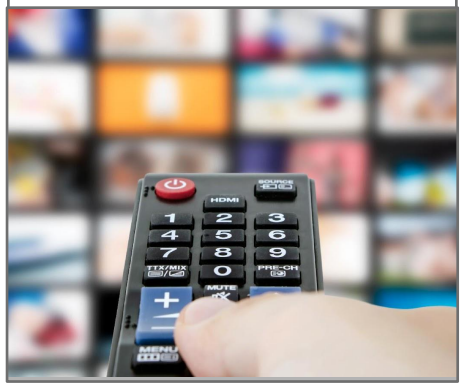
Building and Deploying Recommender Systems Quickly and Easily with NVIDIA Merlin

Recommender Systems

Personalization Engine of Online Services

DIGITAL CONTENT



4.3B Watch Videos Online

E-COMMERCE

3.7B Shop Online

SOCIAL MEDIA

4.3B Active Users

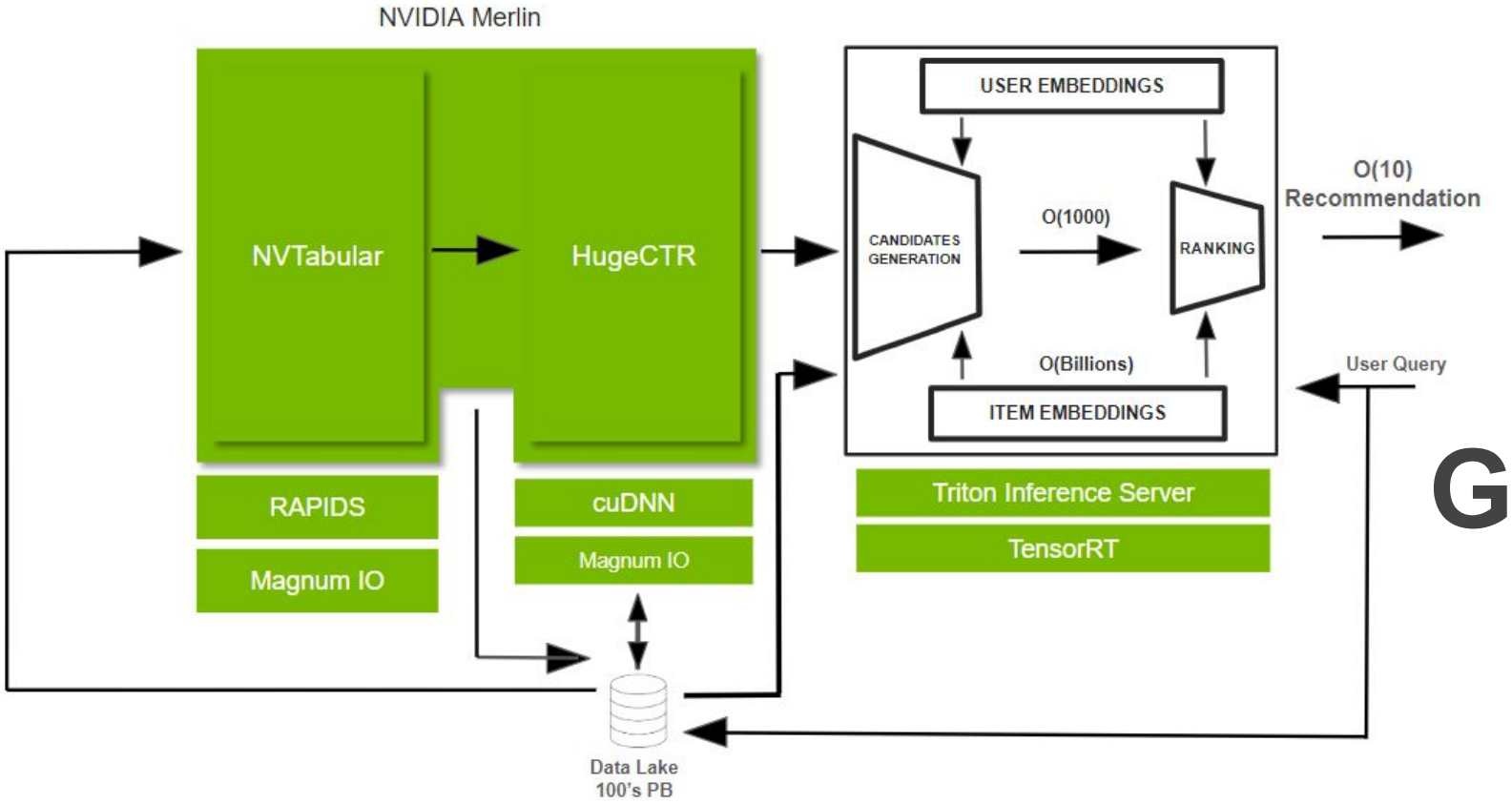
DIGITAL ADVERTISING

4.7B Internet Users

"Already, 35 percent of what consumers purchase on Amazon and 75 percent of what they watch on Netflix come from product recommendations based on such algorithms."

Source: McKinsey

Merlin’s 2 year Birthday!



Open Source

Easy to Use

GPU Accelerated

End to End

Merlin Recommendation Framework

NVTabular

Feature Engineering

Prepare massive datasets in minutes allowing for more exploration and better models.

Dataloading

Asynchronous batch dataloading means the GPU is always utilized

HugeCTR

Scaling Training

Easy to use data and model parallel training allow you to scale to TB sized embeddings

Triton

Deployment

Inference time data transforms and multi model support provide maximum throughput with latency constraints

Merlin HugeCTR

Merlin HugeCTR: GPU-accelerated Recommender System Training and Inference [S41352]

Merlin HugeCTR is a recommender system-specific framework which accelerates the training and deployment of complex deep learning models on GPUs at scale. Since its public release in early 2020, we've added a lot of enhancements to performance and usability. We'll introduce some of them, including the inference architecture, sparse operation kit (SOK), training embedding cache, unified embedding, and the MLPerf optimizations. We'd like to especially highlight two features that have highly extended HugeCTR applicability: • HugeCTR inference architecture, which dramatically accelerates the inference via a well-designed parameter server and GPU embedding cache on top of NVIDIA Triton Inference Server. • Sparse operation kit, which can enable HugeCTR optimized embeddings on common deep learning frameworks such as Horovod and Tensorflow.

Minseok Lee, AI Developer Technology Engineer, NVIDIA

Industry Segment: All Industries

Primary Topic: Recommenders / Personalization

ADD TO SCHEDULE **Wednesday, March 23 | 2:00 PM - 2:50 PM PDT**

Distributed Embedding Caches for Inference

Merlin HugeCTR：使用 GPU 嵌入式缓存的分布式分层推理参数服务器 Merlin HugeCTR: Distributed Hierarchical Inference Parameter Server Using GPU Embedding Cache [S41126]

☆

In advertising/recommendation/search scenarios, the current mainstream algorithm adopts the model structure with embeddings and Deep Neural Network. Especially for doing CTR tasks, searching for embeddings is a highly parallel and memory-intensive step and is very suitable to run on GPUs. However, deep learning models in online advertising industries may have terabyte-scale parameters that do not fit in the GPU memory nor the CPU main memory on a computing node. In this talk we will cover two topics： - Introduce a Distributed Hierarchical GPU-based Inference Parameter Server, abbreviated as HugeCTR PS, for massive scale deep learning recommendation systems. We propose a hierarchical memory storage and model stream updating mechanism that utilizes GPU embedding cache, CPU memory and SSD as 3-layer hierarchical storage while all of the neural network training computations are processed on GPUs. - As the most important data structure of the HPS architecture, the GPU embedding cache is implemented on the NVIDIA GPU which is used to accelerate the embedding tables look-up process. The GPU embedding cache introduced in this talk offloads the majority workload of embedding table look-up to the GPU by utilizing the locality of embeddings that have been queried. This will lower down the latency of the CTR pipeline and improve the throughput.

Yingcan Wei, AI & Distributed Machine Learning Expert, NVIDIA
Fan Yu, Developer Technology Engineer , NVIDIA
Matthias Langer, Senior GPU Computing Engineer, NVIDIA

Industry Segment: All Industries

Primary Topic: Recommenders / Personalization

ADD TO SCHEDULE **Tuesday, March 22 | 11:00 PM - 11:50 PM PDT**

Triton Inference Server

Fast, Scalable, and Standardized AI Inference Deployment for Multiple Frameworks, Diverse Models on CPUs and GPUs with Open-source NVIDIA Triton [S41755]

☆

We'll go over NVIDIA Triton Inference Server software and what's new. Triton is an open-source inference-serving software for fast and scalable AI in applications. Learn how Triton helps deploy models from all popular frameworks — including TensorFlow, PyTorch, ONNX, TensorRT, RAPIDS FIL (for XGBoost, Scikit-learn Random Forest, LightGBM), OpenVINO, Python, and even custom C++ backends. Also learn about the features that help optimize inference for multiple query types — real-time, batch, streaming, and model ensembles. We'll cover how to deploy a standardized inference in production on both NVIDIA GPUs and x86 and ARM CPUs in cloud or data center, enterprise edge, and even on embedded devices like the NVIDIA Jetson, as well as how to use Triton in virtualized environments (e.g. VMware vSphere), Kubernetes, and machine-learning platforms like Amazon SageMaker, Azure ML, and Google Vertex AI.

Shankar Chandrasekaran, Product Marketing Manager, NVIDIA

Mahan Salehi, Product Manager, NVIDIA

Industry Segment: All Industries

Primary Topic: Deep Learning - Inference

ADD TO SCHEDULE **Wednesday, March 23 | 8:00 AM - 8:50 AM PDT**

NVIDIA Merlin 1.0
Develop, deploy and maintain recommender systems

	Model Development	Deployment	Production (1)
Who:	Data Scientists / ML Engineers	ML Engineers / Product Engineers	Product Engineers / ML Ops
Needs:	Quick iteration over feature engineering and model training	Easily deploying new models and workflows into production	Monitoring and maintaining many recommender systems
Merlin:	▪ Accelerates pipelines for fast experimentation cycle ▪ Integrates ETL and model training ▪ Implements common architectures, loss functions, sampling strategies, etc. ▪ Flexibility to build your own	▪ Simple API to push to production ▪ Deploys ETL and multi-stage models as ensemble ▪ Supports retrieval, filtering, and other common pipeline stages ▪ Scalable and accelerated components	▪ Standardize production workflow for all use cases ▪ Integration to other components for logging, feature storage, etc.

(1) Coming soon in Merlin v2.0

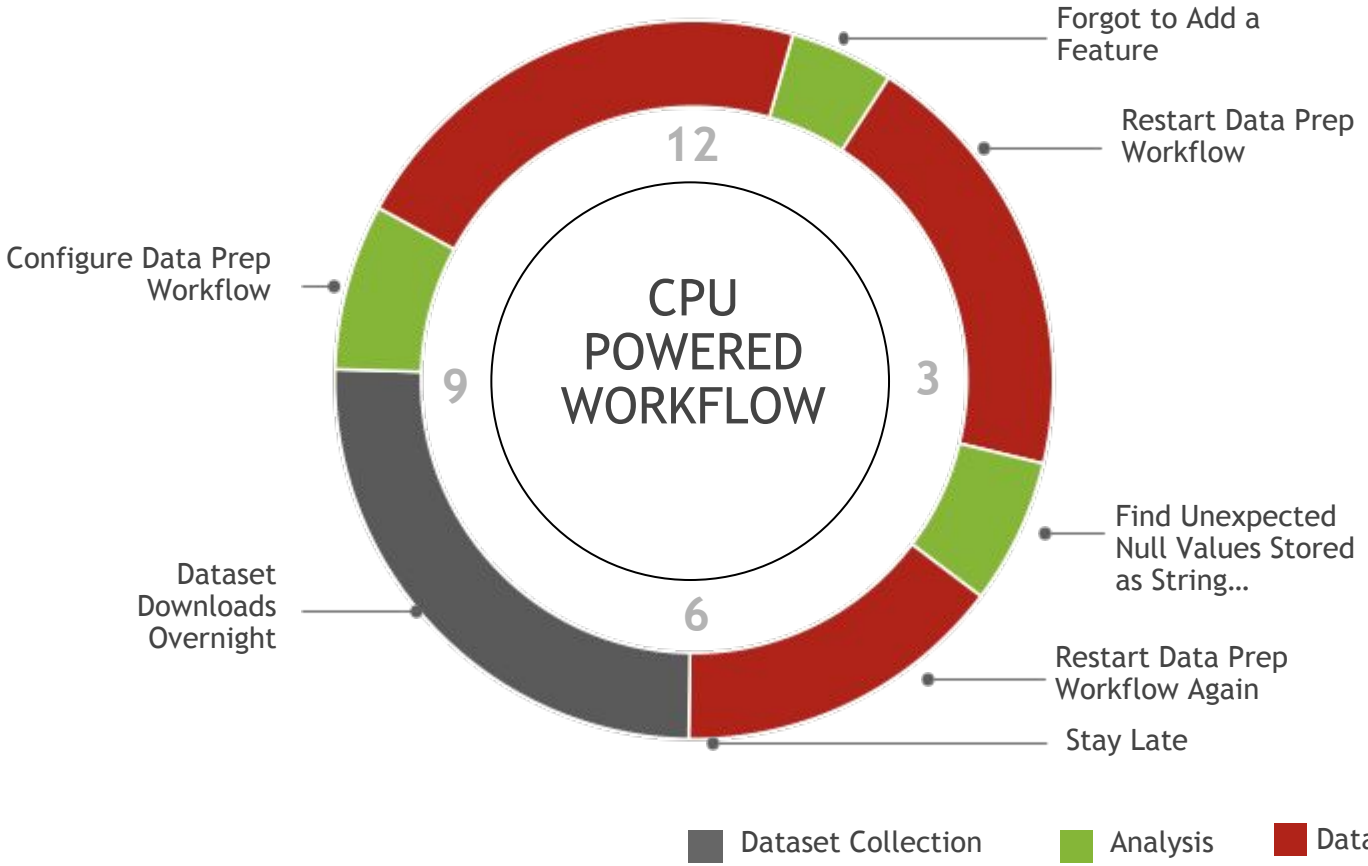
NVIDIA Merlin supports common tasks required to develop, deploy and maintain recommender systems

	Model Development	Deployment	Production (1)
Who:	Data Scientists / ML Engineers	ML Engineers / Product Engineers	Product Engineers / ML Ops
Needs:	Quick iteration over feature engineering and model training	Easily deploying new models and workflows into production	Monitoring and maintaining many recommender systems
Merlin:	▪ Accelerates pipelines for fast experimentation cycle ▪ Integrates ETL and model training ▪ Implements common architectures, loss functions, sampling strategies, etc. ▪ Flexibility to build your own	▪ Simple API to push to production ▪ Deploys ETL and multi-stage models as ensemble ▪ Supports retrieval, filtering, and other common pipeline stages ▪ Scalable and accelerated components	▪ Standardize production workflow for all use cases ▪ Integration to other components for logging, feature storage, etc.

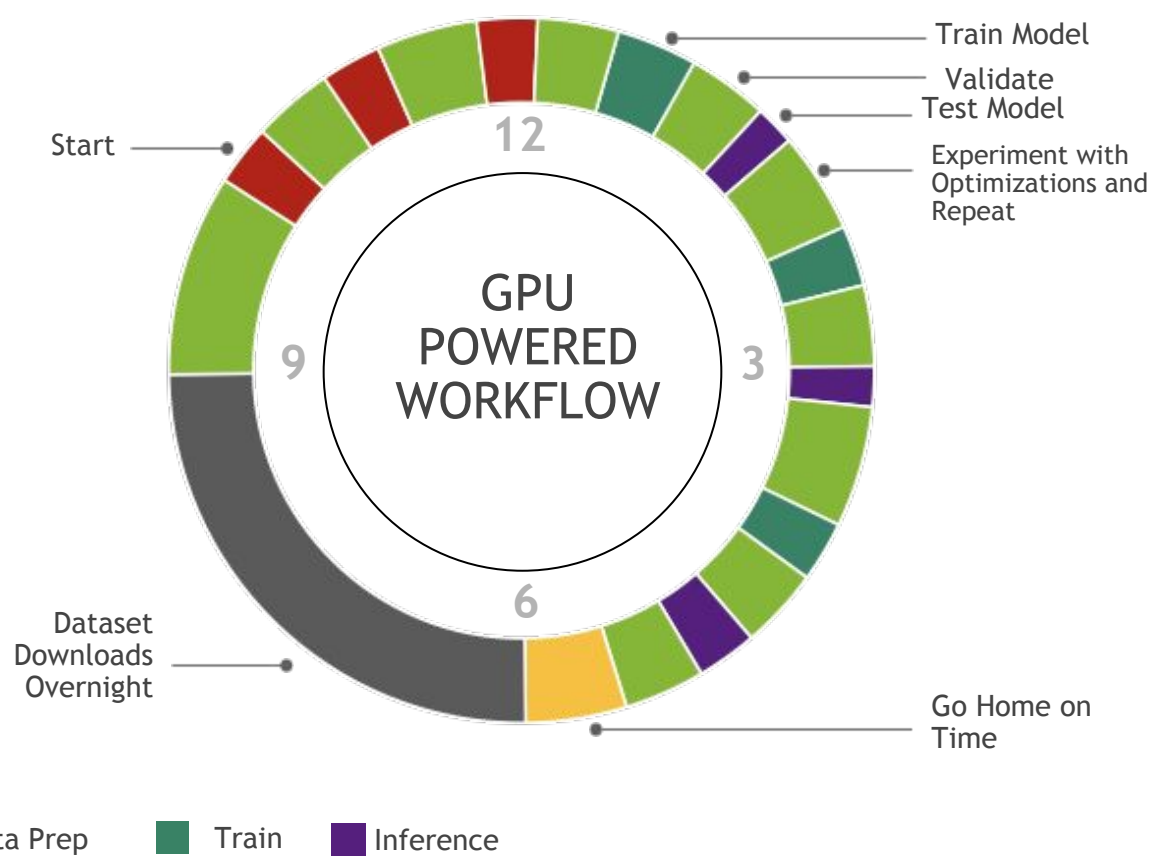
(1) Coming soon in Merlin v2.0

Day In The Life Of A Data Scientist

Regular Compute



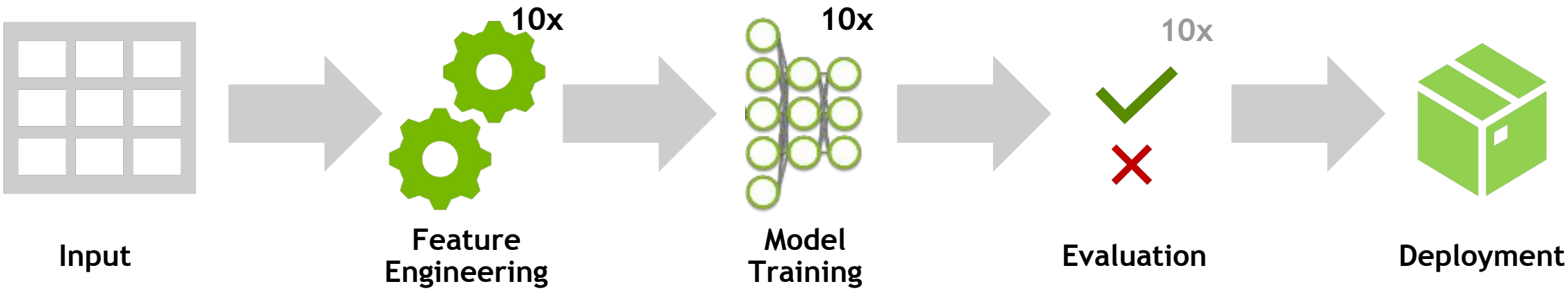
Accelerated Compute



Dataset Collection | Analysis | Data Prep | Train | Inference

The average data scientist spends >75% of their time iterating over features or model parameters

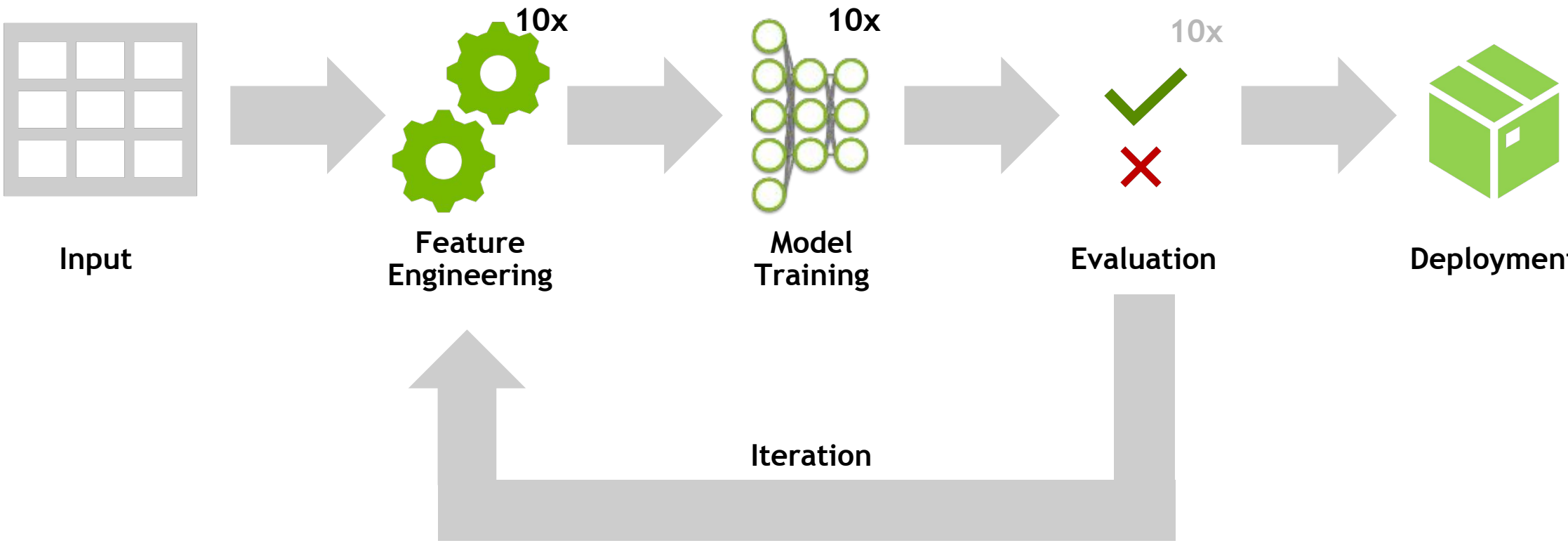
GPU acceleration allows us to do each stage much faster



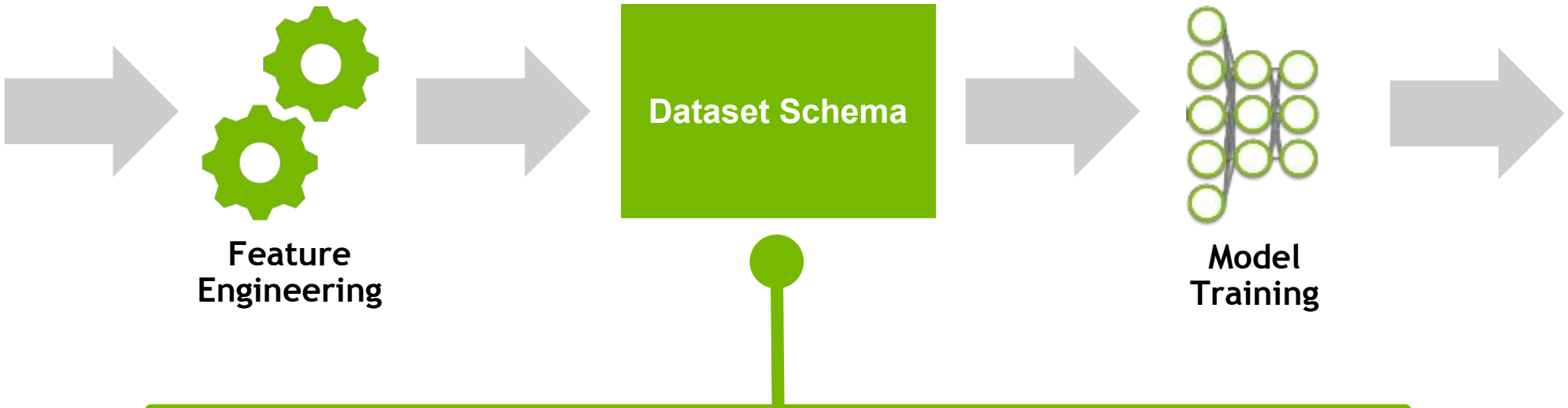
This changes the bottleneck from compute time to development time…

Fast Iteration is Key

Merlin enables quick experimentation cycles to find high accuracy models



Close integration between feature engineering and training reduces the development time

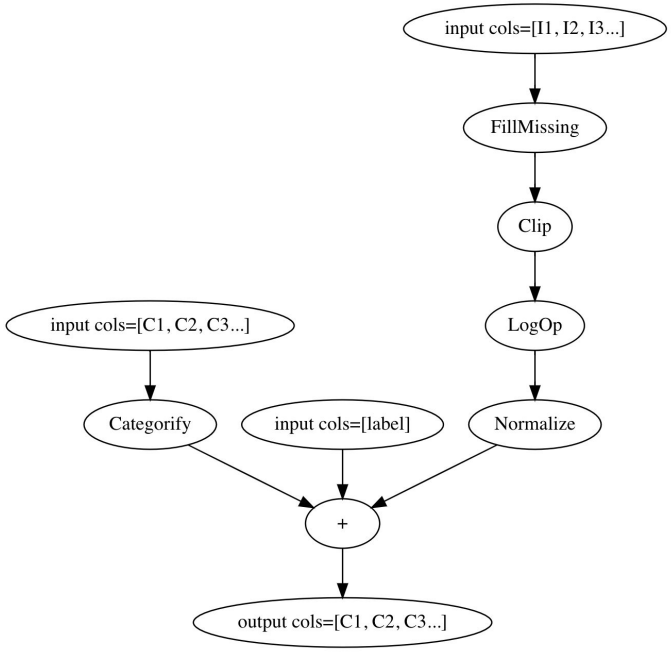


- Model training requires information about the dataset, for example which columns are categorical and numerical, cardinality of categorical features, etc.
- Feature engineering step can collect the information and provide them in a defined schema file
- Model can be defined based on the provided information from the schema file
- Updating the feature engineering (for example adding new features) does not require to change the model training workflow (and vice-versa)

Merlin simplifies the dependency by providing a schema which connects feature engineering and model training

Feature Engineering with NVTabular

NVTabular



Visualization of feature engineering and preprocessing pipeline for Criteo Click Ads Prediction dataset

ETL Workflow:

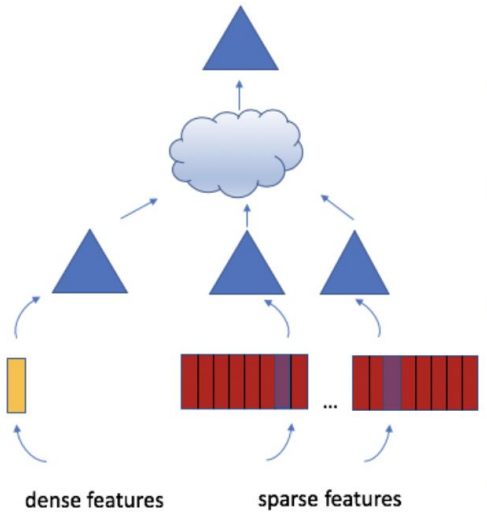
```
%%time
## Define the pipeline

# We add tags for user_id, item_id and target. NVTabular will provide an output file
# We categorify the user_id and item_id to be continuous integers 0, ..., |C|
user_id = ["user_id"] >> AddMetadata(tags=[Tags.USER_ID]) >> Categorify()
item_id = ["item_id"] >> AddMetadata(tags=[Tags.ITEM_ID]) >> Categorify()
targets = ["click"] >> AddMetadata(
    tags=[str(Tags.BINARY_CLASSIFICATION), "target"]
)

outputs = user_id+item_id+targets
```

We will also use a util function to wrap up the workflow transform to a one line of code.

- Common ops used to transform RecSys data
- CPU & GPU compatible
- Same workflow for training and inference
- **Automatic generation of configurable Merlin Schemas for your data**

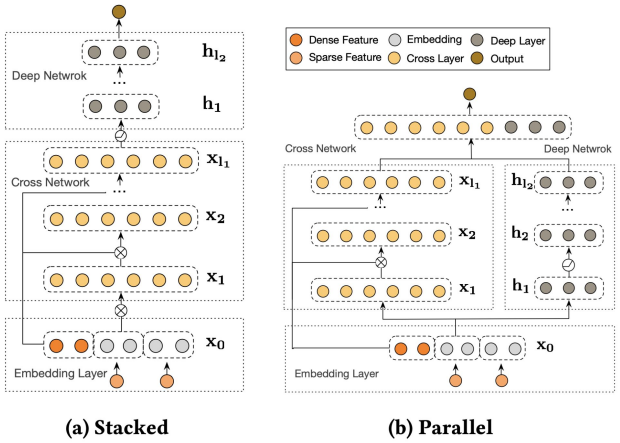
Defining model architectures with Merlin Models in TensorFlow

Facebook DRLM




```
model = mm.DLRMModel(
    schema,
    embedding_dim=64,
    bottom_block=mm.MLPBlock([128, 64]),
    top_block=mm.MLPBlock([128, 64, 32]),
    prediction_tasks=mm.BinaryClassificationTask(
        target_column,
        metrics=[tf.keras.metrics.AUC()]
    )
)
```

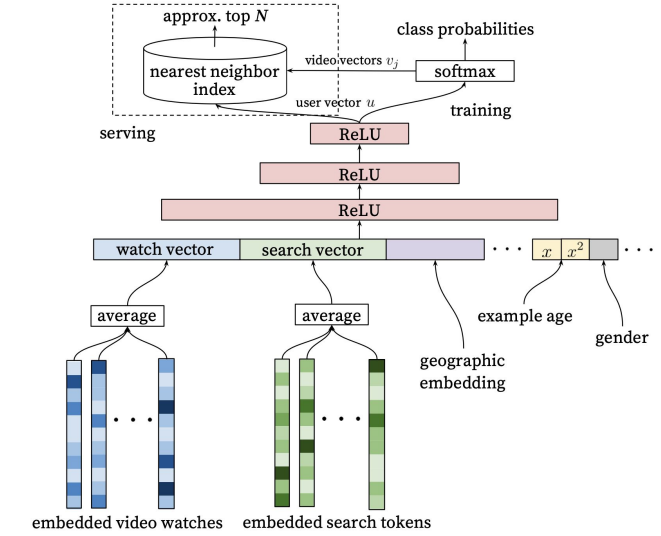
Google DCN



(a) Stacked (b) Parallel

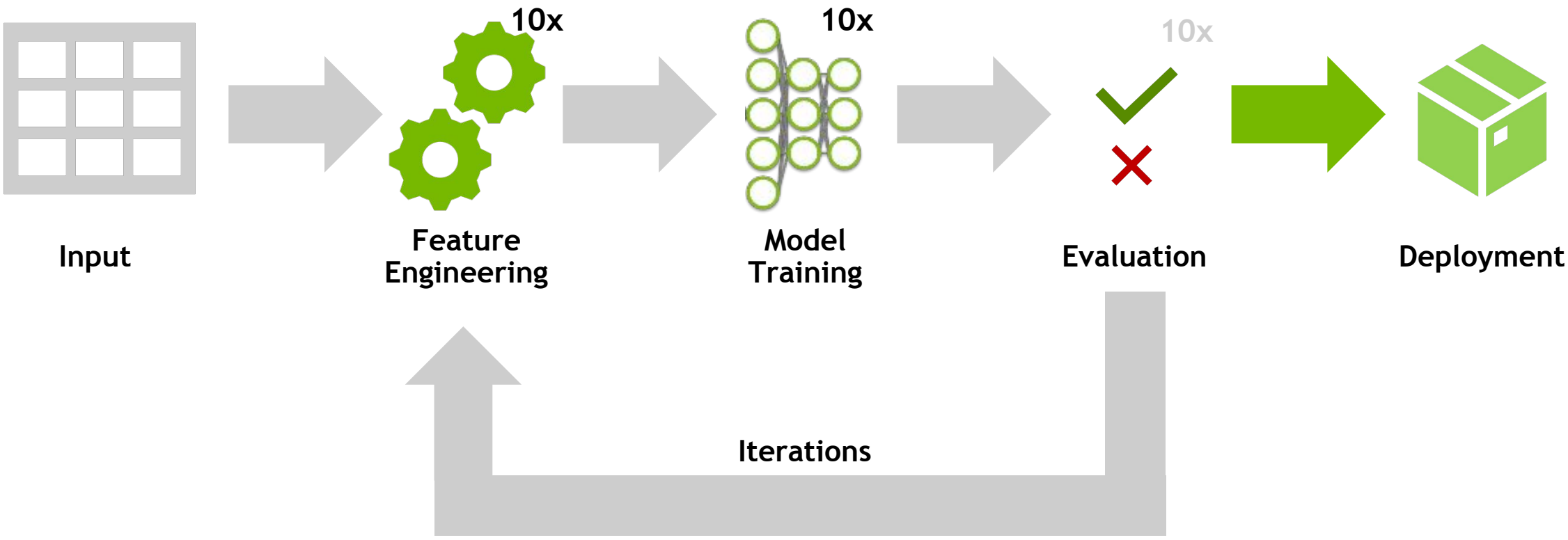
```
model = mm.DCNModel(
    schema,
    depth=2,
    deep_block=mm.MLPBlock([128, 64]),
    prediction_tasks=mm.BinaryClassificationTask(
        target_column,
        metrics=[tf.keras.metrics.AUC()]
    )
)
```

YouTube DNN




```
model = mm.YoutubeDNNRetrievalModel(
    schema,
    top_block=MLPBlock([128, 64]),
    num_sampled=100
)
```

Deployment of a single ranking model to Triton Inference Server



Merlin Models Hands on Tutorial

Building Recommender Systems More Easily using Merlin Models [DLIT2043]

Recommender systems (RecSys) help users to discover products, contents, and more in online services. RecSys models may use diverse input types, architectures (e.g., two-tower or sequential) or tasks (e.g., binary, multi-class classification, multi-task). Building scalable RecSys is not trivial, and it requires multiple sub-models (e.g., a candidate generation and ranking model) to work systematically. Merlin Models is an open-source library providing flexible building blocks to define a broad range of models with various prediction tasks, loss functions, and negative sampling techniques. Its unified API enables users to create models in TensorFlow or PyTorch. We'll present the main concepts and techniques in RecSys, and show how to design an end-to-end recommender system easily using the Merlin Models on GPU. You need basic knowledge of Python and Tensorflow or PyTorch framework, and should understand machine and deep learning concepts/pipelines.

***IMPORTANT:** DLI Training Labs are free to attend with your GTC registration, but limited capacity and first-come, first-served. You may favorite or add a training lab to your schedule, but this does not guarantee you a seat. Rooms will be accessible 15 minutes before the session begins and can be accessed by clicking the "Join Now" button in the GTC Session Catalog. If the "Join Now" button isn't visible, you may need to refresh the page. Once the lab reaches capacity, you will no longer be able to enter the room. To get the most from your hands-on learning experience, please complete these steps prior to getting started.

Ronay Ak, Senior Data Scientist, NVIDIA

Benedikt Schifferer, Deep Learning Engineer, NVIDIA

Industry Segment: Consumer Internet

Primary Topic: Recommenders / Personalization

ADD TO SCHEDULE **Wednesday, March 23 | 9:00 AM - 11:00 AM PDT**

Benedikt Schifferer

- Deep Learning Engineer for Recommender Systems at NVIDIA
- Built Recommender System 2 years at a German ecommerce
- MSc. Data Science from Columbia University
- https://www.linkedin.com/in/benedikt-schifferer/

Ronay Ak

- Sr. Data Scientist for Recommender Systems at NVIDIA
- PhD. in Energy & Power Systems (Engineering) from CentraleSupelec, FR.
- https://www.linkedin.com/ronay-ak

Recommender Models vs Recommender Systems

A **system** is a group of interacting or interrelated elements that act according to a set of rules to form a unified whole.

The Four Stages of a Recommender System

1. **Retrieval**: Fetch a **small set of candidate items** from the massive item catalog relevant for the current user

2. **Filtering: Remove candidate items that aren't appropriate** or available

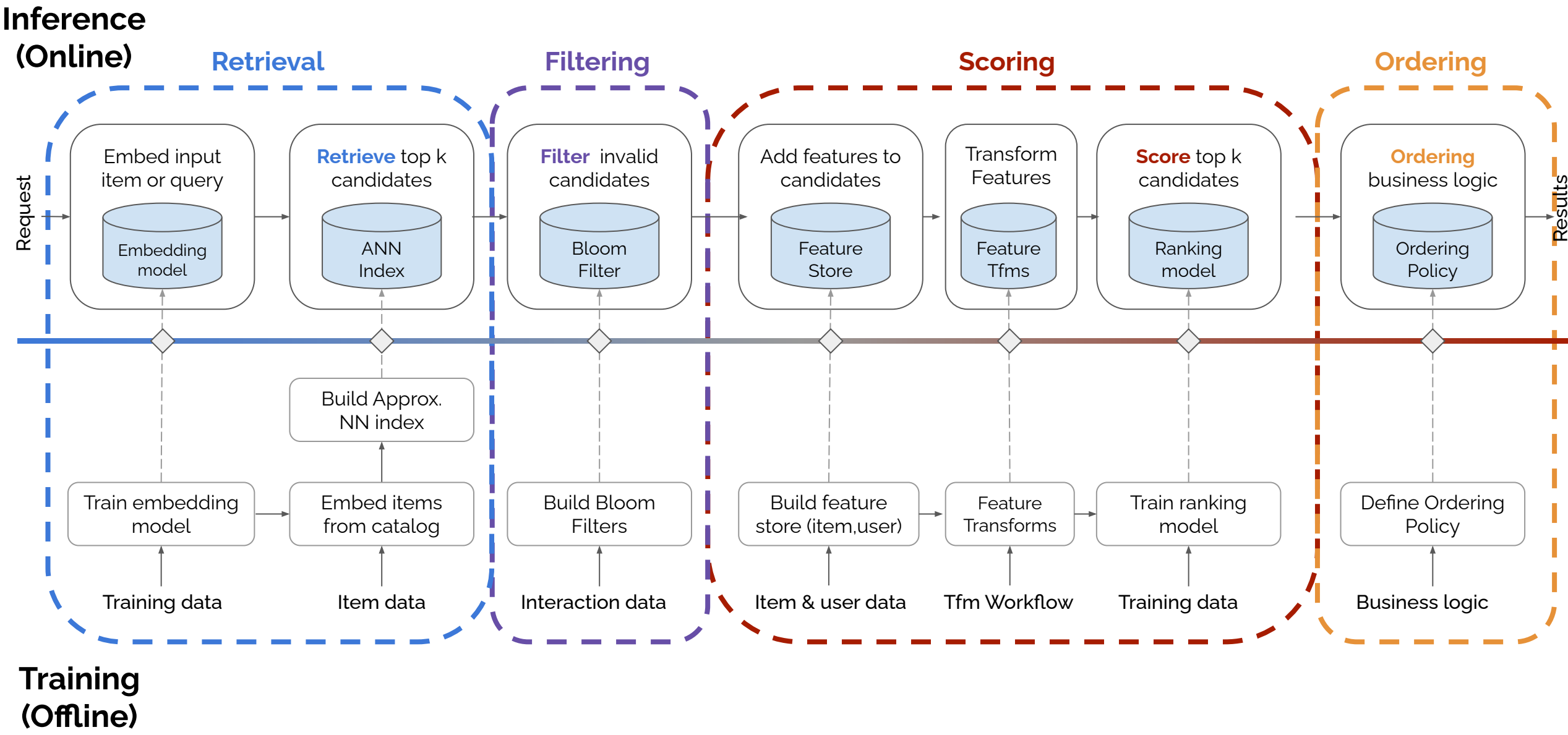
3. **Scoring**: **Assign a** relevance **score** to each remaining item

4. **Ordering:** Choose which of the candidate items to include in the final list of recommendations and **put them in an optimal order**

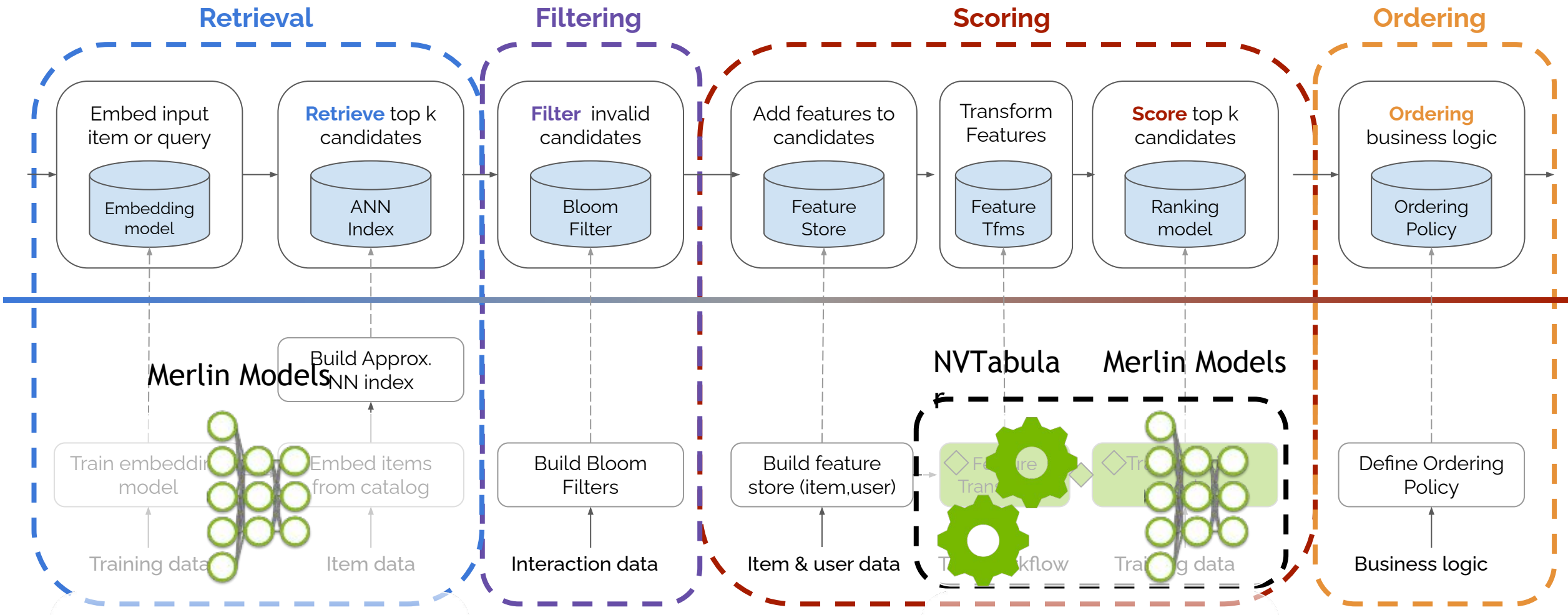
Examples for the 4 stage recommender systems

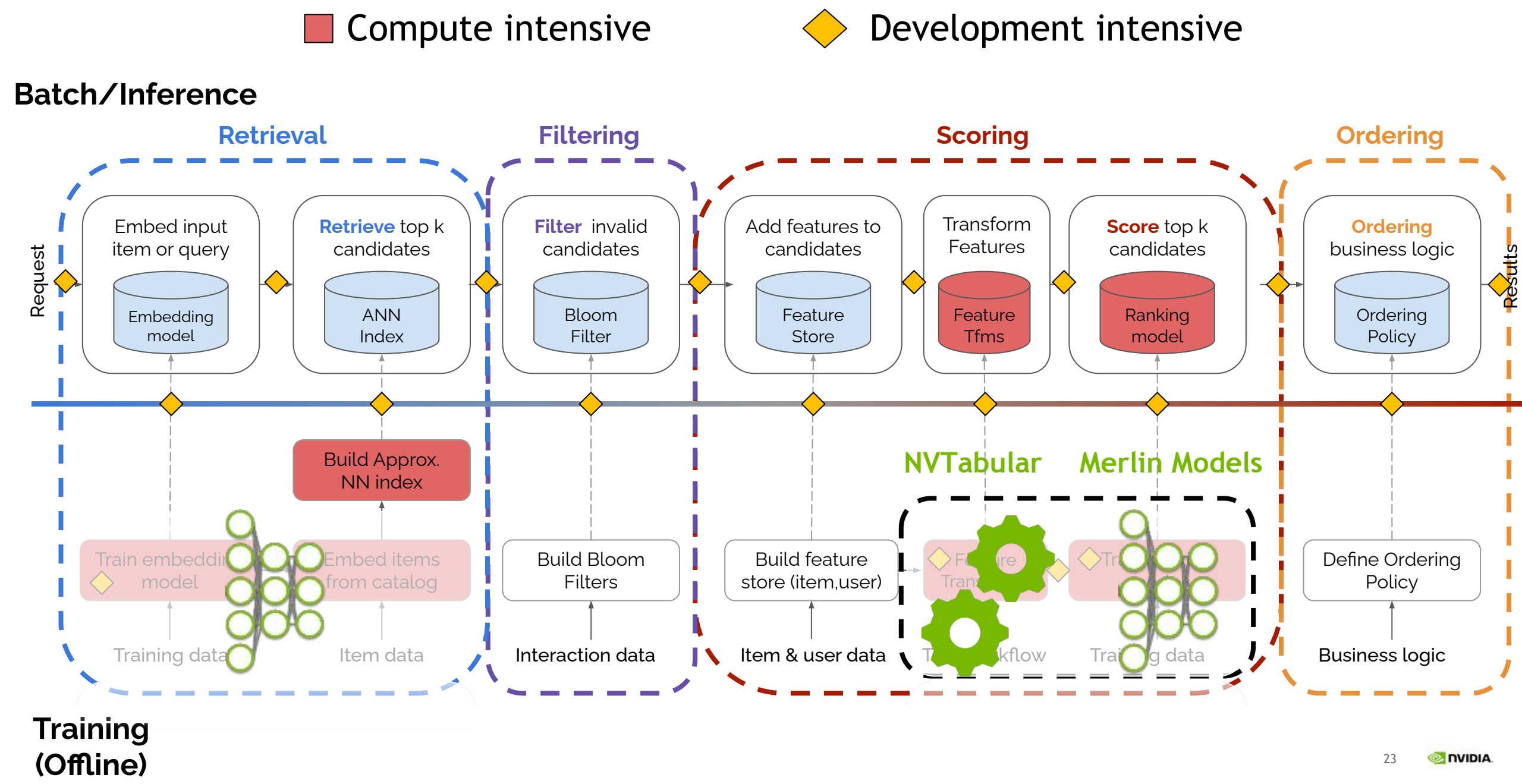
	Retrieval	Filtering	Scoring	Ordering
Music Discovery	Find similar songs based on nearest neighbour search	Remove tracks users listened before	Predict likelihood that a user will listen to a song	Trade-Off between score, similarity, BPM, etc
Social Media	Find new posts in user's network	Remove posts from blocked and muted users	Predict likelihood that a user will interact with it	Change order that adjust posts are from different authors
Online Store	Find items which are usually co-purchased	Remove items which are out of stock	Predict likelihood that a user will purchase an item	Reorder items based on price points
Streaming Service	Find items based on different rows/shelves/topics	Remove items which are not available for user's country	Predict user's stream time per item	Organize recommendations to fit genre distributions

End to End Recommender Systems in Production



Merlin 1.0 for Model Development




Compute intensive
Development intensive
Batch/Inference
Retrieval
Filtering
Scoring
Ordering
Request
Embed input item or query
Embedding model
Retrieve top k candidates
ANN Index
Filter invalid candidates
Bloom Filter
Add features to candidates
Feature Store
Transform Features
Feature Tfms
Score top k candidates
Ranking model
Ordering business logic
Ordering Policy
Results
Build Approx. NN index
NVTabular
Merlin Models
Train embeddi model
Embed items from catalog
Build Bloom Filters
Build feature store (item,user)
Define Ordering Policy
Training data
Item data
Interaction data
Item & user data
Business logic
Training (Offline)

NVIDIA Merlin supports common tasks required to develop, deploy and maintain recommender systems

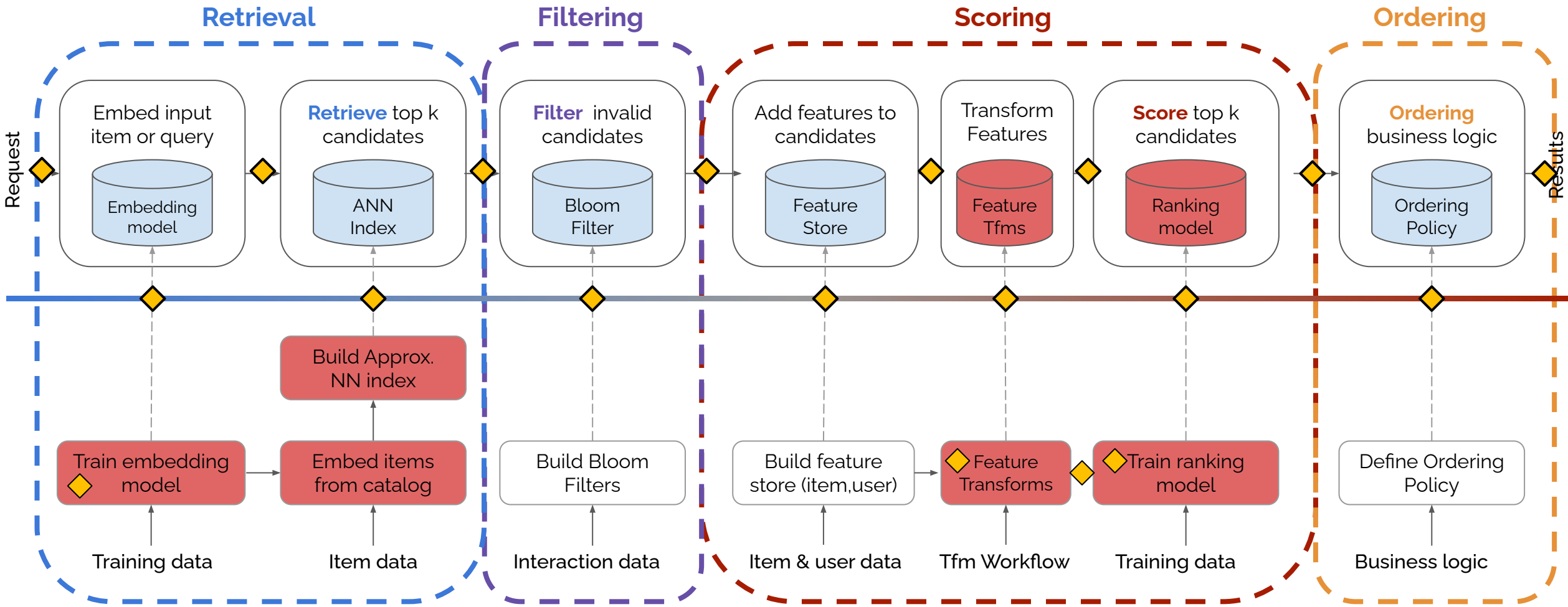
	Model Development	Deployment	Production (1)
Who:	Data Scientists / ML Engineers	ML Engineers / Product Engineers	Product Engineers / ML Ops
Needs:	Quick iteration over feature engineering and model training	Easily deploying new models and workflows into production	Monitoring and maintaining many recommender systems
Merlin:	▪ Accelerates pipelines for fast experimentation cycle ▪ Integrates ETL and model training ▪ Implements common architectures, loss functions, sampling strategies, etc. ▪ Flexibility to build your own	▪ Simple API to push to production ▪ Deploys ETL and multi-stage models as ensemble ▪ Supports retrieval, filtering, and other common pipeline stages ▪ Scalable and accelerated components	▪ Standardize production workflow for all use cases ▪ Integration to other components for logging, feature storage, etc.

(1) Coming soon in Merlin v2.0

Compute intensive

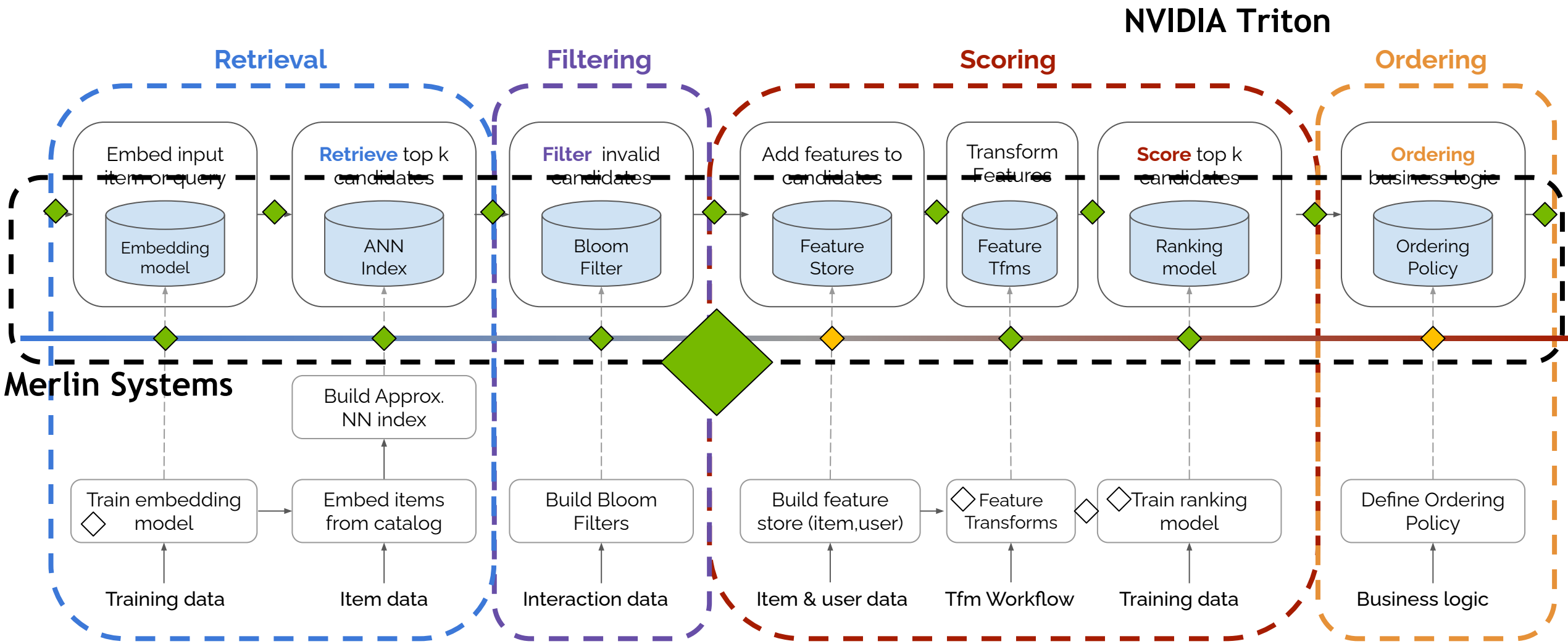
Development intensive

Batch/Inference



Training (Offline)

Merlin 1.0 for ML Engineers / Product Engineers



Merlin Systems (Experimental)

- Combines all workflow stages into a single Triton ensemble for ease of deployment
- Single artifact for deployment for the entire pipeline
- Type checking for pipeline connections
- Same graph representation as NVTabular

Future work:

- Pipeline training and Batch execution using the same artifact

```
user_features = ["user_id"] >> QueryFeast(
    feast_repo_path,
    entity_view="user_features",
    entity_id="user_id",
    entity_column="user_id",
    features=["movie_id_count"],
    mh_features=["movie_ids", "genres", "search_terms"],
    input_schema=feast_user_in_schema,
    output_schema=feast_user_out_schema,
)

retrieval = (
    user_features
    >> PredictTensorflow(
        retrieval_model_path,
        custom_objects={"sampled_softmax_loss": sampled_softmax_loss},
    )
    >> QueryFaiss(faiss_index_path, query_vector_col="output_1", topk=100)
)

filtering = user_features["movie_ids_1"] + retrieval["candidate_ids"] >> FilterCandidates(
    candidate_col="candidate_ids", filter_col="movie_ids_1"
)

item_features = filtering >> QueryFeast(
    feast_repo_path,
    entity_view="movie_features",
    entity_id="movie_id",
    entity_column="filtered_ids",
    features=["tags_nunique"],
    mh_features=["genres", "tags_unique"],
    input_schema=Schema([ColumnSchema("filtered_ids", dtype=np.int32)]),
    output_schema=feast_item_out_schema,
    include_id=True,
    output_prefix="movie",
)

combined_features = user_features + item_features >> UnrollFeatures(
    "movie_id", feast_user_out_schema.column_names, unrolled_prefix="user"
)

ranking = combined_features >> PredictTensorflow(ranking_model_path)

ordering = (combined_features + ranking)["movie_id", "output_1"] >> SoftmaxSampling(
    "movie_id", relevance_col="output_1", topk=10, temperature=20.0
)

export_path = str("/nvtabular/test_poc/")

ensemble = Ensemble(ordering, request_schema)
ens_config, node_configs = ensemble.export(export_path)
```

We've come far but there's a long way to go…

	Model Development	Deployment	Production
Who:	Data Scientists / ML Engineers	ML Engineers / Product Engineers	Product Engineers / ML Ops
Needs:	Quick iteration over feature engineering and model training	Easily deploying new models and workflows into production	Monitoring and maintaining many recommender systems
Merlin:	▪ Accelerates pipelines for fast experimentation cycle ▪ Integrates ETL and model training ▪ Implements common architectures, loss functions, sampling strategies, etc. ▪ Flexibility to build your own	▪ Simple API to push to production ▪ Deploys ETL and multi-stage models as ensemble ▪ Supports retrieval, filtering, and other common pipeline stages ▪ Scalable and accelerated components	▪ Standardize production workflow for all use cases ▪ Integration to other components for logging, feature storage, etc.

NVIDIA Merlin Team

Providing easy to develop, performant, end to end recommender systems on the GPU


NVIDIA
NVIDIA-Merlin
NVIDIA Merlin Team
Join us!
Providing easy to develop, performant, end to end recommender systems on the GPU

Don’t forget about our other Merlin talks!

Merlin HugeCTR: GPU-accelerated Recommender System Training and Inference [S41352]

Merlin HugeCTR is a recommender system-specific framework which accelerates the training and deployment of complex deep learning models on GPUs at scale. Since its public release in early 2020, we've added a lot of enhancements to performance and usability. We'll introduce some of them, including the inference architecture, sparse operation kit (SOK), training embedding cache, unified embedding, and the MLPerf optimizations. We'd like to especially highlight two features that have highly extended HugeCTR applicability: • HugeCTR inference architecture, which dramatically accelerates the inference via a well-designed parameter server and GPU embedding cache on top of NVIDIA Triton Inference Server. • Sparse operation kit, which can enable HugeCTR optimized embeddings on common deep learning frameworks such as Horovod and Tensorflow.

Minseok Lee, AI Developer Technology Engineer, NVIDIA

Industry Segment: All Industries

Primary Topic: Recommenders / Personalization

ADD TO SCHEDULE **Wednesday, March 23 | 2:00 PM - 2:50 PM PDT**

Merlin HugeCTR：使用 GPU 嵌入式缓存的分布式分层推理参数服务器 Merlin HugeCTR: Distributed Hierarchical Inference Parameter Server Using GPU Embedding Cache [S41126]

In advertising/recommendation/search scenarios, the current mainstream algorithm adopts the model structure with embeddings and Deep Neural Network. Especially for doing CTR tasks, searching for embeddings is a highly parallel and memory-intensive step and is very suitable to run on GPUs. However, deep learning models in online advertising industries may have terabyte-scale parameters that do not fit in the GPU memory nor the CPU main memory on a computing node. In this talk we will cover two topics： - Introduce a Distributed Hierarchical GPU-based Inference Parameter Server, abbreviated as HugeCTR PS, for massive scale deep learning recommendation systems. We propose a hierarchical memory storage and model stream updating mechanism that utilizes GPU embedding cache, CPU memory and SSD as 3-layer hierarchical storage while all of the neural network training computations are processed on GPUs. - As the most important data structure of the HPS architecture, the GPU embedding cache is implemented on the NVIDIA GPU which is used to accelerate the embedding tables look-up process. The GPU embedding cache introduced in this talk offloads the majority workload of embedding table look-up to the GPU by utilizing the locality of embeddings that have been queried. This will lower down the latency of the CTR pipeline and improve the throughput.

Yingcan Wei, AI & Distributed Machine Learning Expert, NVIDIA

Fan Yu, Developer Technology Engineer , NVIDIA

Matthias Langer, Senior GPU Computing Engineer, NVIDIA

Industry Segment: All Industries

Primary Topic: Recommenders / Personalization

ADD TO SCHEDULE **Tuesday, March 22 | 11:00 PM - 11:50 PM PDT**

Fast, Scalable, and Standardized AI Inference Deployment for Multiple Frameworks, Diverse Models on CPUs and GPUs with Open-source NVIDIA Triton [S41755]

We'll go over NVIDIA Triton Inference Server software and what's new. Triton is an open-source inference-serving software for fast and scalable AI in applications. Learn how Triton helps deploy models from all popular frameworks — including TensorFlow, PyTorch, ONNX, TensorRT, RAPIDS FIL (for XGBoost, Scikit-learn Random Forest, LightGBM), OpenVINO, Python, and even custom C++ backends. Also learn about the features that help optimize inference for multiple query types — real-time, batch, streaming, and model ensembles. We'll cover how to deploy a standardized inference in production on both NVIDIA GPUs and x86 and ARM CPUs in cloud or data center, enterprise edge, and even on embedded devices like the NVIDIA Jetson, as well as how to use Triton in virtualized environments (e.g. VMware vSphere), Kubernetes, and machine-learning platforms like Amazon SageMaker, Azure ML, and Google Vertex AI.

Shankar Chandrasekaran, Product Marketing Manager, NVIDIA

Mahan Salehi, Product Manager, NVIDIA

Industry Segment: All Industries

Primary Topic: Deep Learning - Inference

ADD TO SCHEDULE **Wednesday, March 23 | 8:00 AM - 8:50 AM PDT**

Building Recommender Systems More Easily using Merlin Models [DLIT2043]

Recommender systems (RecSys) help users to discover products, contents, and more in online services. RecSys models may use diverse input types, architectures (e.g., two-tower or sequential) or tasks (e.g., binary, multi-class classification, multi-task). Building scalable RecSys is not trivial, and it requires multiple sub-models (e.g., a candidate generation and ranking model) to work systematically. Merlin Models is an open-source library providing flexible building blocks to define a broad range of models with various prediction tasks, loss functions, and negative sampling techniques. Its unified API enables users to create models in TensorFlow or PyTorch. We'll present the main concepts and techniques in RecSys, and show how to design an end-to-end recommender system easily using the Merlin Models on GPU. You need basic knowledge of Python and Tensorflow or PyTorch framework, and should understand machine and deep learning concepts/pipelines.

***IMPORTANT:** DLI Training Labs are free to attend with your GTC registration, but limited capacity and first-come, first-served. You may favorite or add a training lab to your schedule, but this does not guarantee you a seat. Rooms will be accessible 15 minutes before the session begins and can be accessed by clicking the "Join Now" button in the GTC Session Catalog. If the "Join Now" button isn't visible, you may need to refresh the page. Once the lab reaches capacity, you will no longer be able to enter the room. To get the most from your hands-on learning experience, please complete these steps prior to getting started.

Ronay Ak, Senior Data Scientist, NVIDIA

Benedikt Schifferer, Deep Learning Engineer, NVIDIA

Industry Segment: Consumer Internet

Primary Topic: Recommenders / Personalization

ADD TO SCHEDULE **Wednesday, March 23 | 9:00 AM - 11:00 AM PDT**

And these great talks by people using Merlin…

Personalization and Recommendations Platform for Omni-channel Commerce [S42123] ☆

I'll outline some practical lessons and challenges learned from training and deploying AI-powered learning systems that power personalization and recommendations for the world's largest omni-channel retailer. How do we handle petabytes of omni-channel data to train recommender systems, model customer preferences and affinities amenable for online inference, and (importantly) how do we account for a multitude of micro intents that manifest in user-session logs? I'll cover the foundational elements in our personalized platform and offer a deep dive on the recommendation engine that powers the omni-channel repurchase journey. I will also highlight the advances NVIDIA GPUs and other software components, like NVIDIA Merlin and Triton Inference Server, have brought to our platform.

Kannan Achan, VP - Personalization and Recommendations, Walmart Global Tech

Industry Segment: Retail

Primary Topic: Recommenders / Personalization

ADD TO SCHEDULE **Tuesday, March 22 | 12:00 PM - 12:50 PM PDT**

Building Next-gen Class Recommendations at Peloton with NVIDIA Merlin [S41259] ☆

Peloton's recommendations, much like the business as a whole, reside at a unique intersection of fitness, software, hardware, content, and music. Learn how we built our next-gen Context-Aware Recommender Systems (CARS) with NVIDIA Merlin that can harness the variety of metadata that our on-demand workout classes have, combined with diverse fitness habits of members. Specifically, we leverage NVTabular (NVT) to quickly preprocess our feature-rich datasets, which are then consumed by HugeCTR models. Moreover, NVT is essential for our online inference systems, as stored NVT workflows guarantee that features retrieved in real time are treated the same way as in training. Overall, Merlin's power has helped Peloton scale our recommendation systems as we continue to grow our user base, class catalogue, and platforms.

Shoya Yoshida, Machine Learning Engineer, Peloton

Gayatri Shandar, Machine Learning Engineer, Peloton

Industry Segment: Cloud Services

Primary Topic: Recommenders / Personalization

ADD TO SCHEDULE **Tuesday, March 22 | 12:00 PM - 12:50 PM PDT**

Building Large-scale Recommendation Systems on Google Cloud with NVIDIA Merlin and Vertex AI [S41882] ☆

Recommender systems have become ubiquitous in virtually every industry. With the rapid growth in the scale of industry datasets, deep learning (DL) recommender models have started to gain advantages over traditional methods by capitalizing on large amounts of training data. However, developing and operationalizing large-scale DL recommender models can be very challenging. In this talk, we will discuss how the NVIDIA Merlin, a recommender system framework, and Google Cloud Vertex AI, a unified artificial intelligence platform, can be used to streamline and accelerate the development and deployment of large-scale DL recommenders.

Jarek Kazmierczak, Solutions Architect, Google

James Sohn, DL SW Product Manager, NVIDIA

Industry Segment: All Industries

Primary Topic: Recommenders / Personalization

ADD TO SCHEDULE **Wednesday, March 23 | 12:00 PM - 12:50 PM PDT**

Scaling Real-time Deep Learning Recommendation Inference at a 150M+ User Scale [S42547] ☆

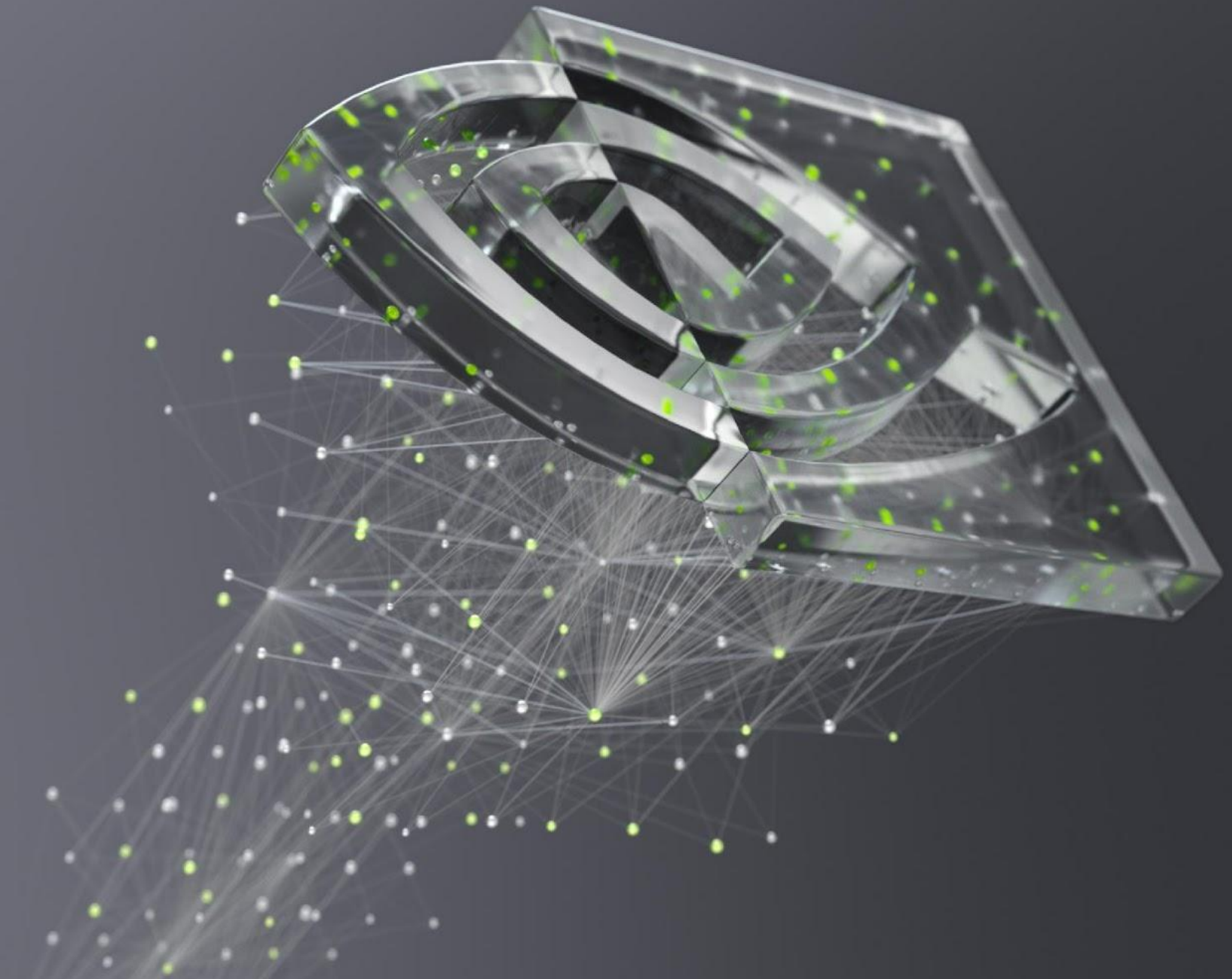
We'll present a comparative study of the impact of NVIDIA Merlin and GPU optimizations to the base case. The optimizations across the pipeline led to more than 80% reduction in cost, reduced latencies by over 100 miliseconds, and led to improved user engagement metrics. We'll focus on accelerating machine learning models with NVIDIA software and hardware, offer details of training and runtime optimizations like dynamic batching, precision optimizations, and XLA, and present the impact of these optimizations on latency, cost, and user metrics.

Aniruddha Zalani, Staff Data Scientist , ShareChat

Industry Segment: All Industries

Primary Topic: Recommenders / Personalization

ADD TO SCHEDULE **Thursday, March 24 | 8:00 AM - 8:50 AM PDT**



Thank You!

Even Oldridge
eoldridge@nvidia.com

Even_Oldridge

NVIDIA-Merlin

NVIDIA