Building Media & Entertainment Predictive Analytics Solutions on AWS

First published December 2016

Updated March 30, 2021

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

This whitepaper is intended for data scientists, data architects, and data engineers who want to design and build Media and Entertainment (M&E) predictive analytics solutions on AWS. Specifically, this paper provides an introduction to common cloud-enabled M&E workloads, and describes how a predictive analytics workload fits into the overall M&E workflows in the cloud.

The paper provides an overview of the main phases for the predictive analytics business process, as well as an overview of common M&E predictive analytics use cases. Then, the paper describes the technical reference architecture and tool options for implementing predictive analytics solutions on AWS.

Introduction

The world of Media and Entertainment (M&E) has shifted from treating customers as mass audiences to forming connections with individuals. This progression was enabled by unlocking insights from data generated through new distribution platforms and web and social networks. M&E companies are aggressively moving from a traditional, mass-broadcasting business model to an Over-The-Top (OTT) model where relevant data can be gathered. In this new model, they are embracing the challenge of acquiring, enriching, and retaining customers through big data and predictive analytics solutions.

As cloud technology adoption becomes mainstream, M&E companies are moving many analytics workloads to AWS to achieve agility, scale, lower cost, rapid innovation, and operational efficiency. As these companies start their journey to the cloud, they have questions about common M&E use cases and how to design, build, and operate these solutions. AWS provides many services in the data and analytics space that are well suited for all M&E analytics workloads, including traditional BI reporting, real-time analytics, and predictive analytics.

In this paper, we discuss the approach to architecture and tools. We'll cover design, build, and operate aspects of predictive analytics in subsequent papers.

Overview of AWS Enabled M&E Workloads

M&E content producers have traditionally relied heavily on systems located on-premises for production and post-production workloads. Content producers are increasingly looking into the AWS Cloud to run workloads. This is due to the huge increase in the volume of content from new business models, such as on-demand and other online delivery, as well as new content formats such as 4k and high dynamic range (HDR).

M&E customers deliver live, linear, on-demand, and OTT content with the AWS Cloud. AWS services also enable media partners to build solutions across M&E lines of business. Examples include:

- Managing digital assets
- Publishing digital content
- Automating media supply chains
- Broadcast master control and play out

- Streamlining content distribution to licensees
- Affiliates (business to business - B2B)
- Direct to consumer (business to consumer - B2C) channels
- Solutions for content and customer analytics using real-time data and machine learning

Figure 1 is a diagram that shows a typical M&E workflow, with a brief description of each area.

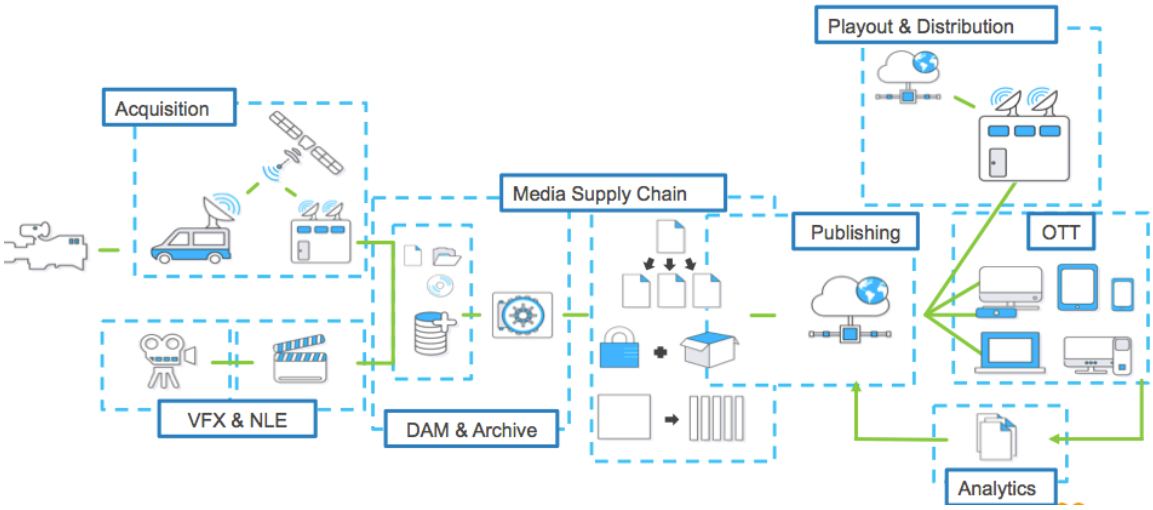

Figure 1 — Cloud-enabled M&E workflow

Acquisition — Workloads that capture and ingest media contents such as videos, audio, and images into AWS.

VFX & NLE — Visual Effects (VFX) and nonlinear editing system (NLE) workloads that allow editing of images for visual effects or nondestructive editing of video and audio source files.

DAM & Archive — Digital asset management (DAM) and archive solutions for the management of media assets.

Media Supply Chain — Workloads that manage the process to deliver digital assets such as video or music from the point of origin to the destination.

Publishing — Solutions for media contents publishing.

OTT — Systems that allow the delivery of audio content and video content over the Internet.

Playout & Distribution — Systems that support the transmission of media contents and channels into the broadcast network.

Analytics — Solutions that provide business intelligence and predictive analytics capabilities on M&E data. Some typical domain questions to be answered by the analytics solutions are: How do I segment my customers for email campaign? What videos should I be promoting at the top of audiences OTT/VOD watchlists? Who is at risk of cancelling a subscription? What ads can I target mid-roll to maximize audience engagement? What is the aggregate trending sentiment regarding titles, brands, properties, and talents across social media and where is it headed?

Overview of the Predictive Analytics Process Flow

There are two main categories of analytics: business and predictive. Business analytics focus on reporting metrics for historical and real-time data. Predictive analytics help predict future events and provide estimations by applying predictive modeling that is based on historical and real-time data. This paper will only cover predictive analytics.

A predictive analytics initiative involves many phases and is a highly iterative process. Figure 2 shows some of the main phases in a predictive analytics project with a brief description of each phase.

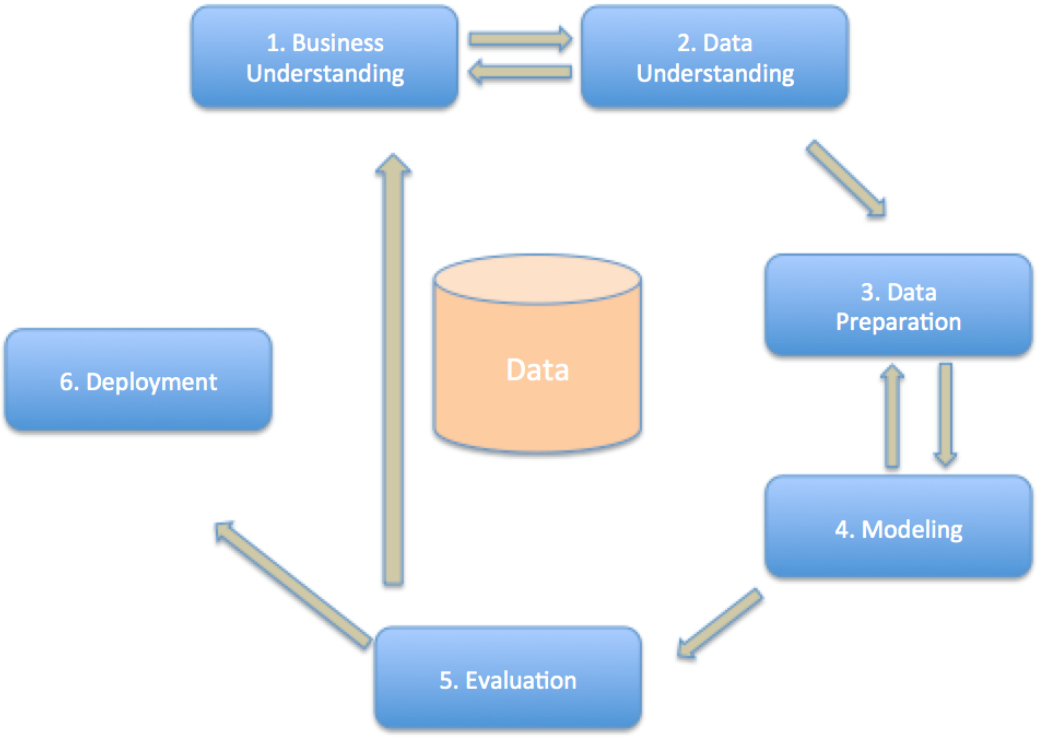

Figure 2 — Cross-industry standards process for data mining

1. **Business Understanding** — The main objective of this phase is to develop an understanding of the business goals, and then translate the goals into predictive analytics objectives. For the M&E industry, examples of business goals could include increasing content consumption by existing customers, or understanding social sentiment toward contents and talents to assist with new content development. The associated predictive analytics goals could also include personalized content recommendations, and sentiment analysis of social data regarding contents and talents.
2. **Data Understanding** — The goal of this phase is to consider the data required for predictive analytics. Initial data collection, exploration, and quality assessment take place during this phase. To develop high-quality models, the dataset needs to be relevant, complete, and large enough to support model training. Model training is the process of training a machine learning model by providing a machine learning algorithm with training data to learn from. Some relevant datasets for M&E use cases are customer information/profile data, content viewing history data, content rating data, social engagement data, customer behavioral data, content subscription data, and purchase data.

3. **Data Preparation** — Data preparation is a critical step to ensure that high-quality predictive models can be generated. In this phase, the data required for each modeling process is selected. Data acquisition mechanisms need to be created. Data is integrated, formatted, transformed, and enriched for the modeling purpose. Supervised machine learning algorithms require a labeled training dataset to generate predictive models. A labeled training dataset has a target prediction variable and other dependent data attributes or features. The quality of the training data is often considered more important than the machine learning algorithms for performance improvement.
4. **Modeling** — In this phase, the appropriate modeling techniques are selected for different modeling and business objectives. For example:
    - A clustering technique could be employed for customer segmentation.
    - A binary classification technique could be used to analyze customer churn.
    - A collaborative filtering technique could be applied to content recommendation.

    The performance of the model can be evaluated and tweaked using technical measures such as Area-Under-Curve (AUC) for binary classification (Logistic Regression), Root-Mean-Square (RMSE) for collaborative filtering (Alternating Least Squares), and Sum-of-Squared Error (SSE) for clustering (K-Means). Based on the initial evaluation of the model result, the model settings can be revised and fine-tuned by going back to the data preparation stage.
5. **Model Evaluation** — The generated models are formally evaluated in this phase not only in terms of technical measures, but also in the context of the business-success criteria set out during the business-understanding phase. If the model properly addresses the initial business objectives, it can be approved and prepared for deployment.
6. **Deployment** — In this phase, the model is deployed into an environment to generate predictions for future events.

Common M&E Predictive Analytics Use Cases

To a certain extent, some of the predictive analytics use cases for the M&E industry do not differ much from other industries. The following are common use cases that apply to the M&E industry.

Customer segmentation — As the engagement between customers and M&E companies becomes more direct across different channels, and as more data is collected on those engagements, appropriate segmentation of customers becomes increasingly important. Customer relationship management (CRM) strategies, including customer acquisition, customer development, and customer retention, greatly rely upon such segmentation. Although customer segmentation can be achieved using basic business rules, it can only efficiently handle a few attributes and dimensions. A data-driven segmentation with a predictive modeling approach is more objective and can handle more complex datasets and volumes.

Customer segmentation solutions can be implemented by leveraging clustering algorithms such as k-means, which is a type of unsupervised learning algorithm. A clustering algorithm is used to find natural clusters of customers based on a list of attributes from the raw customer data.

Content recommendation — One of the most widely adopted predictive analytics by M&E companies, this type of analytics is an important technique to maintain customer engagement and increase content consumption. Due to the huge volume of available content, customers need to be guided to the content they might find most interesting. Two common algorithms leveraged in recommendation solutions are content-based filtering and collaborative filtering.

- **Content-based filtering** is based on how similar a particular item is to other items, based on usage and rating. The model uses the content attributes of items (categories, tags, descriptions, and other data) to generate a matrix of each item to other items, and calculates similarity based on the ratings provided. Then the most similar items are listed together with a similarity score. Items with the highest score are most similar.

- **Collaborative filtering** is based on making predictions to find a specific item or user, based on similarity with other items or users. The filter applies weights based on peer user preferences. The assumption is users who display similar profile or behavior have similar preferences for items.

More advanced recommendation solutions can leverage deep learning techniques for better performance. One example of this would be using Recurrent Neural Networks (RNN) with collaborative filtering by predicting the sequence of items in previous streams, such as past purchases.

Sentiment analysis — This is the process of categorizing words, phrases, and other contextual information into subjective feelings. A common outcome for sentiment analysis is positive, negative, or neutral sentiment. Impressions publicized by consumers can be a valuable source of insight into the opinions of broader audiences. These insights, when employed in real time, can be used to significantly enhance audience engagement. Insights can also be used with other analytic learnings such as customer segmentation to identify a positive match between an audience segment and associated content. There are many tools to analyze and identify sentiment, and many of them rely on linguistic analysis that is optimized for a specific context.

From a machine learning perspective, one traditional approach is to consider sentiment analysis as a classification problem. The sentiment of a document, sentence, or word is classified with positive, negative, or neutral labels. In general, the algorithm consists of tokenization of the text, feature extraction, and classification using different classifiers such as linear classifiers (e.g., Support Vector Machine, Logistic Regression) or probabilistic classifiers (e.g., Naïve Bayes, Bayesian Network). However, this traditional approach lacks recognition for the structure and subtleties of written language.

A more advanced approach is to use deep learning algorithms for sentiment analysis. You don’t need to provide these models with predefined features, as the model can learn sophisticated features from the dataset. The words are represented in highly dimensional vectors and features are extracted by the neural network. Examples of deep learning algorithms that can be used for sentiment analysis are Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN). MXNet, Tensorflow, and Caffe are some deep learning frameworks that are well suited for RNN and CNN model training. AWS makes it easy to get started with these frameworks by providing an Amazon Machine Image (AMI) that includes these frameworks preinstalled. This AMI can be run on a large number of instance types, including the P2 instances that provide general

purpose GPU processing for deep learning applications. The Deep Learning AMI is available in the AWS Marketplace.

Churn prediction — This is the identification of customers who are at risk of no longer being customers. Churn prediction helps to identify where to deploy retention resources most effectively. The data used in churn prediction is generally user activity data related to a specific service or content offering. This type of analysis is generally solved using a logistic regression with a binary classification. The binary classification is designated as customer leave predicted or customer retention predicted. Weightings and cutoff values can be used with predictive models to tweak the sensitivity of predictions to minimize false positives or false negatives to optimize for business objectives. For example Amazon Machine Learning (Amazon ML) has an input for cutoff and sliders for precision, recall, false positive rate, and accuracy.

Predictive Analytics Architecture on AWS

AWS includes the components needed to enable pipelines for predictive analytics workflows. There are many viable architectural patterns to effectively compute predictive analytics. In this section, we discuss some of the technology options for building predictive analytics architecture on AWS. Figure 3 shows one such conceptual architecture.

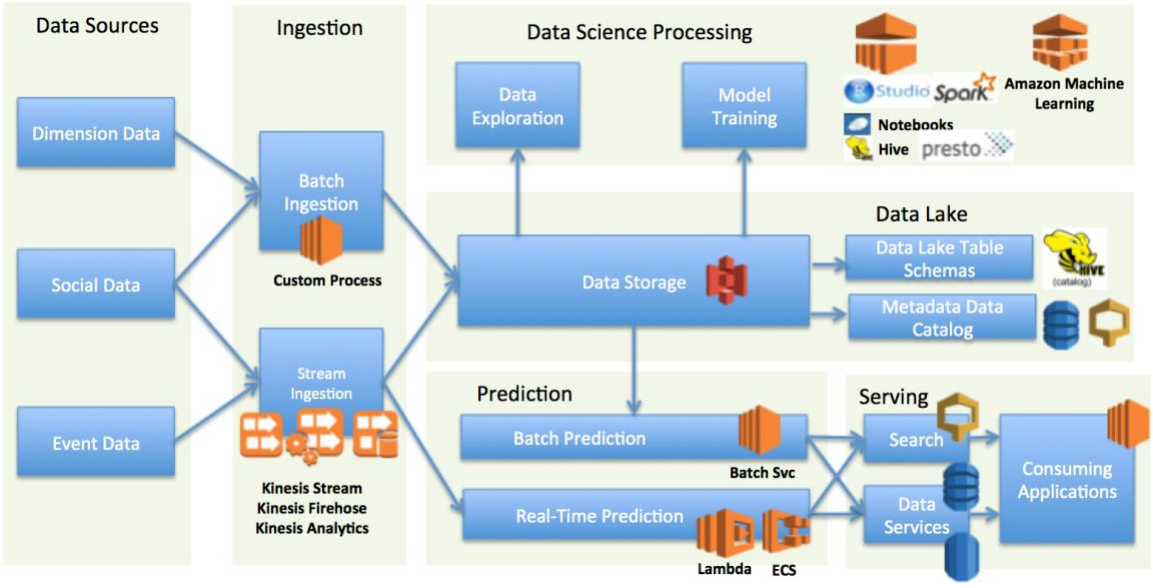

Figure 3 — Conceptual reference architecture

Data Sources and Data Ingestion

Data collection and ingestion is the first step and one of the most important technical architecture components to the overall predictive analytics architecture. At a high level, the main source data required for M&E analytics can be classified into the following categories.

- **Dimension data** — Provides structured labeling information to numeric measures. Dimension data is mainly used for grouping, filtering, and labeling of information. Examples of dimension data are customer master data, demographics data, transaction or subscription data, content metadata, and other reference data. These are mostly structured data stored in relational databases, such as CRM, Master Data Management (MDM), or Digital Asset Management (DAM) databases.
- **Social media data** — Can be used for sentiment analysis. Some of the main social data sources for M&E are Twitter, YouTube, and Facebook. The data could encompass content ratings, reviews, social sharing, tagging, and bookmarking events.
- **Event data** — In OTT and online media, examples of event data are audience engagement behaviors with streaming videos such as web browsing patterns, searching events for content, video play/watch/stop events, and device data. These are mostly real-time click-streaming data from websites, mobile apps, and OTT players.
- **Other relevant data** — Includes data from aggregators (Nielson, comScore, etc.), advertising response data, customer contacts, and service case data.

There are two main modes of data ingestion into AWS: batch and streaming.

Batch Ingestion

In this mode, data is ingested as files (e.g., database extracts) following a specified schedule. Data ingestion approaches include the following.

- **Third-party applications** — These applications have connector integration with Amazon Simple Storage Service (Amazon S3) object storage that can ingest data into Amazon S3 buckets. The applications can either take source files or extract data from the source database directly and store them in Amazon S3. There are commercial products (e.g., Informatica, Talend) and open-source utilities (e.g., Embulk) that can extract data from databases and export the data into an Amazon S3 bucket directly.
- **Custom applications using AWS SDK/APIs** — Custom applications can use AWS SDKs and the Amazon S3 application programming interface (API) to ingest data into target Amazon S3 buckets. The SDKs and API also support multipart upload for faster data transfer to Amazon S3 buckets.
- **AWS Data Pipeline** — This service facilitates moving data between different sources, including AWS services. AWS Data Pipeline launches a task runner that is a Linux-based Amazon Elastic Compute Cloud (Amazon EC2) instance, which can run scripts and commands to move data on an event-based or scheduled basis.
- **Command line interface (CLI)** — Amazon S3 also provides a CLI for interacting and ingesting data into Amazon S3 buckets.
- **File synchronization utilities** — Utilities such as rsynch and s3synch can keep source data directories in sync with Amazon S3 buckets, as a way to move files from source locations to Amazon S3 buckets.

Streaming Ingestion

In this mode, data is ingested in streams (e.g., clickstream data). Architecturally, there must be a streaming store that accepts and stores streaming data at scale, and in real time. Additionally, data collectors that collect data at the sources are needed to send data to the streaming store.

- **Streaming stores** — There are various options for the streaming stores. Amazon Kinesis Streams and Amazon Kinesis Firehose are fully managed streaming stores. Streams and Firehose also provide SDKs and APIs for programmatic integration. Alternatively, open-source platforms such as Kafka can be installed and configured on EC2 clusters to manage streaming data ingestion and storage.

- **Data collectors** — These can be web, mobile, or OTT applications that send data directly to the streaming store, or collector agents running next to the data sources (e.g., clickstream logs) that send data to the streaming store in real time. There are several options for the data collectors. Flume and Flentd are two open-source data collectors that can collect log data and send data to streaming stores. An Amazon Kinesis agent can be used as the data collector for Streams and Firehose.

One common practice is to ingest all the input data into staging Amazon S3 buckets or folders first, perform further data processing, and then store the data in target Amazon S3 buckets or folders.

Any data processing related to data quality (e.g., data completeness, invalid data) should be handled at the sources when possible, and is not discussed in this document. During this stage, the following data processing might be needed.

- **Data transformation** — This could be transformation of source data to the defined common standards. For example, breaking up a single name field into first name, middle name, and last name fields.
- **Metadata extraction and persistence** — Any metadata associated with input files should be extracted and stored in a persistent store. This could include file name, file or record size, content description, data source information, and date or time information.
- **Data enrichment** — Raw data can be enhanced and refined with additional information. For example, you can enrich source IP addresses with geographic data.
- **Table schema creation and maintenance** — Once the data is processed into a target structure, you can create the schemas for the target systems.

File Formats

The various file formats have tradeoffs regarding compatibility, storage efficiency, read performance, write performance, and schema extensibility. In the Hadoop ecosystem there are many variations of file-based data stores. The following are some of the more common ones in use.

- **Comma Separated Values (CSV)** — CSV, typically the lowest common denominator of file formats, excels at providing tremendous compatibility between platforms. It's a common format for going into and out of the Hadoop ecosystem. This file type can be easily inspected and edited with a text editor, which provides flexibility for ad hoc usage. One drawback is poor support for compression, so the files tend to take up more storage space than some other available formats. You should also note that CSV sometimes has a header row with column names. Avoid using this with machine learning tools because it inhibits the ability to arbitrarily split files.
- **JavaScript Object Notation (JSON)** — JSON is similar to CSV in that text editors can consume this format easily. JSON records can be stored using a delimiter such as a newline character, as a demarcation to split large data sets across multiple files. However, JSON files include some additional metadata whereas CSV files typically do not when used in Hadoop. JSON files with one record should be avoided because this would generally result in too many small files.
- **Apache Parquet** — A columnar storage format that is integrated into much of the Hadoop ecosystem, Parquet allows for compression schemes to be specified on a per-column level. This provides the flexibility to take advantage of compression in the right places without the penalty of wasted CPU cycles compressing and decompressing data that doesn't need compressing. Parquet is also flexible for encoding columns. Selecting the right encoding mechanism is also important to maximize CPU utilization when reading and writing data. Because of the columnar format, Parquet can be very efficient when processing jobs that only require reading a subset of columns. However, this columnar format also comes with a write penalty if your processing includes writes.
- **Apache Avro** — Avro can be used as a file format or as an object format that is used within a file format, such as Parquet. Avro uses a binary data format requiring less space to represent the same data in a text format. This results in lower processing demands in terms of I/O and memory. Avro also has the advantage of being compressible, further reducing the storage size and increasing disk read performance. Avro includes schema data and data that is defined in JSON, while still being persisted in a binary format. The Avro data format is flexible and expressive, allowing for schema evolution and support for more complex data structures such as nested types.

- **Apache ORC** — Another column-based file format designed for high speed within Hadoop. For flat data structures, ORC has the advantage of being optimized for reads that use predicates in WHERE clauses in Hadoop ecosystem queries. It also compresses quite efficiently with compression schemes such as Snappy, Zlib, or GZip.
- **Sequence files** — Hadoop often uses sequence files as temporary files during processing steps of a MapReduce job. Sequence files are binary and can be compressed to improve performance and reduce required storage volume. Sequence files are stored row based, with sync ending markers enabling splitting. However, any edits will require the entire file to be rewritten.

Data Store

For the data store portion of your solution, you need storage for the data, derived data lake schemas, and a metadata data catalogue. As part of that, a critical decision to make is the type or types of data file formats you will process. Many types of object models and storage formats are used for machine learning. Common storage locations include databases and files.

From a storage perspective, Amazon S3 is the preferred storage option for data science processing on AWS. Amazon S3 provides highly durable storage and seamless integration with various data processing services and machine learning platforms on AWS.

Data Lake Schemas

Data lake schemas are Apache HIVE tables that support SQL-like data querying using Hadoop-based query tools such as Apache HIVE, Spark SQL, and Presto. Data lake schemas are based on the schema-on-read design, which means table schemas can be created after the source data is already loaded into the data store. A data lake schema uses a HIVE metastore as the schema repository, which can be accessed by different query engines. In addition, the tables can be created and managed using the HIVE engine directly.

Metadata Data Catalogue

A metadata data catalogue contains information about the data in the data store. It can be loosely categorized into three areas: technical, operational, and business.

- **Technical metadata** refers to the forms and structure of the data. In addition to data types, technical metadata can also contain information about what data is valid and the data's sensitivity.
- **Operational metadata** captures information such as the source of the data, time of ingestion, and what ingested the data. Operational metadata can show data lineage, movement, and transformation.
- **Business metadata** provides labels and tags for data with business-level attributes to make it easier for someone to search and browse data in the data store.

There are different options to process and store metadata on AWS. One way is to trigger AWS Lambda functions by using Amazon S3 events to extract or derive metadata from the input files and store metadata in Amazon DynamoDB.

Processing by Data Scientists

When all relevant data is available in the data store, data scientists can perform offline data exploration and model selection, data preparation, and model training and generation based on the defined business objectives. The following solutions were selected because they are ideal for handling the large amount of data M&E use cases generate.

Interactive Data Exploration

To develop the data understanding needed to support the modeling process, data scientists often must explore the available datasets and determine their usefulness. This is normally an interactive and iterative process and requires tools that can query data quickly across massive amounts of datasets. It is also useful to be able to visualize the data with graphs, charts, and maps. Table 1 provides a list of data exploration tools available on AWS, followed by some specific examples that can be used to explore the data interactively.

Table 1: Data exploration tool options on AWS

Query Style	Query Engine	User Interface Tools	AWS Services
SQL	Presto	AirPal JDBC/ODBC Clients Presto CLI	EMR

Query Style	Query Engine	User Interface Tools	AWS Services
	Spark SQL	Zeppelin Spark Interactive Shell	EMR
	Apache HIVE	Apache HUE HIVE Interactive Shell	EMR
Programmatic	R/SparkR (R)	RStudio R Interactive Shell	EMR
	Spark(PySpark, Scala)	Zeppelin Spark Interactive Shell	EMR

Presto on Amazon EMR

The M&E datasets can be stored in Amazon S3 and are accessible as external HIVE tables. An external Amazon RDS database can be deployed for the HIVE metastore data. Presto running in an Amazon EMR cluster can be used to run interactive SQL queries against the datasets. Presto supports ANSI SQL so you can run complex queries, as well as aggregation against any dataset size, from gigabytes to petabytes.

Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) drivers support connections from data visualization tools such as Qlikview, Tableau, and Presto for rich data visualization. Web tools such as AirPal provide an easy-to-use web front end to run Presto queries directly.

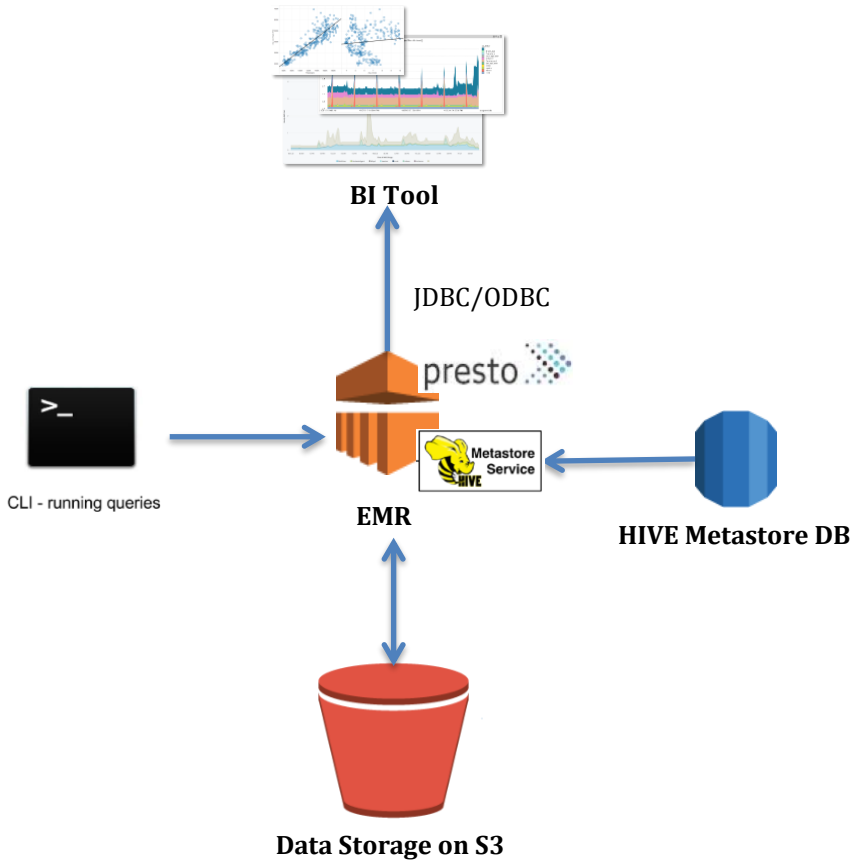

Figure 4 — Data exploration with Presto

Apache Zeppelin with Spark on EMR

Another tool for data exploration is Apache Zeppelin notebook with Spark. Spark is a general-purpose cluster computing system. It provides high-level APIs for Java, Python, Scala, and R. Spark SQL, an in-memory SQL engine, can integrate with HIVE external tables using HiveContext to query the dataset. Zeppelin provides a friendly user interface to interact with Spark and visualize data using a range of charts and tables. Spark SQL can also support JDBC/ODBC connectivity through a server running Thrift.

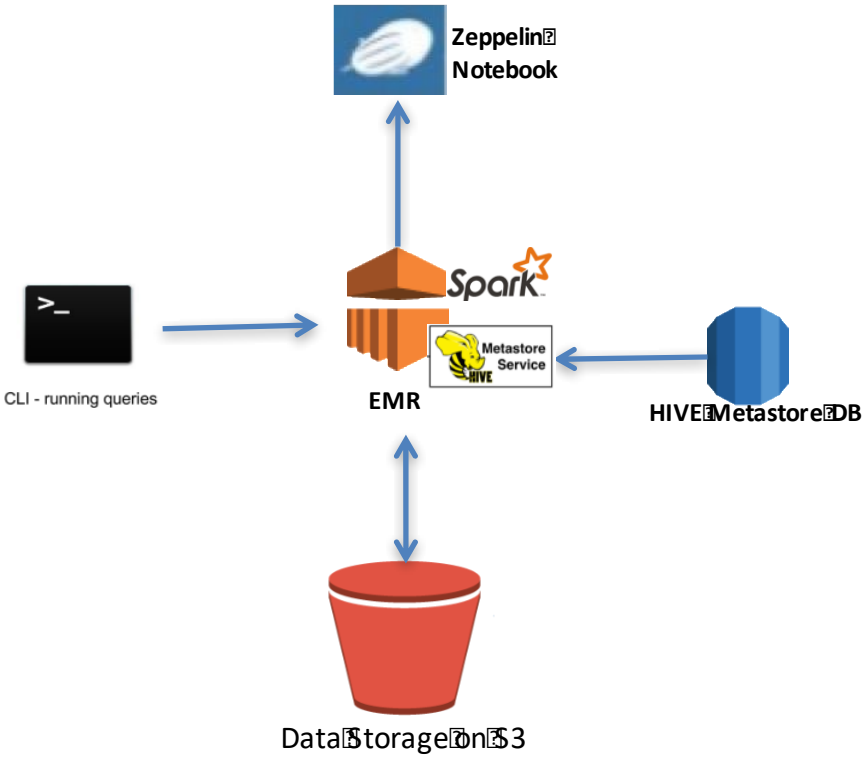

Figure 5 — Data exploration with Zeppelin

R/SparkR on EMR

Some data scientists like to use R/RStudio as the tool for data exploration and analysis, but feel constrained by the limitations of R, such as single-threaded execution and small data size support. SparkR provides both the interactive environment, rich statistical libraries, and visualization of R. Additionally, SparkR provides the scalable, fast distributed storage and processing capability of Spark. SparkR uses DataFrames as the data structure, which is a distributed collection of data organized into named columns. DataFrames can be constructed from wide array of data sources, including HIVE tables.

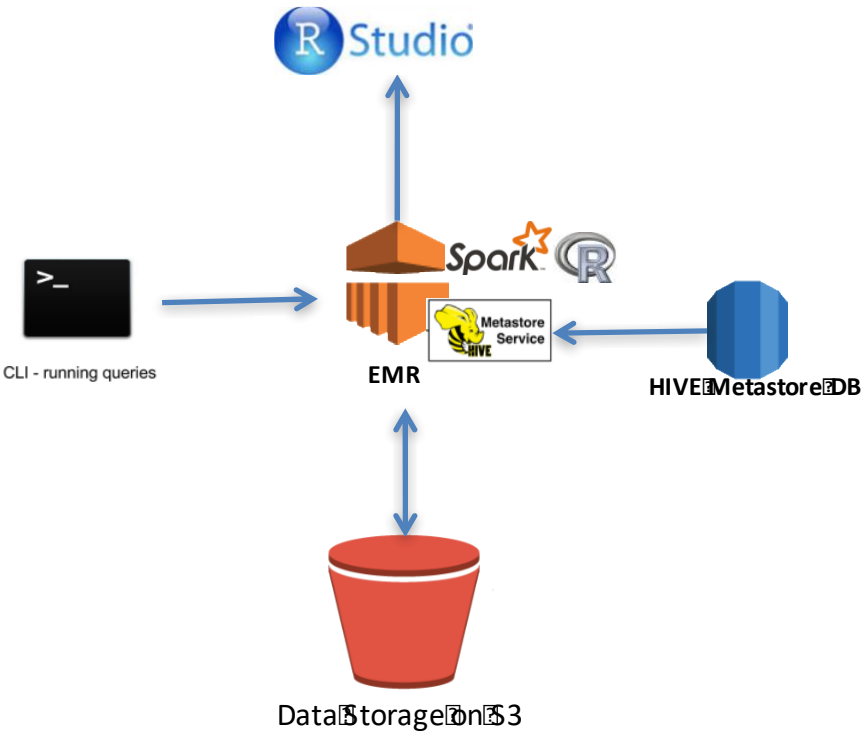

Figure 6 — Data exploration with Spark + R

Training Data Preparation

Data scientists will need to prepare training data to support supervised and unsupervised model training. Data is formatted, transformed, and enriched for the modeling purpose. As only the relevant data variable should be included in the model training, feature selection is often performed to remove unneeded and irrelevant attributes that do not contribute to the accuracy of the predictive model. Amazon ML provides feature transformation and feature selection capability that simplifies this process. Labeled training datasets can be stored in Amazon S3 for easy access by machine learning services and frameworks.

Interactive Model Training

To generate and select the right models for the target business use cases, data scientists must perform interactive model training against the training data. Table 2 provides a list of use cases with potential products that can you can use to create your solution, followed by several example architectures for interactive model training.

Table 2 — Machine learning options on AWS

M&E Use Case	ML Algorithms	ML Software	AWS Services
Segmentation	Clustering (e.g., k-Means)	Spark ML Mahout R	EMR
Recommendation	Collaborative Filtering (e.g., Alternating Least Square)	Spark ML Apache Mahout	EMR
	Neural Network	MXNet	Amazon EC2/GPU
Customer Churn	Classification (e.g., Logistic Regression)	Managed Service	Amazon Machine Learning
		Spark ML Apache Mahout R	EMR
Sentiment Analysis	Classification (e.g., Logistic Regression)	Managed Service	Amazon Machine Learning
	Classification (e.g., Support Vector Machines, Naïve Bayes)	Spark ML Mahout R	EMR
	Neural Network	MXNet Caffe Tensorflow Torch Theano	Amazon EC2/GPU

Amazon ML Architecture

Amazon ML is a fully managed machine learning service that provides the quickest way to get started with model training. Amazon ML can support long tail use cases, such as churn and sentiment analysis, where logistic regression (for classification) or linear regression (for the prediction of a numeric value) algorithms can be applied. The following are the main steps of model training using Amazon ML.

1. **Data source creation** — Label training data is loaded directly from the Amazon S3 bucket where the data is stored. A target column indicating the prediction field must be selected as part the data source creation.
2. **Feature processing** — Certain variables can be transformed to improve the predictive power of the model.
3. **ML model generation** — After the data source is created, it can be used to train the machine learning model. Amazon ML automatically splits the labeled training set into a training set (70%) and an evaluation set (30%). Depending on the selected target column, Amazon ML automatically picks one of three algorithms (binary logistic regression, multinomial logistic regression, or linear regression) for the training.
4. **Performance evaluation** — Amazon ML provides model evaluation features for model performance assessment and allows for adjustment to the error tolerance threshold.

All trained models are stored and managed directly within the Amazon ML service and can be used for both batch and real-time prediction.

Spark ML/Spark MLlib on Amazon EMR Architecture

For the use cases that require other machine learning algorithms, such as clustering (for segmentation) and collaborative filtering (for recommendation), Amazon EMR provides cluster management support for running Spark ML.

To use Spark ML and Spark MLlib for interactive data modeling, data scientists have two choices. They can use Spark shell by SSH'ing onto the master node of the EMR cluster, or use data science notebook Zeppelin running on the EMR cluster master node.

Spark ML or Spark MLlib supports a range of machine learning algorithms for classification, regression, collaborative filtering, clustering, decomposition, and optimization. Another key benefit of Spark is that the same engine can perform data extraction model training and interactive query.

A data scientist will need to programmatically train the model using languages such as Java, Python, or Scala. Spark ML provides a set of APIs for creating and tuning machine learning pipelines. The following are the main concepts to understand for pipelines.

- **DataFrame** — Spark ML uses a DataFrame from Spark SQL as an ML dataset. For example, a DataFrame can have different columns corresponding to different columns in the training dataset that is stored in Amazon S3.
- **Transformer** — An algorithm that can transform one DataFrame into another DataFrame. For instance, an ML model is a `Transformer` that transforms a `DataFrame` with features into a `DataFrame` with predictions.
- **Estimator** — An algorithm that can fit on a DataFrame to produce a transformer.
- **Parameter** — All transformers and estimators share a common API for specifying parameters.
- **Pipeline** — Chains multiple Transformers and Estimators to specify an ML workflow.

Spark ML provides two approaches for model selection: cross-validation and validation split.

With **cross-validation**, the dataset is split into multiple folds that are used as separate training and test datasets. Two-thirds of each fold are used for training and one-third of each fold is used for testing. This approach is a well-established method for choosing parameters and is more statistically sound than heuristic tuning by hand. However, it can be very expensive as it cross-validates over a grid of parameters.

With **validation split**, the dataset is split into a training asset and a test data asset. This approach is less expensive, but when the training data is not sufficiently large it won't produce results that are as reliable as using cross-validation.

Spark ML supports a method to export models in the Predictive Model Markup Language (PMML) format. The trained model can be exported and persisted into an Amazon S3 bucket using the model save function. The saved models can then be deployed into other environments and loaded for generating prediction.

Machine Learning on EC2/GPU/EMR Architectures

For use cases that require different machine learning frameworks that are not supported by Amazon ML or Amazon EMR, these frameworks can be installed and run on EC2 fleets. An AMI is available with preinstalled machine learning

packages, including MXNet, CNTK, Caffe, Tensorflow, Theano, and Torch. Additional machine learning packages can be added easily to EC2 instances.

Other machine learning frameworks can also be installed on Amazon EMR via bootstrap actions to take advantage of the EMR cluster management. Examples include Vowpal Wabbit, Skytree, and H2O.

Prediction Processing and Serving

One architecture pattern for serving predictions quickly using both historic and new data is the lambda architecture. The components for this architecture include a batch layer, speed layer, and serving layer all working together to enable up-to-date predictions as new data flows into the system.

Despite its name, this pattern is not related to the AWS Lambda service. The

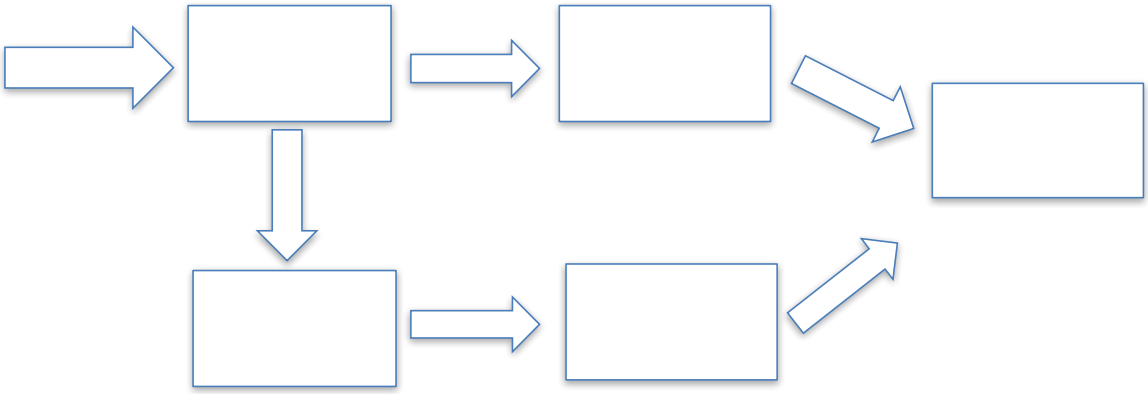

Figure 7 — Lambda architecture components

following is a brief description for each portion of the pattern shown in Figure 7.

- **Event data** — Event-level data is typically log data based on user activity. This could be data captured on websites, mobile devices, or social media activities. Amazon Mobile Analytics provides an easy way to capture user activity for mobile devices. The Amazon Kinesis Agent makes it easy to ingest log data such as web logs. Also, the Amazon Kinesis Producer Library (KPL) makes it easy to programmatically ingest data into a stream.
- **Streaming** — The streaming layer ingests data as it flows into the system. A popular choice for processing streams is Amazon Kinesis Streams because it is a managed service that minimizes administration and maintenance. Amazon Kinesis Firehose can be used as a stream that stores all the records to a data lake, such as an Amazon S3 bucket.

- **Data lake** — The data lake is the storage layer for big data associated with event-level data generated by M&E users. The popular choice in AWS is Amazon S3 for highly durable and scalable data.
- **Speed layer** — The speed layer continually updates predictive results as new data arrives. This layer processes less data than the batch layer so the results may not be as accurate as the batch layer. However, the results are more readily available. This layer can be implemented in Amazon EMR using Spark Streaming.
- **Batch layer** — The batch layer processes machine learning models using the full set of event-level data available. This processing can take longer but can produce higher fidelity predictions. This layer can be implemented using Spark ML in Amazon EMR.
- **Serving layer** — The serving layer responds to predictions on an ongoing basis. This layer arbitrates between the results generated by the batch and speed layers. One way to accomplish this is by storing predictive results in a NoSQL database such as DynamoDB. With this approach predictions are stored on an ongoing basis by both the batch and speed layers as they are processed.

AWS Services and Benefits

Machine learning solutions come in many shapes and sizes. Some of the AWS services commonly used to build machine learning solutions are described in the following sections. During the predictive analytics process workflow, different resources are needed throughout different parts of the lifecycle. AWS services work well in this scenario because resources can run on demand and you pay only for the services you consume. Once you stop using them, there are no additional costs or termination fees.

Amazon S3

In the context of machine learning, Amazon S3 is an excellent choice for storing training and evaluation data. Reasons for this choice include its provision of highly parallelized low latency access, that it can store vast amounts of structured and unstructured data, and is low cost. Amazon S3 is also integrated into a useful ecosystem of tools and other services, extending the functionality of Amazon S3 for ingestion and processing of new data. For example, Amazon Kinesis Firehose can be used to capture streaming data. AWS Lambda event-

based triggers enable serverless compute processing when data arrives in an Amazon S3 bucket. Amazon ML uses Amazon S3 as input for training and evaluation datasets, as well as for batch predictions. Amazon EMR, with its ecosystem of machine learning tools, also benefits from using Amazon S3 buckets for storage. By using Amazon S3, EMR clusters can decouple storage and compute, which has the advantage of scaling each independently. It also facilitates using transient clusters or multiple clusters for reading the same data at the same time.

Amazon Kinesis

Amazon Kinesis is a platform for streaming data on AWS, offering powerful services to make it easy to load and analyze streaming data. The Amazon suite of services also provides the ability for you to build custom streaming data applications for specialized needs. One such use case is applying machine learning to streaming data. There are three Amazon Kinesis services that fit different needs:

- Amazon Kinesis Firehose accepts streaming data and persists the data to persistent storage including Amazon S3, Amazon Redshift, and Amazon Elasticsearch Service.
- Amazon Kinesis Analytics lets you gain insights from streaming data in real time using standard SQL. Analytics also include advanced functions such as the Random Cut Forest, which calculates anomalies on streaming datasets.
- Amazon Kinesis Streams is a streaming service that can be used to create custom streaming applications or integrate into other applications, such as Spark Streaming in Amazon EMR for real-time Machine Learning Library (MLlib) workloads.

Amazon EMR

Amazon EMR simplifies big data processing, providing a managed Hadoop framework. This approach makes it easy, fast, and cost-effective for you to distribute and process vast amounts of data across dynamically scalable Amazon EC2 instances. You can also run other popular distributed frameworks such as Apache Spark and Presto in Amazon EMR, and interact with data in other AWS data stores such as Amazon S3 and Amazon DynamoDB. The large ecosystem of Hadoop-based machine learning tools can be used in Amazon EMR.

Amazon Machine Learning (Amazon ML)

Amazon ML is a service that makes it easy for developers of all skill levels to use machine learning technology. Amazon ML provides visualization tools and wizards that guide you through the process of creating machine learning models without having to learn complex machine learning algorithms and technology. Once your models are ready, Amazon ML makes it easy to obtain predictions for your application using simple APIs, without having to implement custom prediction generation code or manage any infrastructure.

Amazon ML is based on the same proven, highly scalable, machine learning technology used for years by Amazon's internal data scientist community. The service uses powerful algorithms to create machine learning models by finding patterns in your existing data. Then, Amazon ML uses these models to process new data and generate predictions for your application.

Amazon ML is highly scalable and can generate billions of predictions daily, and serve those predictions in real time and at high throughput. With Amazon ML, there is no upfront hardware or software investment, and you pay as you go, so you can start small and scale as your application grows.

AWS Data Pipeline

AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, as well as on-premises data sources, at specified intervals. With Data Pipeline, you can regularly access your data where it's stored, transform and process it at scale, and efficiently transfer the results to AWS services such as Amazon S3, Amazon RDS, Amazon DynamoDB, and Amazon EMR. Data Pipeline helps you easily create complex data processing workloads that are fault tolerant, repeatable, and highly available. You don't have to worry about ensuring resource availability, managing intertask dependencies, retrying transient failures or timeouts in individual tasks, or creating a failure notification system. Data Pipeline also enables you to move and process data that was previously locked up in on-premises data silos, unlocking new predictive analytics workloads.

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2 is a simple yet powerful compute service that provides complete control of server instances that can be used to run many machine learning packages. The EC2 instance type options include a wide variety of options to

meet the various needs of machine learning packages. These include compute optimized instances with relatively more CPU cores, memory optimized instances for packages that use lots of RAM, and massively powerful GPU optimized instances for packages that can take advantage of GPU processing power.

Amazon CloudSearch

Amazon CloudSearch is a managed service in the AWS Cloud that makes it simple and cost-effective to set up, manage, and scale a search solution for your website or application. In the context of predictive analytics architecture, CloudSearch can be used to serve prediction outputs for the various use cases.

AWS Lambda

AWS Lambda lets you run code without provisioning or managing servers. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration. In the predictive analytics architecture, Lambda can be used for tasks such as data processing triggered by events, machine learning batch job scheduling, or as the back end for microservices to serve prediction results.

Amazon Relational Database Service (Amazon RDS)

Amazon RDS makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business. In the predictive analytics architecture, Amazon RDS can be used as the data store for HIVE metastores, and as the database for servicing prediction results.

Amazon DynamoDB

Amazon DynamoDB is a fast and flexible NoSQL database service, ideal for any applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed cloud database and supports both document and key-value store models. In the predictive analytics architecture, DynamoDB can be used to store data processing status or metadata, or as a database to serve prediction results.

Conclusion

In this paper, we provided an overview of the common Media and Entertainment (M&E) predictive analytics use cases. We presented an architecture that uses a broad set of services and capabilities of the AWS Cloud to enable both the data scientist workflow and the predictive analytics generation workflow in production.

Contributors

The following individuals and organizations contributed to this document:

- David Ping, solutions architect, Amazon Web Services
- Chris Marshall, solutions architect, Amazon Web Services

Document revisions

Date	Description
March 30, 2021	Reviewed for technical accuracy
February 24, 2017	Corrected broken links, added links to libraries, and incorporated minor text updates throughout.
December 2016	First publication