**HPC**) is the use of parallel processing for running advanced application programs efficiently, reliably and quickly. The term applies especially to systems that function above a teraflop or 1012 floating-point operations per second.

End-to-end Computation on the GPU with the GPU Data Frame (GDF)

One of the things we are most excited about as a [newly open source company](https://www.mapd.com/blog/2017/05/08/mapd-open-sources-gpu-powered-database/) is the potential to help kickstart a larger ecosystem of GPU computing. This is why we are particularly excited about our work with [Continuum Analytics](https://www.continuum.io/) and [H2O.ai](https://www.h2o.ai/) to found the [GPU Open Analytics Initiative (GOAI)](http://gpuopenanalytics.com/) and its first project, the [GPU Data Frame (GDF)](https://github.com/gpuopenanalytics/pygdf), as our first step toward an open ecosystem of end-to-end GPU computing.

A revolution is occurring across the GPU software stack, driven by the disruptive performance gains GPUs have seen generation after generation. The modern field of deep learning would have not been possible without GPUs (with special credit due to [Nvidia](http://www.nvidia.com/) for innovating both on the hardware and software side), and as a database we’re often seeing two-or-more orders of magnitude performance gains compared to CPU systems.

But for all of the innovation occurring in the GPU software ecosystem, the systems and platforms themselves still remain isolated from each other. Even though the individual components are seeing significant acceleration from running on the GPU, they must intercommunicate over the relatively thin straw of the PCIe (or via faster [NVLink](http://www.nvidia.com/object/nvlink.html) on certain IBM power systems) and then through CPU memory.

For example, until the advent of the GDF project, to take the results of a SQL query in MapD and feed them into a regression algorithm in H20.ai, the following would need to occur:

1) An external process (client) requests that a query is executed in the MapD Core database.   
2) The query is executed in MapD Core on the GPU(s). If the data had previously been queried it should already be in GPU RAM.   
3) MapD Core then copies the results across the PCIe bus to CPU memory.   
4) MapD Core then serializes the results and sends it to the client over a network socket via Thrift, JDBC or ODBC.   
5) The client then takes the query results and puts them in a format usable by H2O.ai.   
6) The H2O.ai framework then copies the input to the GPU.   
7) The H20.ai framework the trains a model on the GPU.   
8) The trained model parameters are then copied back to CPU and returned to the client/

As one can imagine, this state of affairs, requiring not only repeated hops from GPU-to-CPU and back, but also the transmission of data across the network (even on a single server), is extremely inefficient. While the aggregate memory bandwidth across a server full of Nvidia Pascal GPUs can approach 6 terabytes per second, the real-world aggregate bandwidth between GPUs and CPUs on a two-socket server is unlikely to exceed 40 gigabytes per second, or 150X slower than the intra-GPU bandwidth available.

This is the situation that the first initiative of GOAI, the GPU Data Frame (GDF), is designed to address. The principal goal of the GDF is to enable efficient intra-GPU communication between different processes running on the GPUs. For example, the GDF allows a process running on the GPU to seamlessly transfer data to another process without copying the data to the CPU. Even more, since the GDF leverages IPC functionality in the Nvidia CUDA programming API, the processes can just pass a handle to the data instead of copying the data itself, meaning that transfers are virtually without overhead. The net result is that the GPU becomes a first class compute citizen and processes can inter-communicate just as easily as processes running on the CPU.

Although the GDF is still a project in beta, the members of the initiative have an aggressive pipeline to make it production-ready by the time of Strata NY in September this year. On our end we are working to enable the GDF to handle the transfer of string and categorical data, to work across multiple GPUs, and to allow enabling bidirectional data flow from the GDF in and out of MapD (currently we only support output to the data format).

I’d also be remiss to not mention that the specification for the data format is based on [Apache Arrow](https://arrow.apache.org/), which until this point has been focused on CPU systems. We look forward to collaboration with the Arrow committers and potentially to the idea of some of the functionality of the GDF becoming part of the core Arrow spec.

It is easy to imagine some of the exciting applications that the GDF will enable. Seamless workflows that combine data processing, machine learning (ML), and visualization will be possible, without ever needing to leave the GPU. Users will be able to build a function with Continuum’s [Numba](http://numba.pydata.org/) to cluster or perform Principal Components Analysis (PCA) on a query result from MapD, or alternatively with a bit of glue code could push the results of a query directly into a deep learning framework like [TensorFlow](https://www.tensorflow.org/), [Theano](https://github.com/Theano/Theano) or [Torch](http://torch.ch/).

We look forward to working with the developers to continue to build out both the GDF as well as embark on other projects under the umbrella of GOAI, with the goal of enabling a larger and more cohesive GPU ecosystem. Please join us on the newly created [Google Groups](https://groups.google.com/forum/#!forum/gpuopenanalytics) for the initiative, we’d love to hear your feedback and thoughts!

**Big Data Analytics**

**Historical Insights:- Historical** analytics refers to the analysis of activity and **data** from the past to discern particular trends, patterns, correlations, and other statistical relationships that may drive **insight** into business performance.

**Prescriptive Analytics**:- **Prescriptive analytics** is comparatively a new field in **data** science. It goes even a step further than descriptive and **predictive analytics**. **Prescriptive analytics** showcases viable solutions to a problem and the impact of considering a solution on future trend. It is considered the aim of any **data analysis** project.

**Perishable Insights:-** Performs usually on streaming data. Involves real time analytics.

**HPC Technologies for Big Data**

**1.**Cluster computing

**2.**Distributed computing

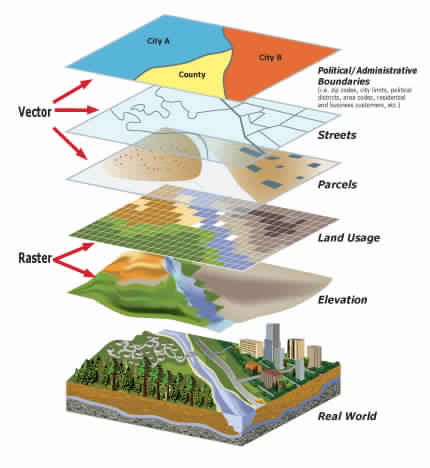
**3.**Specialized Hardware: FPGA

**4.**Speacialized Hardware

**HPC based big data analytics**

Case Study of Data

Eg:-Geospatial data



**MPI(Multilevel parallelism):-** By default, processes in an MPI application can only do one task at a time. Such processes are single-threaded processes. This means that each process has an address space with a single program counter, a set of registers, and a stack.

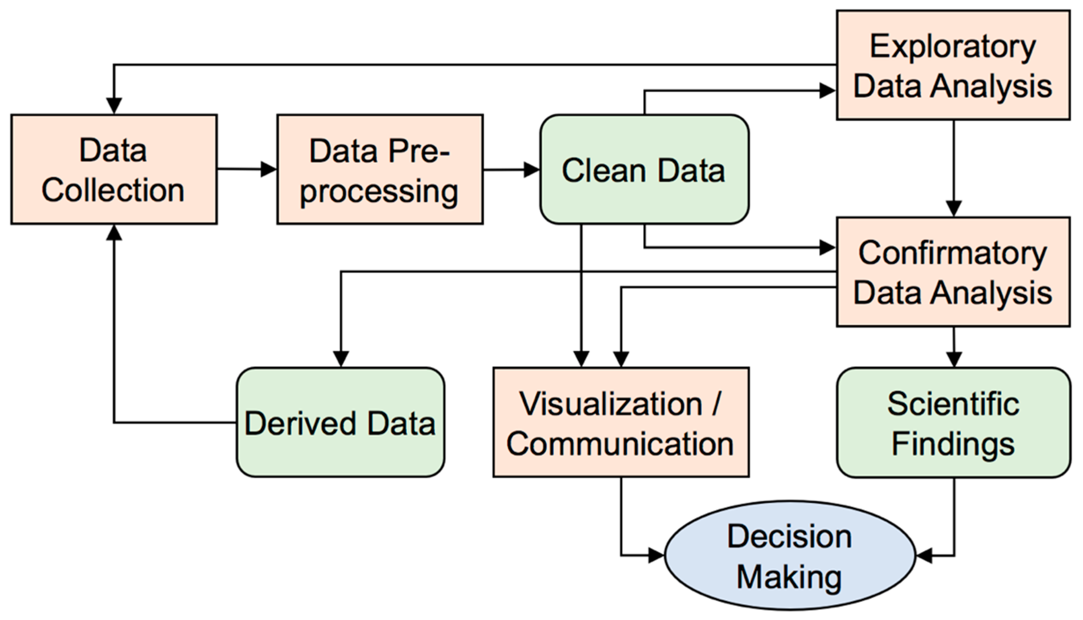
A process with multiple threads has one address space, but each process thread has its own counter, registers, and stack.

Multilevel parallelism refers to MPI processes that have multiple threads. Processes become multithreaded through calls to multithreaded libraries, parallel directives and pragmas, or auto-compiler parallelism.

Multilevel parallelism is beneficial for problems that are possible to decompose into logical parts for parallel execution (for example, a looping construct that spawn’s multiple threads to do a computation and joins after the computation is complete).

**Hierarchical clustering:-** is a clustering algorithm which is used to build hierarchy of clusters. Dendogram is used to show results of hierarchical clustering. The complexity of agglomerative clustering is O(n3),thus in case of large datasets performance of agglomerative clustering is too slow.

**Exploratory Data Analysis**

****

**EDA(exploratory data analysis):-** In statistics, exploratory data analysis is an approach analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

**Qualitative Attribute:-** Qualitative research is a scientific method of observation to gather non-numerical data. This type of research "refers to the meanings, concepts definitions, characteristics, metaphors, symbols, and description of things" and not to their "counts or measures."

In summary, **nominal** variables are used to “name,” or label a series of values. **Ordinal** scales provide good information about the order of choices, such as in a customer satisfaction survey. Interval scales give us the order of values + the ability to quantify the difference between each one.

**Discrete vs Continues Attributes :- Discrete** data are distinct and there is no grey area in between, while **continuous** data occupy any value over a **continuous** data value. **Discrete** data They can take particular values. They are numeric. **Discrete** data can take on only integer values whereas **continuous** data can take on any value.

**Best book for Data mining is han kamber.**

**Natural Language Processing.**

NLP is a way for computers to analyze, understand, and derive meaning from human language in a smart and useful way. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation.

**Machine Translation:- Machine translation**, sometimes referred to by the abbreviation **MT** (not to be confused with [**computer-aided translation**](https://en.wikipedia.org/wiki/Computer-assisted_translation), **machine-aided human translation** (**MAHT**) or [**interactive translation**](https://en.wikipedia.org/wiki/Interactive_machine_translation)) is a sub-field of [computational linguistics](https://en.wikipedia.org/wiki/Computational_linguistics) that investigates the use of software to [translate](https://en.wikipedia.org/wiki/Translation) text or speech from one [language](https://en.wikipedia.org/wiki/Language) to another.

that alone usually cannot produce a good translation of a text because recognition of whole phrases and their closest counterparts in the target language is needed. Solving this problem with [corpus](https://en.wikipedia.org/wiki/Corpus_linguistics) statistical, and [neural](https://en.wikipedia.org/wiki/Machine_translation#Neural_MT) techniques is a rapidly growing field that is leading to better translations, handling differences in [linguistic typology](https://en.wikipedia.org/wiki/Linguistic_typology), translation of [idioms](https://en.wikipedia.org/wiki/Idiom), and the isolation of anomalies.

Current machine translation software often allows for customization by domain or [profession](https://en.wikipedia.org/wiki/Profession) (such as [weather reports](https://en.wikipedia.org/wiki/Meteorology)), improving output by limiting the scope of allowable substitutions. This technique is particularly effective in domains where formal or formulaic language is used. It follows that machine translation of government and legal documents more readily produces usable output than conversation or less standardised text.

**Day-2**

**Artificial Intelligence**

**What is AI?**

**Access Data: -Sensors, Files, Databases**

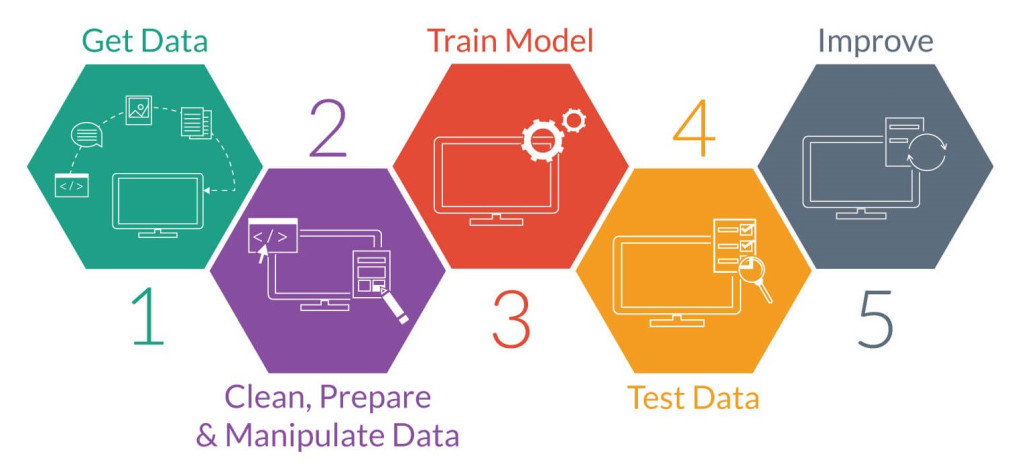
**Analyze Data: -data exploration, preprocessing, Domain-Specific algorithm**

**Develop: - Ai model, Algorithm development**

**Deploy: - Desktop apps, Enterprise system, Embedded Devices**

**Machine learning work flow**

**1.Data 2.Feature Extraction 3.Classification**



**Supervised Machine learning: -** Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples.

**Unsupervised Machine Learning: -** Unsupervised learning is a branch of machine learning that learns from test data that has not been labeled, classified or categorized.

**Neural Networks**

**CS763Github**