SingleStore

The Future Starts Now

Achieving Successful Operation of ML
& AI-Driven Applications

Table of Contents

Introduction

In order to effectively compete and win in the next 10 years, every organization must become a data-driven organization. Leading companies in every category are using data collected and analyzed in real time to make better decisions, optimize the customer experience, and drive efficiency and profitability.

The old days of using analysis of past events to drive decisions is over, and the days of predicting outcomes and automating responses to events is here.

A recent report by Gartner describes AI and Machine Learning as being at the peak of hype in organizations today. Although these technologies are still emerging, they are already delivering practical benefits to help solve real-world problems. As enterprises adopt AI and machine learning, they are gaining significant momentum through their use of data.

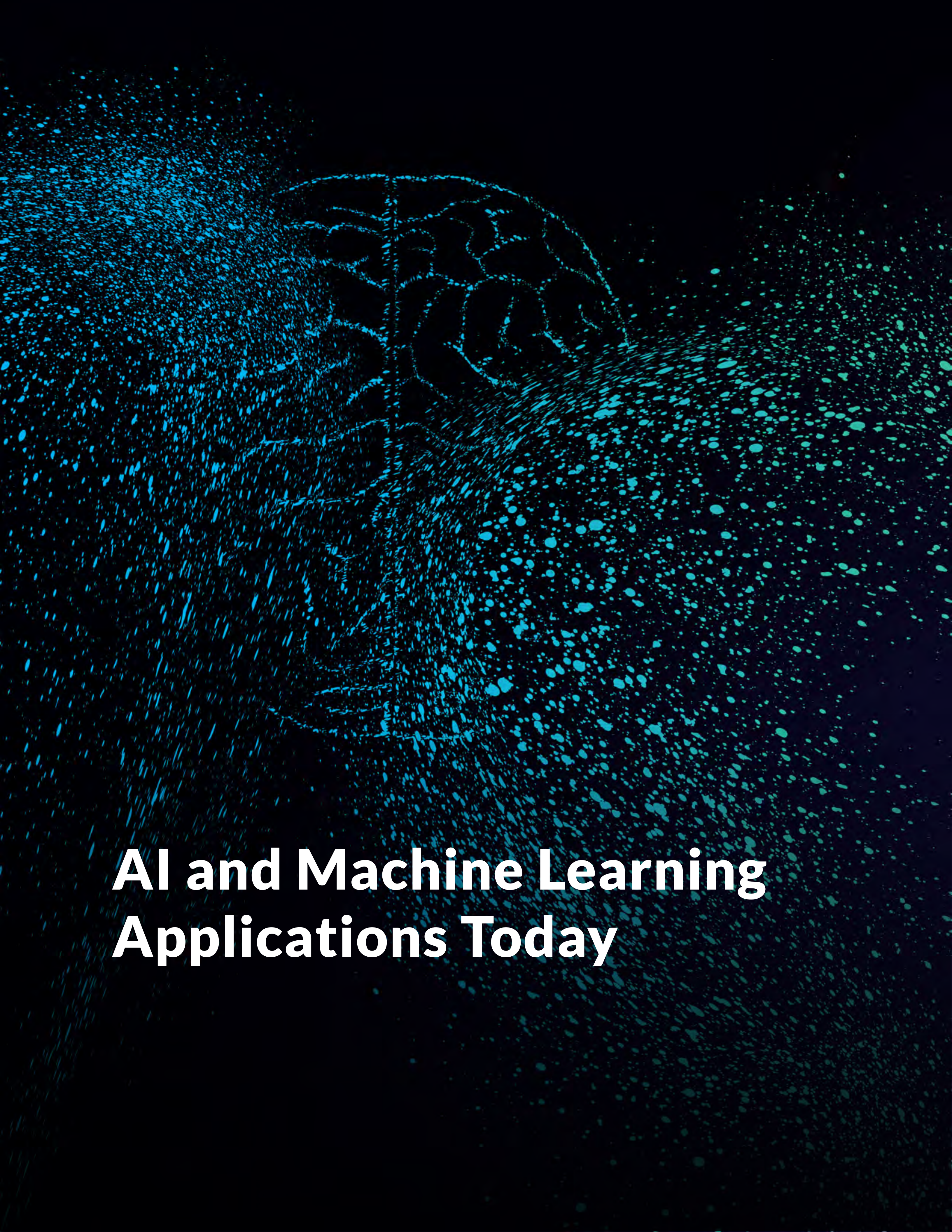
Models are crucial to this transition. A model is developed by identifying the rules that explain trends within historical data, then building data-driven applications to exploit these insights.

Models can transform how we work, how we make decisions, the products and services we provide, and how we serve customers.

In the next few years, machine learning and deep learning projects—both of which fit under the AI umbrella—will move out of the data science lab and into production. This whitepaper lays out a superior approach and architecture to both developing and deploying AI, machine learning, and predictive applications.

We in the tech industry have spent the last 20 years responding to global digital transformation by building millions of applications, each automating interactions through defined logic. With the onset of predictive models, machine learning, and deep learning, digital transformation has become a data revolution. With this change, we will spend the next 20 years rewriting all of those applications to be powered by models. With that tremendous change, a modern, cloud-native, ultra-fast, and highly scalable data architecture will be needed to bring this new enterprise to life.

And it all starts now. Continue reading to learn how to successfully deploy your model-driven applications into production.



AI and Machine Learning Applications Today

Clarifying Machine Learning vs. AI

There are many ways that artificial intelligence, machine learning, and model-driven applications can be defined. We define these terms as follows.

Artificial intelligence (AI) is the development of computer systems to accomplish tasks such as image recognition, facial recognition, speech recognition, decision-making, business predictions, etc. that normally require human intelligence to perform. Deep learning is a subset of machine learning (see below), and both are within the overall AI umbrella. Deep artificial neural networks train themselves to solve problems in areas such as sound recognition, recommendation systems, thumb impression detection, natural language processing, etc.

Machine learning (ML) is a technique whereby programs can iteratively learn from data, instead of being static in nature. ML functions are used to build models, based on input data, that can be used for predictions or decisions. Feeding in more data can also cause the model to improve itself, which generally improves the accuracy of results. Machine learning can be considered as a separate topic, but it is generally considered to be a subset of AI.

Hot research areas in AI include machine learning, cognitive science, image processing, deep learning, etc. For example, AI can manage manufacturing processes by enabling systems to think the way a human brain generally does. Machine learning is more about parsing data, updating models from input data, and applying models to make informed decisions. For example, an ML model can help media companies to learn the search criteria used by customers, and provide intelligent recommendations to enhance overall customer experience and expand the revenue pipeline.

Data Science Project Lifecycle

The data science life cycle is described by several different analytical models, including Microsoft's Team Data Science Process (TDSP). Several companies are following this popular approach (Fig.1) to structuring data science projects.

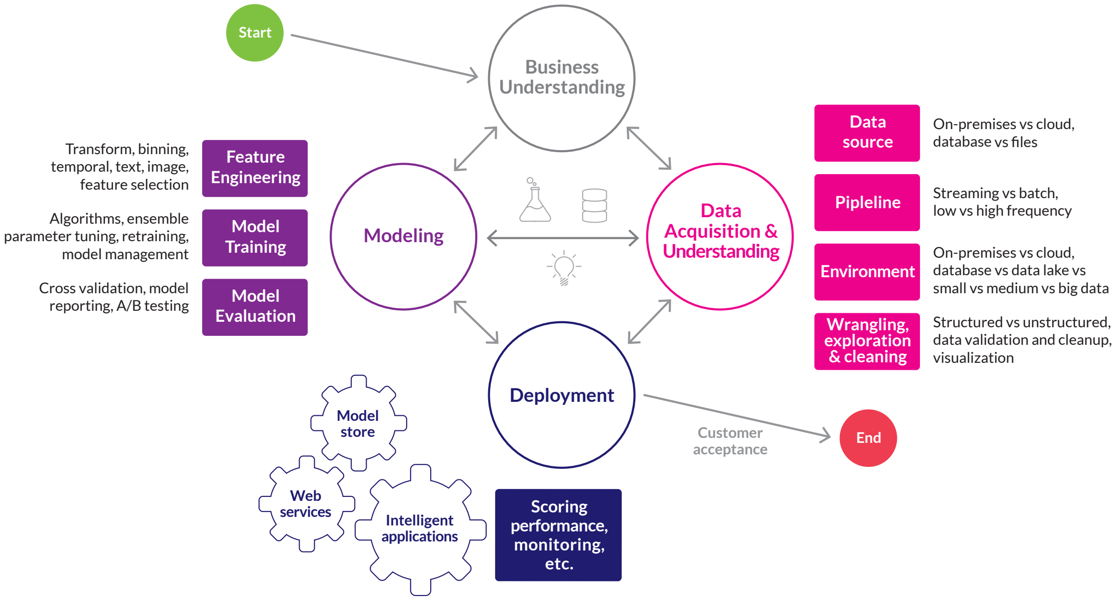

Start
Business Understanding
Transform, binning, temporal, text, image, feature selection
Feature Engineering
Algorithms, ensemble parameter tuning, retraining, model management
Model Training
Cross validation, model reporting, A/B testing
Model Evaluation
Modeling
Data Acquisition & Understanding
Data source
On-premises vs cloud, database vs files
Pipleline
Streaming vs batch, low vs high frequency
Environment
On-premises vs cloud, database vs data lake vs small vs medium vs big data
Wrangling, exploration & cleaning
Structured vs unstructured, data validation and cleanup, visualization
Deployment
Customer acceptance
End
Model store
Web services
Intelligent applications
Scoring performance, monitoring, etc.

Figure 1. Data science lifecycle

The three key pillars of the data science project lifecycle are:

- Data (collection, cleaning, transformation)
- Model (creation, evaluation)
- Operationalization (deployment, monitoring, retraining)

One of the most crucial aspects of this life cycle is operationalization of machine learning models. Operationalization refers to the deployment of models to be incorporated into business applications to predict the target class of a classification problem or the target value of a regression problem.

Operationalization of machine learning models includes the areas below:

- Deployment of models
- Model scoring/serving
- Monitoring model performance
- Collecting metrics/failures
- Analyzing results and errors
- Retraining models

Key Business Benefits

Enterprise companies have started investing heavily in automation by operationalizing AI/ML applications, in order to speed up the delivery of results and achieve critical business goals. Machine learning helps to improve tasks that are too difficult or too costly to do on a human scale.

Some of the key business benefits are below.

Real-time analytics

This approach leverages one of the greatest advantages of AI: the ability to process large sets of data and interpret the results in real time. Businesses can benefit from making the right decisions based on real-time insights, helping the company remains competitive in the market.

Pattern recognition and identification

This technique is another great use case of AI that can provide capabilities such as image processing and speech recognition, normally reserved to humans, by machines. For example, we can extract and process images to achieve facial recognition and identification.

Predictive analytics

AI/ML technologies are capable of handling large sets of data, identifying the patterns in the data set, and making predictions as to the future. For example, predictive analytics can help forecast which customer base has the highest probability of buying your product, then send targeted campaigns, discounts, coupons, etc. to the appropriate sets of people to expand the revenue pipeline. ML can also help in delivering highly efficient predictive maintenance plans for machines and connected devices.

Automation

AI is capable of automating service delivery and production operations in the manufacturing sector. For example, carmakers now use AI to manage and control the robotic lines in the factory assembly unit. In addition, AI is capable of monitoring supply and demand requirements in the warehouse, managing customer requests, payments, etc.

Fraud detection

Machine learning plays an instrumental role in solving some critical business requirements to prevent fraud such as email spam detection, unauthorized account access, and illicit bank transactions. Fraud detection has always been a challenging problem for the financial sector. As the number of credit/debit card transactions grows, the potential for hacking-related threats is rising at a tremendous level. ML-based automated fraud detection systems developed by data scientists have been successful in solving some complex problems and helping organizations to forestall fraud.

> *Organizations that want to thrive in world that is getting smarter at speed will be leveraging smart streaming that combines real time adaptable data combined with AI. SingleStore provides the adaptable data infrastructure that enables predictive/smarter automation and adaptable customer experiences.*
>
> **Jim Sinur, Independent Analyst**

Challenges in Delivering Real-Time AI/ML Applications

Key Challenges in Operationalizing AI and ML

The key challenges in creating and deploying real-time AI/ML applications are:

1. Data exploration & preparation

The data exploration and preparation phase deals with key operations such as exploratory analysis, fixing data inconsistencies, missing value treatment, noise removal, and more, within a given set of data. Exploratory Data Analysis (EDA) is generally used to understand and analyze the contents of a dataset for preparing more advanced modeling.

2. Algorithm selection, model training & refinement

The model development in machine learning happens in two stages. The first stage is learning and validation, in which we apply algorithms to data to uncover patterns between features and one or more target variables. The second stage is scoring, in which we apply trained models to a new dataset.

3. Assembling data for scoring

Scoring in ML is the process of applying an algorithm built from a historical dataset to a new dataset to unlock the insights that will help solve a business problem.

4. Scoring

Scoring can be categorized as (i) batch scoring and (ii) real-time scoring. Batch scoring is done when the model's decision is not time-sensitive. For example, collecting survey data and running a targeted campaign against it is an example of batch scoring. Real-time scoring is done when realizing the value from data is highly time-sensitive. For example, a credit card company needs to build a fraud detection model to score real-time credit card transactions within milliseconds, and take preventive actions (acceptance/denial) on the fly to safeguard customer accounts. This is a classic problem showcasing how businesses are leveraging operational analytics - "analytics with an SLA," as we call it here at SingleStore—using AI and ML functions.

Scoring is a key component that helps in choosing the most accurate ML models that can produce the most valuable insights. As a final step, the trained model will be deployed into production to address the specific business problem.

5. Model monitoring, refinement & retraining

The data science model lifecycle built by modern enterprises is continuous and iterative. This lifecycle may consist of a variety of different algorithms to be applied to data generated in real time. Modern business applications demand ML models built for real-time use to make data-driven decisions. Unfortunately, legacy database systems don't have the speed and scale capabilities to meet the requirements of real-time AI/ML applications. In addition, neural network training and massively parallel processing of deep learning will be too slow in a legacy data lake.

In order to solve these real-time challenges, the AI and ML applications need to be operationalized with the support of a database platform that's capable of handling highly concurrent workloads at unlimited speed and scale.

Making Your Data's Intelligence Work for You

Defining data intelligence

Just like humans, machines can do more with data when it's rich in associations, context, and connections. With computers, this is called metadata.

There are many forms of metadata, but the most important form, in data stored by all types of organizations around the world, is schema. Schema is most simply represented as the column headings in tabular data, though there should also be detailed definitions of the type of data in each column, and accompanying descriptions of data's provenance, and the processes used to verify its correctness.

JSON data, time series data (often stored in JSON format), and geospatial data also carry meaning in the arrangement of data, as well as in the actual values stored in data fields. Fig. 2, from JSON.org, shows the definition of a value in JSON.

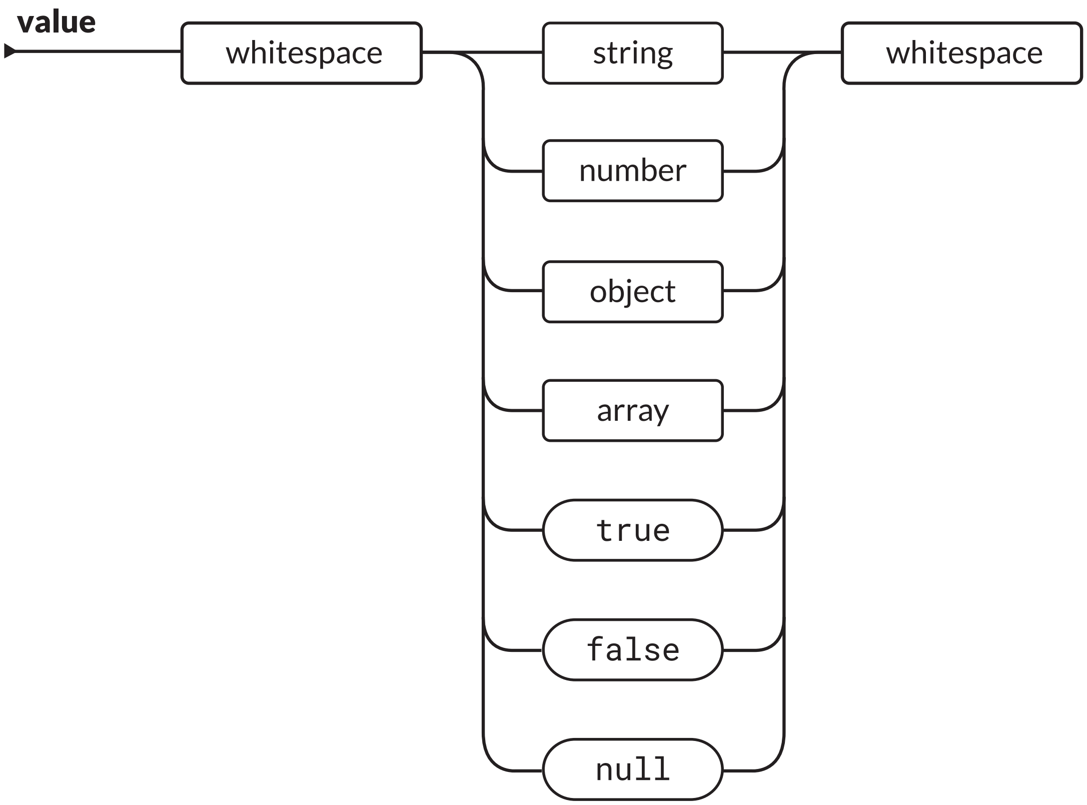


Figure 2. JSON data is semi-structured, with meaningful field descriptions

Graph databases see the relationships among data items as so important that the database is organized around these connections, which have discrete values assigned to them. Some platforms can convert graph data to relational data, and vice versa.

The NoSQL approach has been used to loosen strict control of data structure, so as to expedite the speed at which data can be ingested and written to persistent storage. The resulting data is in unstructured and semi-structured forms. As a result, many AI and machine learning programs have been designed to find associations among data values which have predictive power. (What those associations mean, in terms useful - and explainable - to humans, is not explained.) However, the downside of the NoSQL approach is that some of the data's meaning is lost through this relaxation.

Relational data, however, and other types of data with carefully managed schema and relationships, retains meaning. Well-designed AI and machine learning programs can maintain this meaning throughout the development process and into operationalization.

The intelligence you're generating out of the data using AI, machine learning, and deep learning capabilities can then be embedded into the business process to achieve positive business outcomes. AI requires well-managed data as input to truly add value.

In order to generate business value, results from distinct AI solutions, such as image recognition and speech generation, need to be joined together. This helps maximize the results from otherwise distinct solutions.

The importance of explainable AI

Explainable AI means AI and machine learning in which the operations and conclusions of models are explainable in terms understandable to humans. Conversely, if managers want the model to work differently, they can explain the changes they want in their own terms, and the model can easily be adjusted to operate in the manner they are describing.

Deep learning models, often used for image recognition, are particularly difficult to inspect to determine how a particular result is achieved. Deep learning models are trained by generating a neural network, which iteratively and recursively feeds outputs of one processing stage to the next, computing statistical correlations at every step to identify patterns in the data.

The trained models which are produced are mostly black boxes, with no external visibility into their workings, yet most people are happy to accept this, provided that useful and accurate results are produced. However, when biased or inequitable results are produced by these black boxes, the model and approach are quickly called into question. In many business contexts, if results can't be reliably explained, they're not usable.

Explainable AI is an important tool in using AI to solve practical problems. If the workings of models can be understood and adjusted by humans, the usefulness of these models is greatly increased. SingleStore is working closely with Fiddler.ai, a leader in explainable AI, to optimize the speed and functionality available in using the two platforms together as referenced in the detailed blog post: Explainable Churn Analysis with SingleStore and Fiddler.

Explainable AI is important in many, if not most, of the domains in which AI is used today, and will be used in the future. For instance, an essay from the Defense Advanced Research Projects Agency (DARPA) describes the importance of explainable AI in defense as follows:

> *Explainable AI—especially explainable machine learning—will be essential if future warfighters are to understand, appropriately trust, and effectively manage an emerging generation of artificially intelligent machine partners.*
>
> **See the article Explainable Artificial Intelligence (XAI), by Dr. Matt Turek, at darpa.mil.**

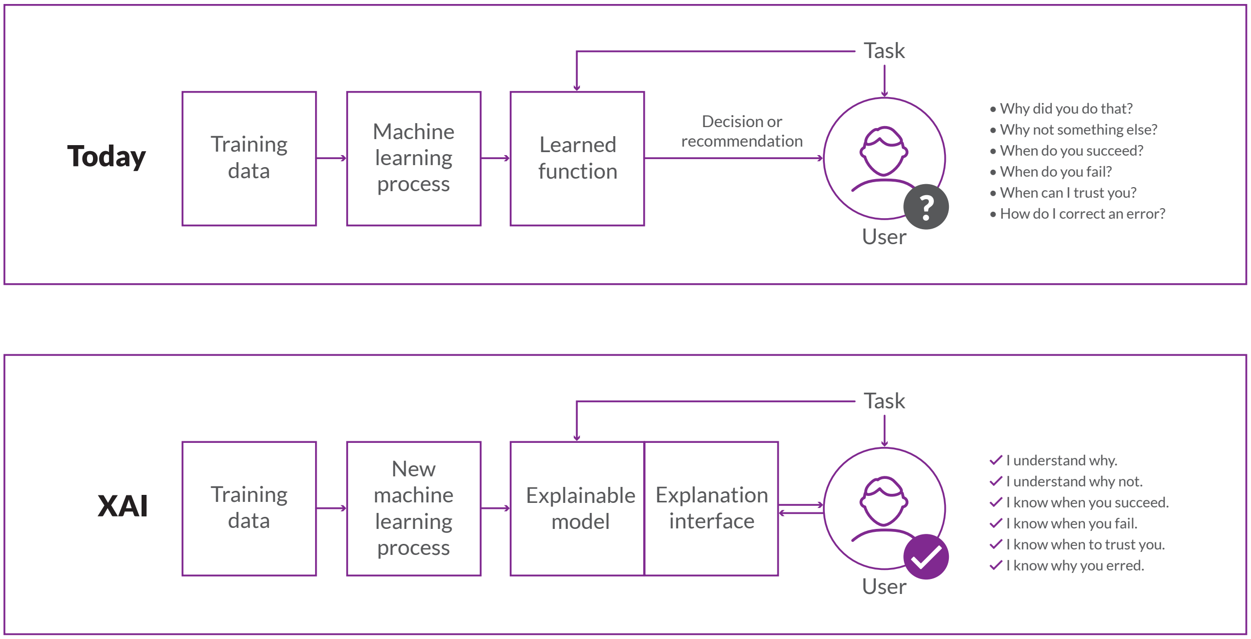


Figure 3. Explainable AI is more easily used, understood, and managed by people

With relational data, for example, data is usually gathered, checked, processed, and stored using carefully defined processes - and with closely associated metadata, in the form of schema. JSON data, incorporated in a relational database, maintains the schema that is so valuable for explainability, while incorporating flexibility that was formerly only possible with NoSQL databases. All of this information can be used at every stage of making the functionality and operations of AI explainable to the people using a system.

Transform Your Business by Operationalizing AI/ML Applications with SingleStore

SingleStore Highlights

The data infrastructure behind AI/ML applications requires a high-performance platform to perform the calculations involved in the model, usually across a mix of streaming and historical data. SingleStore was built to accomplish just that; a cloud-native, operational database built for speed and scale. As a distributed, highly scalable SQL database, SingleStore delivers maximum performance for both transactional and analytical workloads, using familiar relational data structures. SingleStore also comprises an ultra-fast ingest

and query platform that enables real-time model scoring on both streaming and historical data.

SingleStore is a great solution for storing training data and for performing real-time calculations involved in machine learning models. The SQL compatibility of SingleStore makes it a solid choice as a platform for data scientists and developers to use in developing complex algorithms for a variety of use cases.

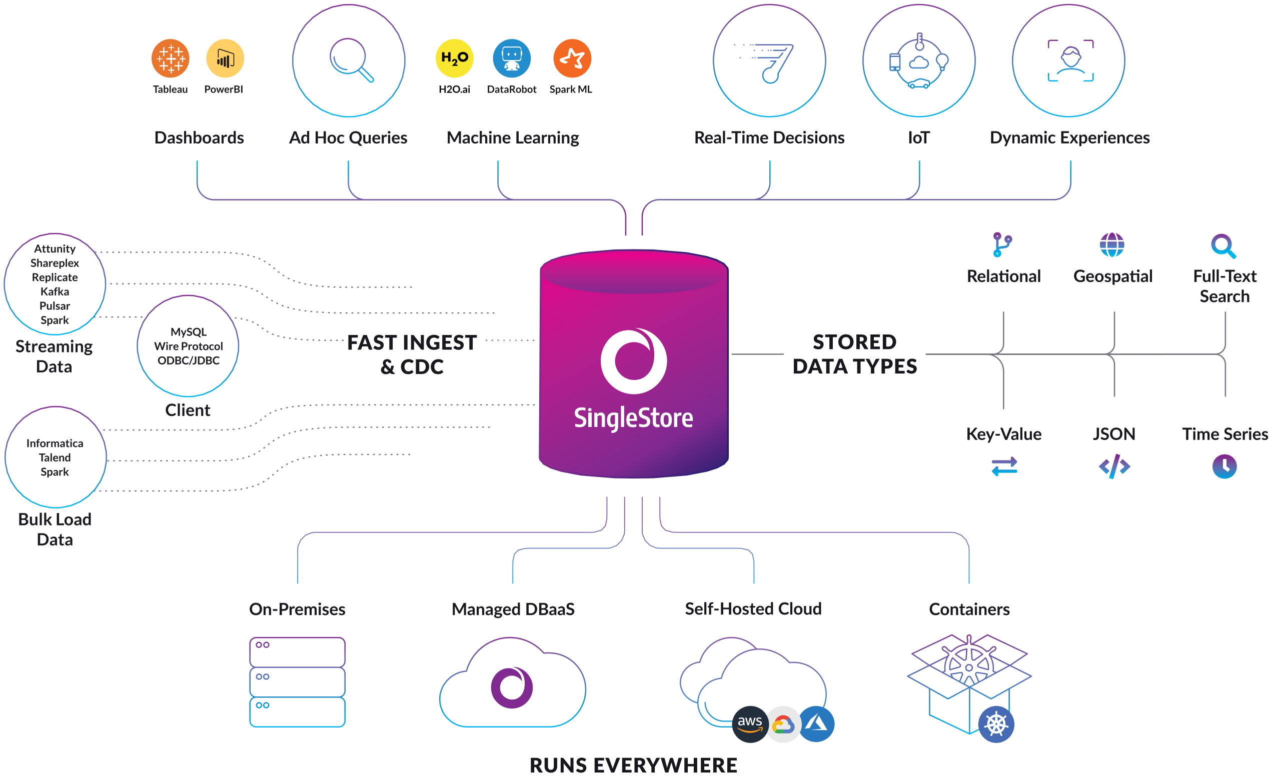


Figure 4. SingleStore architecture overview

Why SingleStore for Operationalizing AI/ML Applications

Once algorithms and models have been developed, they need to be deployed. The following key capabilities makes SingleStore a great solution to operationalize and scale AI/ML applications.

Fast, Scalable Data Exploration

SingleStore can centralize all operational data with built-in pipelines while performing ultra-fast queries to help in identifying new models. Key capabilities include:

- **Streaming ingest** - Eliminates the need for costly data integration and extract, transform, and load (ETL) tools through built-in batch and real-time data pipelines.
- **Columnar-powered analytics** - Fast petabyte-scale analytics with columnstore compression, delivering up to a 10x reduction in data size, and vectorized execution.
- **Compiled queries** - Repeat queries are compiled into low-level machine code to deliver record-breaking responsiveness for analytics queries.

Ultra-Fast Analysis and Scoring

SingleStore is capable of discovering anomalies or predicting events as they happen, by combining real-time and historical sources and querying them with standard SQL.

- **Fast query response** - Perform instant queries across millions of events or devices to discover real-time anomalies or predict events leveraging historical data using memory-optimized SQL.
- **Real-time scoring** - Deliver instant matching, applying complex models against big data datasets.
- **Built-in ML functions** - Apply built-in models or leverage extensibility functions to minimize response time.
- **Data pipeline scoring** - Score models as data is ingested to provide immediate response to changing conditions.

Proven Compatibility

SingleStore leverages the familiarity of ANSI SQL with full data persistence to drive sophisticated analytics while seamlessly working with existing business intelligence, data integration, and middleware tools.

- **Ingest** - SingleStore can handle streaming inputs from sources such as Kafka and Spark, as well as data sourced from databases such as AWS S3, Hadoop/HDFS, and data stored in files.
- **Analyze** - Perform ad hoc analysis with business intelligence tools like Tableau, Looker, Microstrategy and more.
- **Streaming data ingest and store** - The database can collect and store multiple streams of data into relational formats to support real-time and historical analysis. Ingestion often requires transactional operations - inserts, updates, and deletes - to ensure data accuracy.
- **Data science-compatible** - Integrate with model design tools like TensorFlow, Pandas, R, NumPy, SAS, SparkML, scikit-learn and more.

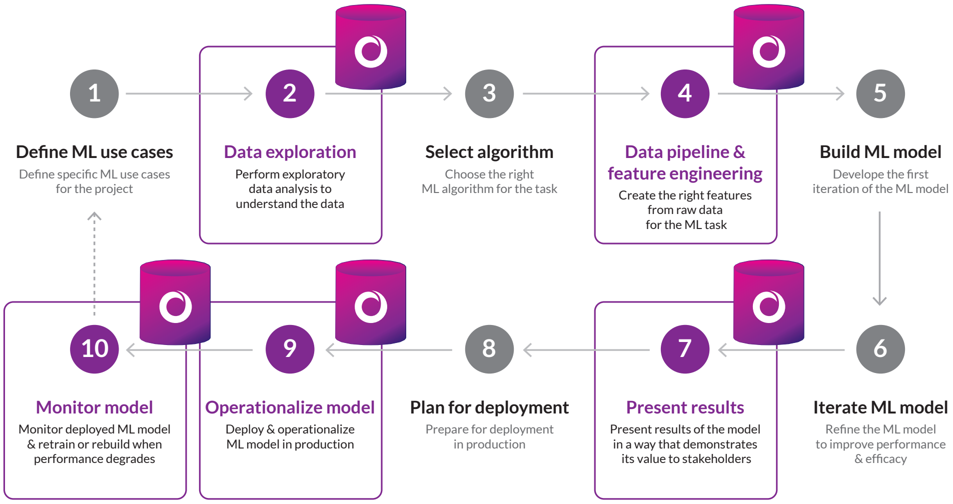
Unlimited Scale

SingleStore has a shared-nothing architecture for scale-out and scale-up, using standard hardware. It can be deployed on-premises, in the cloud, as a service, or as a hybrid.

- **Scalability and availability** - Utilize a modern, shared-nothing architecture to scale out with industry-standard hardware. Built-in resilience keeps the database online across cloud or on-premises deployments.
- **Distributed SQL** - Balance data and queries across a cluster of cloud instances or commodity hardware for maximum performance.
- **Resource governor** - Prevent unintended queries from consuming all available query execution memory or CPU in the cluster.
- **Memory-optimized** - Ultra-low latency for scalable transactions and analytics.

How SingleStore Drives Improvement across the ML Lifecycle

The following diagram shows how SingleStore can drive improvement across various stages in the ML lifecycle. Machine learning models are researched, built, assessed, and deployed.



1
Define ML use cases
Define specific ML use cases for the project
2
Data exploration
Perform exploratory data analysis to understand the data
3
Select algorithm
Choose the right ML algorithm for the task
4
Data pipeline & feature engineering
Create the right features from raw data for the ML task
5
Build ML model
Develope the first iteration of the ML model
6
Iterate ML model
Refine the ML model to improve performance & efficacy
7
Present results
Present results of the model in a way that demonstrates its value to stakeholders
8
Plan for deployment
Prepare for deployment in production
9
Operationalize model
Deploy & operationalize ML model in production
10
Monitor model
Monitor deployed ML model & retrain or rebuild when performance degrades

Figure 5. SingleStore drives improvement across the ML lifecycle

Areas where SingleStore can help directly are highlighted in purple:

2 **Data exploration.** Data stored in SingleStore can be explored faster than in other databases, and using standard SQL.

4 **Data pipeline & feature engineering.** These engineering tasks can require many iterations, so a fast database such as SingleStore can significantly shorten time to completion and allow for more approaches to be tried within a given timeframe.

7 **Present results.** This is another area where a high-performing database makes a difference in the success of the model, generating better results to share with stakeholders.

9 **Operationalize model.** This is where SingleStore really shines. Response time is absolutely critical in a model that's deployed to production. The ability to use a standard SQL database is also of the highest importance, as this way the same database can support all kinds of users.

10 **Monitor model.** A fast model supports monitoring as well as normal operations and makes it quicker and easier to retrain, rebuild, and re-deploy the model.

Operationalizing AI and ML with SingleStore

SingleStore offers the ideal architecture for AI and ML applications with an all-in-one database solution that can quickly model and score fast-changing data for real-time responsiveness to dynamic conditions driven by customers, machines, and 24/7 operations.

There are three different ways in which we can operationalize machine learning functions using SingleStore: **calculations outside the database**, **calculations on ingest**, and **calculations in the database**.

1. Calculations Outside the Database

SingleStore can be used as a high-performance solution to store raw data and deliver results to the customer. SingleStore can integrate AI and ML functions outside the database using Spark and TensorFlow. This integration can be achieved with the open source SingleStore Spark Connector. The connector opens up numerous ML functionalities that can be coupled with the scalable, high-performing, and durable SingleStore database.

Real-time machine learning scoring is a great use case for doing calculations outside the database. Predictive Model Markup Language (PMML), exported to a Spark cluster, does the real-time scoring of data. The scored data will land in SingleStore for further analysis, helping you to gain real-time insights on rapidly moving data.

The SingleStore PowerStream use case is a specific example of combining machine learning with a database to score data in real time and predict the likelihood of an equipment failure. This is an interactive demonstration simulating 200,000 wind turbines, sending sensor information at a rate of approximately 2 million inserts per second to SingleStore. From there, the user interface shows live status on real-time information, and ML scoring predicts the likelihood of turbine failures.

Read more about the SingleStore PowerStream use case in our blog post: PowerStream Wind Farm Analytics with Spark.

TensorFlow, and the results of applying machine learning models in TensorFlow, are shown in this video: Scoring Machine Learning Models at Scale from Strata New York.

2. Calculations on Ingest

A SingleStore Pipeline - a built-in streaming data carrier and ETL engine - can be used for scenarios where calculations need to be performed during data ingestion. SingleStore Pipelines are ideal for scenarios where data from a supported source must be ingested and processed in real time. This pipeline feature is also a good alternative to third-party middleware for basic ETL operations that must be executed as fast as possible. This also helps data engineers by eliminating the need for a separate cluster for calculations.

In addition, SingleStore Pipelines come with the option to execute customized transformations on extracted data prior to insertion into the target database. Your data engineering team can make use of this transform feature by executing ML algorithms on data as it arrives in the database. The massively parallel processing distributed system in SingleStore is smart enough to uniformly distribute the workload among the available resources in the cluster.

3. Calculations in the Database

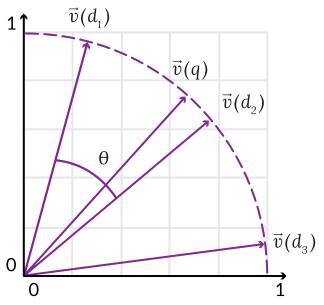
This method of ML integration happens within the database itself. This is a preferred approach when a customer wants to perform calculations closer to the data. In addition, the extensibility to leverage User Defined Functions (UDFs), User Defined Aggregates (UDAs), compiled code, scale-out performance, and stored procedures in a database are among the key capabilities. (SingleStore's Pipelines to stored procedures capability allows arbitrary code procedures to be run against data during ingest or after being stored, replacing ETL tools with a fast, flexible processing capability.)

Real-time facial recognition means taking a picture of a person and quickly performing a similarity match, to confirm or disconfirm their identity. Legacy systems take hours or days to process this information, making them useless for real-time and near-real-time applications, such as passport control workstations. With SingleStore, we can perform real-time image recognition matching within minutes or seconds.

Real-Time Facial Recognition with SingleStore

Facial recognition has recently emerged as a hot topic for research and implementation in the field of artificial intelligence. Today's data-driven world demands real-time image recognition for various different business scenarios. Our customers are actively using deep neural networks to classify images, and then using these networks to produce vector embeddings for images. The vectors produced this way are then inserted in the database or used in queries for matching. There are a growing collection of pre-trained deep neural nets available for developers to use for image recognition and other tasks, including VGG-face, facenet, etc. This can be achieved by using high-performance vector DOT_PRODUCT, which is a built-in function in SingleStore. Another built-in function called EUCLIDEAN_DISTANCE can also be used to efficiently measure the similarity between vectors.

SingleStore uses two different approaches in measuring similarity between vectors: (i) Cosine similarity (cosine of the angle between the vectors) and (ii) Euclidean distance. Cosine similarity is defined as the dot product of the vectors, divided by the product of the vector norms (length of the vectors). If the vectors are normalized, the cosine similarity is simply the dot product of the vectors (since the product of the norms is 1).



$$\left(\left| X, Y \right|\right) = \frac{x \cdot y}{||x|| \;\; ||y||}$$

Figure 6. Cosine similarity

Euclidean distance is also frequently used to measure similarity. It is defined as the norm of the vector resulting from the subtraction of two input vectors. The more similar two vectors are, the more likely that they represent the same target object - the same image, the same person's face, etc.

The feature vector must be extracted from the input image to store it in the database. For every image, we store an ID and a normalized FEATURE vector in a SingleStore table called features. A typical way to insert the vectors is to use Apache Spark, which enables quick parallel data transfer into SingleStore. Once those feature vectors are produced, all you need to do is insert them into a SingleStore table with the following simple schema:

```
CREATE TABLE features (
     id bigint(11) NOT NULL AUTO_INCREMENT,
     feature_vector binary(4096) DEFAULT NULL,
     KEY id (id) USING CLUSTERED COLUMNSTORE
                    );
```

To find similar images, we use this SQL query:

```
SELECT
     id, DOT_PRODUCT(feature, <input>) as score
FROM
     features
WHERE
     DOT_PRODUCT(feature, <input>) > 0.9
ORDER BY score DESC;
```

Input is a feature vector produced by a deep neural network for an incoming image, and 0.9 is a constant that was experimentally tuned, which corresponds to an angle of less than 26 degrees between the feature vector and the input - representing a likely match.

SingleStore is very fast for operations of this type, allowing image recognition to proceed through thousands of images in a small amount of time (for batch jobs) or within tight SLAs measured in milliseconds (for real-time applications).

Real-Time Image Recognition Workflow

Here's an example workflow for real-time image recognition:

- Use Spark, TensorFlow, and Gluon to train the model
- Use the trained model to extract feature vectors (embeddings) from images
- Store each of the feature vectors in a SingleStore table to use in performing real-time image recognition
- Application quickly matches the user-provided real-time images with reference images in the database and provides the scoring

Key use cases include face matching, photo matching, document similarity search, etc.

SingleStore deployments have proved in several use cases to deliver real-time image recognition at memory-bandwidth speed. Our existing customers are able to match a query image against millions of existing images in a fraction of a second.

Customer Reference Architectures

Epigen Powers Facial Recognition in the Cloud with SingleStore

Epigen Technology

Epigen Technology uses software to innovate in many fast-changing areas of technology, including cybersecurity, analytics, machine learning, artificial intelligence (AI), and the emerging area of cognitive computing. Now, Epigen has developed a framework for facial recognition in the cloud that combines AWS S3, SingleStore, machine learning, AI, and visualization.

Use case

Epigen's clients in areas such as the military, law enforcement, and air transportation need facial recognition that works under increasingly challenging circumstances, including working at scale and using video frame captures, not just static images.

Problem statement

How can facial recognition be made to work better with SingleStore, and to scale to handle more comparisons, at higher speeds, and to handle video with a strict SLA (10 ms response time)?

Reference Architecture

Epigen is developing a scalable architecture for facial recognition, which is a solid use case of AI.
A SingleStore-based data platform will serve as input to a cognitive computing application for facial recognition.

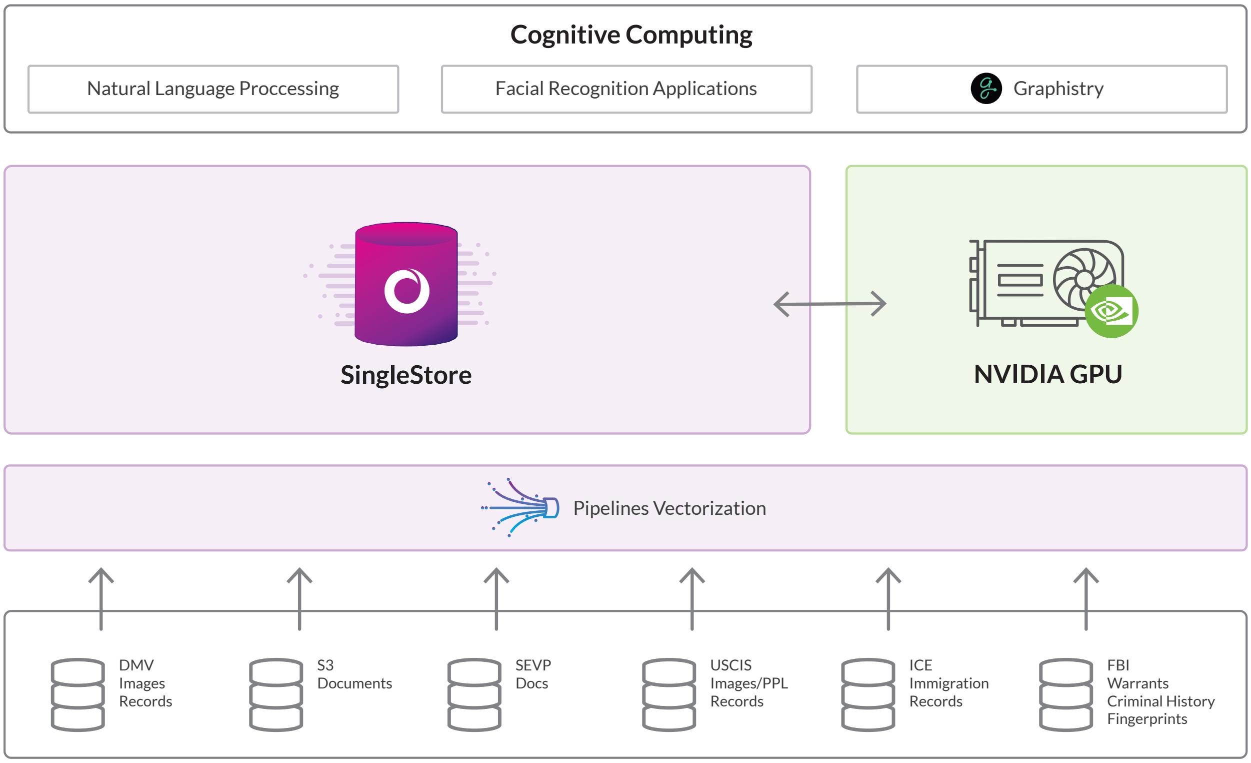


Figure 7. Reference architecture for facial recognition in AI

With this facial recognition application, Epigen is creating a reference architecture for cognitive computing. Data is initially read into an Amazon S3 database in the cloud. Epigen then uses a SingleStore Pipeline to extract the data from S3 at top speed, then rapidly transform and load it into SingleStore.

From there, the data goes to the GPU array. This design goes beyond many of today's AI and machine learning tools to make decisions and offer support for decision-making by humans.

For instance, a narrowly targeted AI tool might be able to show that someone boarding an airplane is not the same person as shown on their passport photo. With cognitive computing, the live photo would then be matched against law enforcement databases, with possible matches identified, and each possible match assigned a probability. A tool such as Graphistry would then be used to present a complete picture of the information uncovered to security personnel for them to act on.

What makes SingleStore a great solution for real-time facial recognition?

Epigen's facial recognition platform takes full advantage of several SingleStore capabilities to solve this business problem using AI, with significant cost savings:

- Rapid data ingest – Rapidly bringing in photographic data from Amazon S3 and, in the future, from other sources.
- Ultra-fast CPU-based processing – SingleStore's ability to handle the "transform" part of an ETL operation within a pipeline is crucial to rapid processing, as is SingleStore's Pipeline to stored procedures capability when more complex processing is required.
- Ultra-fast GPU-based processing – A GPU (Graphics Processing Unit) is a processor designed to handle graphics operations. This includes both 2D and 3D calculations, though GPUs were originally designed for rendering 3D graphics. The GPU is used by the application and in ML training. SingleStore does not run on the GPU. Instead, it helps in rapid transfer of processed data into and out of the GPUs, which is crucial to the performance of the entire system.
- Highly scalable - SingleStore was proven to be a scalable solution at every point in this process.

Next Steps with SingleStore

In the future, Epigen would like to investigate using SingleStore in the following use cases:

- Replacing a data warehouse implementation so as to improve fraud detection
- Preventing opioid abuse
- Geospatial applications
- Audio and voice detection
- Consolidating many disparate legacy database instances into a single data platform on SingleStore

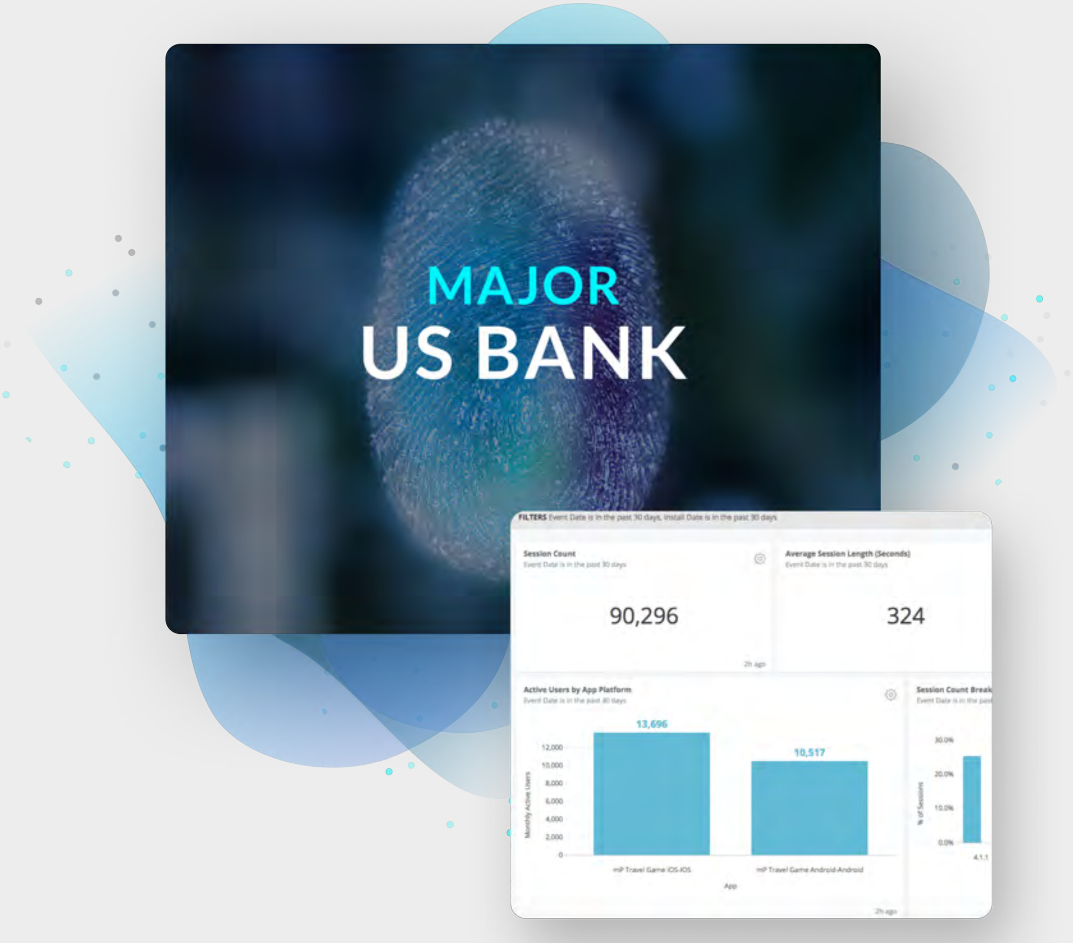
Please refer to this detailed case study for more details: Epigen Powers Facial Recognition in the Cloud with SingleStore.

Nyris.io also uses SingleStore to match product photos using deep learning. Nyris, a provider of visual product and object search technology, allows anyone to take a photo of an item - shoes, bikes, lamps - and Nyris will search its database to quickly return matches of the item for sale. Using SingleStore technology, Nyris scans images and finds matches in under a second.

Also Thorn, a non-profit organization, has developed applications based on SingleStore that perform image recognition to identify missing and exploited children.

Please refer to this blog post to learn more about how SingleStore makes real-time image recognition a reality: Case Study: Thorn Frees Up Resources with SingleStore Managed Service to Identify Trafficked Children Faster.

Major US Bank Using SingleStore for Real-Time Fraud Detection


MAJOR
US BANK
FILTERS Event Date is in the past 30 days, Install Date is in the past 30 days
Session Count
Event Date is in the past 30 days
90,296
2h ago
Average Session Length (Seconds)
Event Date is in the past 30 days
324
Active Users by App Platform
Event Date is in the past 30 days
13,696
10,517
12,000
10,000
8,000
6,000
4,000
2,000
0
Monthly Active Users
mP Travel Game iOS-iOS
mP Travel Game Android-Android
App
2h ago
Session Count Break
Event Date is in the past
30.0%
20.0%
10.0%
0.0%
% of Sessions
4.1.1

Use Case

This is a solid use case of operationalizing machine learning with SingleStore. This reference architecture (Fig. 8, appears on the next page) shows the capability of SingleStore to deliver real-time fraud detection and enhance customer experience for a top-ranked financial institution in the US.

Problem Statement

A scalable data platform to be built for real-time credit card fraud detection, with the following strict requirements as part of the SLA:

- One-second total communications and processing budget, from the time the card is swiped to approval or denial
- 50 ms budget to run queries that generate a 70-field feature record to score
- Complete the fraud detection within one second

Old method:

Run nightly batch job to accumulate feature record for each customer in a traditional operational data store (ODS), then look it up for scoring to identify fraudulent transactions after the fact. The old approach may take up to a few business days to identify any fraud transaction, leaving customers in a potentially risky situation, with a poor customer experience.

New method (with SingleStore):

In parallel, run up to 70 queries concurrently (one per feature or few features) on the latest data to get features in real time (<50 ms total), and return the results to the operational data store, enabling the fraud detection to be completed within one second. This new approach with SingleStore can deliver instantaneous decision-making capability against any fraudulent transactions by taking necessary action to block or prevent any illegal transaction. Frequently, credit card fraud is a part of a broader theft of your identity and your personal information. Hence this new method achieves superior customer experience and enhanced security for credit card transactions.

Reference Architecture for ML-Based Real-time Fraud Detection with SingleStore

The workflow defined in the reference architecture diagram (Fig.6) starts with a user request hitting an on-line transaction processing (OLTP) application, followed by this event being stored in the bank's OLTP database at the backend. The request is then converted into 70 different queries which are run in parallel against the operational database, SingleStore, to validate key factors.

These include (i) Engagement trends between customers and specific merchants for the past few days, weeks, and months, (ii) Geographical location at which the transaction takes place, compared with the registered address of the customer, (iii) Transaction trend patterns between merchants and customers etc., all to identify whether an event is fraudulent or not.

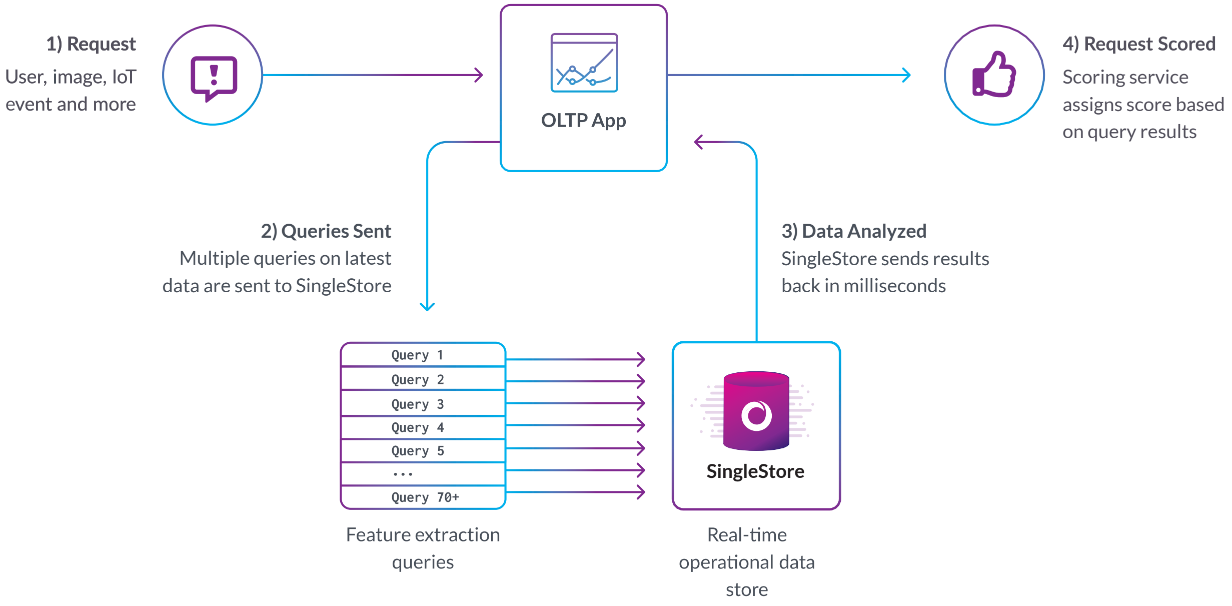

1) Request
User, image, IoT event and more
OLTP App
4) Request Scored
Scoring service assigns score based on query results
2) Queries Sent
Multiple queries on latest data are sent to SingleStore
3) Data Analyzed
SingleStore sends results back in milliseconds
Query 1
Query 2
Query 3
Query 4
Query 5
...
Query 70+
SingleStore
Feature extraction queries
Real-time operational data store

Figure 8. Reference architecture for real-time fraud detection

The data is analyzed in SingleStore using the sophisticated machine learning model built for real-time scoring to decide transaction approval or denial. With SingleStore in place, this bank was able to perform the scoring service within a 50 ms. window and complete the fraud detection cycle for each event within a second.

Here are some of the key features of SingleStore that helped to solve this business problem:

- ✓ Shared-nothing, massively parallel processing architecture
- ✓ High concurrency
- ✓ Performance at scale
- ✓ Ultra-fast data ingestion
- ✓ Query performance in a window of milliseconds

Not only the speed, but also the quality, of fraud detection is improved by using a real-time feature record, rather than one generated the night before. For example, you can have a feature like "number of purchases by customer at this vendor in the previous two hours," which is not feasible with the old-fashioned feature record created the night before. Faster responsiveness can help identify fraud and reduce false positives as well, giving reduced losses while improving customer experience for the vast majority of transactions that are non-fraudulent. And the fact that fraud can be detected in real time reduces the opportunity for multiple fraudulent transactions to be carried out before a pattern can be detected.

Please refer to the detailed case study for more details: Fraud Detection "On the Swipe" For a Major US Bank.

Large Energy Company Using SingleStore for Preventive Maintenance

Use Case

A large energy company is using SingleStore for upstream processing, which is a solid use case of predictive analytics in a machine learning application. The company delivers preventive maintenance for their oil drills across the globe using this data platform. The preventive maintenance approach helps them to correct the root cause of detected failure and avoid breakdowns to their equipment.

Problem Statement

Design a highly scalable data platform which is capable of continuously ingesting raw data from a variety of sensors and allowing predictive machine models to run in parallel to determine the scoring for oil drill health, all in real time.

Reference Architecture for ML-based Preventive Maintenance

Oil drills are deployed across different locations around the globe. These drills are equipped with various sensors (such as vibration, direction, and temperature) and generate data continuously. The raw data generated gets pushed into a Kafka queue as a first step. Data is then pulled from the Kafka queue into a Spark cluster, where a Predictive Model Markup Language (PMML) model calculates the health of a drill based on real-time data. The scored data then lands in SingleStore to power a real-time operational dashboard, where the drill operators can make informed judgments based on events or anomalies detected by processing real-time data.

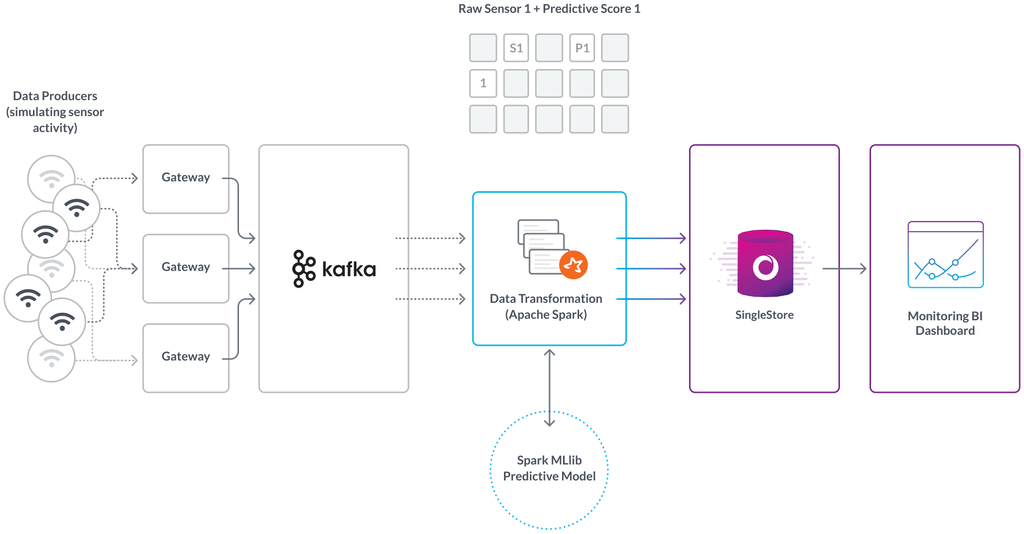

Raw Sensor 1 + Predictive Score 1
S1
P1
1
Data Producers (simulating sensor activity)
Gateway
Gateway
Gateway
kafka
Data Transformation (Apache Spark)
SingleStore
Monitoring BI Dashboard
Spark MLlib Predictive Model

Figure 9. Reference architecture for predictive maintenance

Key considerations for using SingleStore included:

- Can query while you ingest data
- Highly scalable
- Ultra-fast ingestion
- Massively parallel processing data architecture
- Perform ML scoring and data ingestion in parallel

Choosing the Right AI/ML Database

SingleStore is an operational database
built for speed and scale

Operationalizing AI and ML has become an unavoidable need in business, as various industries heavily rely on large volumes of real-time data as input to automated decision-making processes to yield the best results. Use cases in the data science field have shown that ML models and AI have few tangible business benefits until they are operationalized.

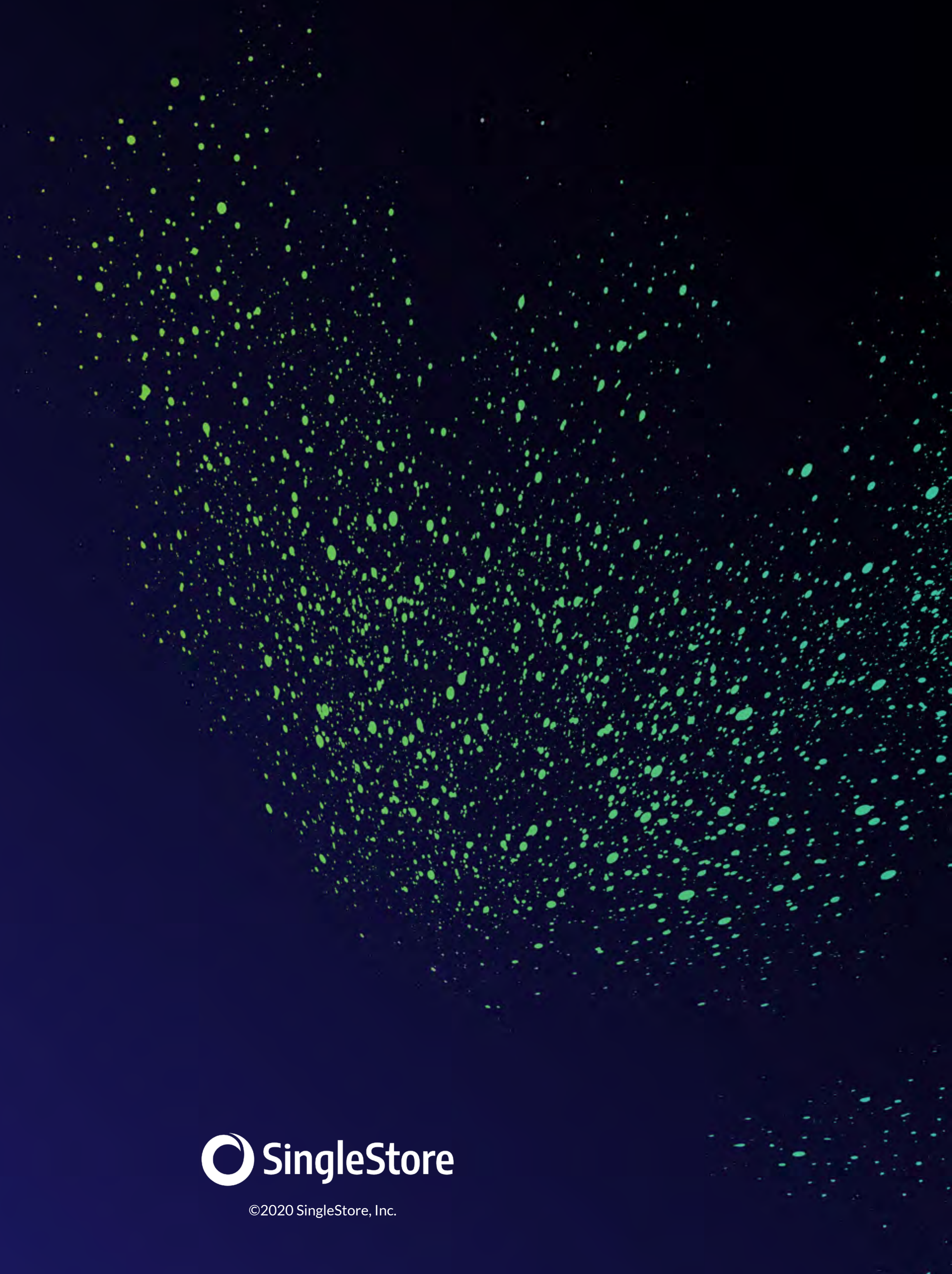
A distributed, scalable architecture, having the capability to do massively parallel processing, is one of the key requirements for operationalizing AI and ML. The use cases may demand both real-time and batch processing for model scoring. Hence the scoring engine used should be capable enough to support deep learning, TensorFlow, neural network architectures, custom-built models, etc.

SingleStore, as an operational database built for speed and scale, is likely to meet all relevant requirements, demanding as they are. Therefore, SingleStore can be considered as an excellent database solution for operationalizing AI-based applications in general, and machine learning models in particular.

In addition, since SingleStore can also be deployed using Docker or Kubernetes, a scoring engine based on SingleStore is truly cloud-native, and can efficiently leverage the lightweight, portable, and scalable features of containerization. SingleStore is a breakthrough in database architecture, allowing transactional and analytical workloads to be processed using a single table type. With SingleStore, and the new system of record improvements, our vision of "one database to rule them all" begins to be realized. And SingleStore's database-as-a-service offering, SingleStore Managed Service, gives you the full capabilities of SingleStore, including the new system of record capabilities, without the operational overhead and complexity of managing it all yourself. The real-world examples discussed above justify consideration of SingleStore as a solid database engine to support and simplify the operationalization of predictive analytics, ML models, and AI applications for an enterprise.

Written by
Nithin Krishna Reghunathan, Technical Evangelist, SingleStore

Nithin has proven expertise in designing and implementing solutions to solve challenging problems in Big Data, including use cases in machine learning, AI, IoT, etc. His prior experience includes work for Fortune 50 global tech companies.



SingleStore