International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Machine Learning and AI Integration in Databricks for Big Data

Ravi Shankar Koppula

Satsyil Corp, Herndon, VA, USA
Email: *ravikoppula100[at]gmail.com*

Abstract: ***This paper explores the integration of machine learning (ML) and artificial intelligence (AI) within the Databricks platform, emphasizing its capabilities for big data analytics. Databricks, built on Apache Spark, provides a unified analytics environment that supports data science collaboration through interactive notebooks and scalable processing. The paper discusses the foundational concepts of machine learning and AI, the significance of data preparation, and the utilization of ML libraries such as MLLib and H2O within Databricks. Advanced techniques including deep learning and neural networks are examined, highlighting their practical applications in real-world scenarios. The integration of Databricks' tools with machine learning frameworks enables efficient data engineering and model deployment, offering a robust solution for enterprises to leverage big data for predictive analytics and decision-making.***

Keywords: Databricks, Apache Spark, Machine Learning (ML), Artificial Intelligence (AI), Big Data, Data Preparation, MLLib, H2O, Deep Learning, Neural Networks

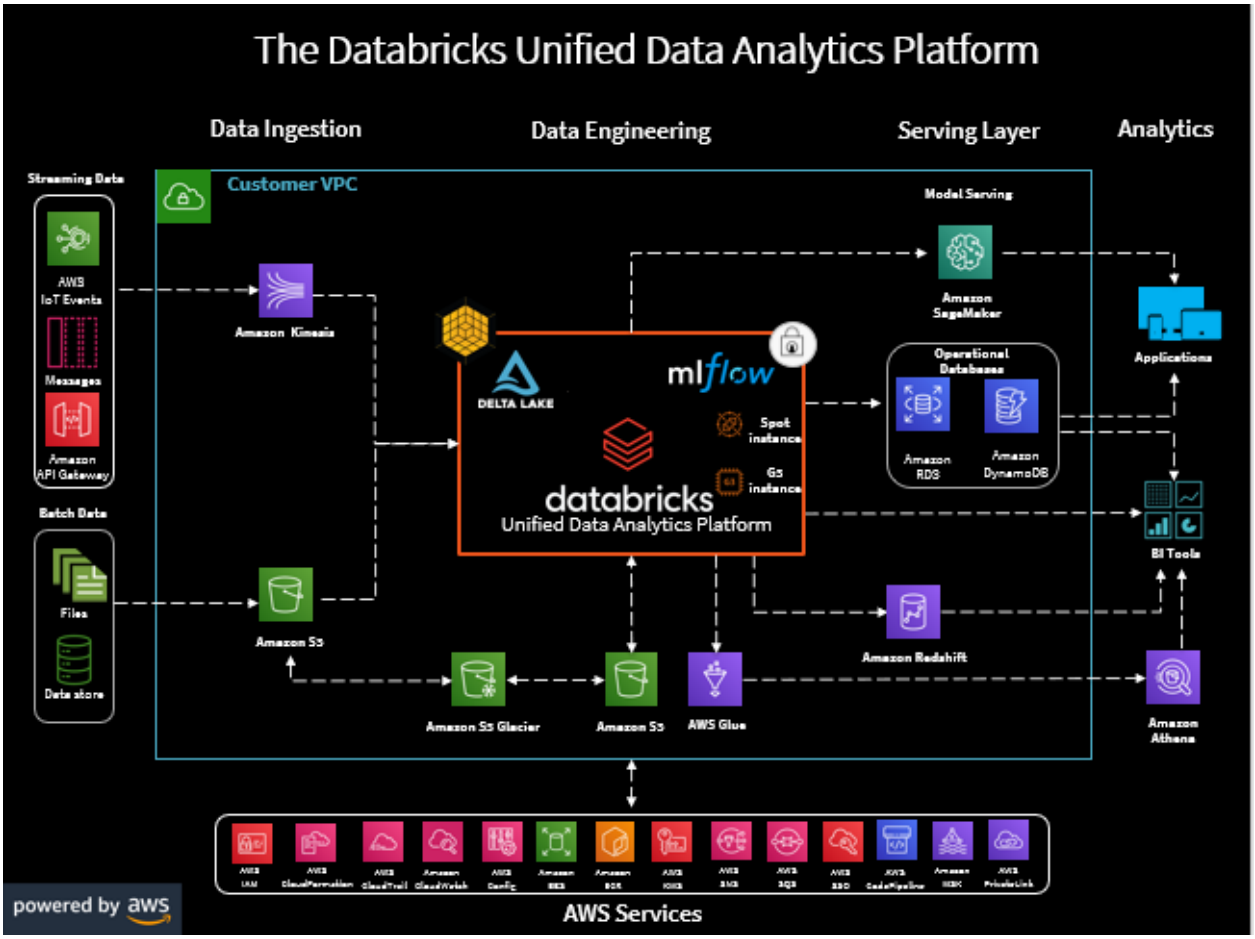
1. Introduction to Databricks

Databricks is a unified analytics platform for big data processing and machine learning. Data scientists and engineers can collaborate through an interactive workspace by working with data, notebooks, and jobs. Databricks provides different ways to connect your data and use it for analytics processing and machine learning. It has integrations with Azure cloud services and storage accounts. Databricks uses Apache Spark for analytics processing and as a highly available and fully managed cloud service, it offers a platform to process big data with more than 12 programming languages [1]. Notebooks in Databricks provide an interactive workspace that combines code execution with horizontal scaling, text-based documentation, and visualization in a collaborative environment. Databricks comes with built-in support for visualizing data processing results with SQL and Spark built-in magic commands. It can utilize a pre-built library of visualization tools or easily import custom visualization libraries as needed. Databricks allows building dashboards based on multiple visualization outputs and is supported in both notebooks and SQL analyses. Moreover, it supports alerting on the results of SQL analysis and any visualization output on dashboards [2]. There is also a separate analytics workspace, SQL Analytics, specifically for BI tools where suggested charts are automatically generated based on the SQL output. Every visualization can also be added to the dashboards directly from this SQL workspace. Databricks also comes with built-in support for scheduling data processing jobs. Data pipelines can be orchestrated through notebooks, SQL analysis, or data processing jobs using pre-built commands or dedicated APIs. Jobs can also be integrated with third-party orchestrators.

1.1. Overview of Databricks Platform

Databricks is a collaborative Spark-based analytics platform, which supports the usage of notebooks, Spark SQL, and DBU-based premium resources. A variety of pre-created libraries can also be installed on clusters within the Databricks ecosystem. All of these advantages contribute to a seamless environment for data scientists and data engineers to build Machine Learning (ML) models [1].

Licensed Under Creative Commons Attribution CC BY



Databricks uses Apache Spark as its core processing engine. Spark supports DataFrames, which are well optimized, when compared to vanilla RDD; it supports SQL queries and many constructed functions (built-in or user-defined). Moreover, Spark is scalable and can work with a variety of protocols for file systems (from S3, HDFS, etc. enabling clusters to collect data from different sources). Finally, it can run either in memory or on disk, allowing users to define the optimal approaches based on the datasets (small or large) [2].

2. Foundations of Machine Learning and AI

Machine learning as a concept: The concept of computing itself dates back for thousands of years. The concept and act of computation are two separate entities, though. The idea of computation is the concept that, with a given set of inputs, the same exact answers will always be given back upon a specific request. In the practice of ancient computation, technologies built off irrational numbers and their approximations were frequently utilized to perform computing tasks.

The early to late 20th century saw the development of machines that could solve mathematical problems. These machines began to accurately solve these problems at a rate that surpassed that of humans. In the late 20th century, the rise of the modern digital computer began. These machines could solve many mathematical problems at absolutely phenomenal speed.

The concept of artificial intelligence also dates back thousands of years. The question of where the body ends and the soul begins fascinated thinkers for millennia. The idea, or question, of non-human intelligence has captured the minds of thinkers for thousands of years, too. Of special interest is the concept of creating mechanisms such that the self-aware, conscious state of mind often associated with mammals (and Homo sapiens in particular), can be replicated in a non-self-aware system (e.g. a machine). The field of AI also, however, encompasses the idea that machines can solve specific problems in a way that a person would solve them.

2.1. Basic Concepts and Terminology

Many data manipulations and transformations that can be achieved using PySpark SQL are good for Data Engineering or preparing data and therefore also important for advanced analysis. pyspark.sql is used to define DataFrames. pyspark.sql.types is used for data types. pyspark.ml is the main package of MLlib for modeling data. pyspark.ml.param is used for parameters. pyspark.sql functions is used to manipulate data and compute descriptive functions.

Machine learning can play a big role for Big Data, which is defined by three principal attributes: high-volume, high-velocity, and high-variety data. Big Data has the potential to unlock a wide range of new insights as data are normally unstructured and unique. Machine learning can help to understand Big Data. There are three main types of machine learning algorithms. With supervised learning, models use known labels to predict outcome of interest. With unsupervised learning, models seek to group individuals

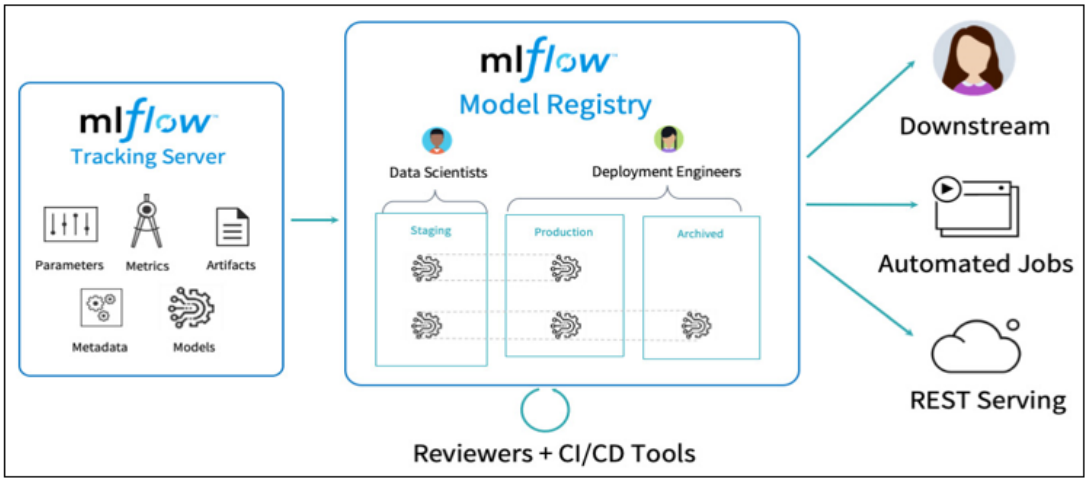
Licensed Under Creative Commons Attribution CC BY

with similar traits. And with reinforcement learning, models are using feedback over time to ultimately drive by experimentation. Although independent from the type of machine learning, there are common fundamental steps using Python for this process.

3. Integration of Machine Learning and AI in Databricks

Databricks Unified Analytics Platform provides an environment to integrate with machine learning libraries and deep learning frameworks for big data. The central technology we use inside Databricks to enable fast, scalable, and distributed ML for datasets of any size is Apache Spark, in combination with libraries such as MLLib or H2O. MLLib is Spark's scalable machine learning library, consisting of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, underlying optimization primitives, and so forth. Automatic Machine Learning (AutoML) library H2O, available in the Databricks ML Runtime and MLflow, is an easy-to-use toolkit that can handle ML processes using various machine learning algorithms. Databricks also provides Databricks Astronomy, a set of tools like RotationForest, Sparkling-Water, XGBoost, and so forth, to extend the reach of Databricks with machine learning models. These libraries and frameworks can be used seamlessly in Databricks for big data and AI use cases since the library provides built-in methods such as read or write, which can be used to interact with Spark big data pipeline.

Databricks allows data engineering and ML teams to work jointly on multi-stage data science problems, where efficiency is essential. They mostly share some of the ML programming technologies such as Apache Spark for big data processing, Pandas for data analysis, Databricks ML library for ML/deep-learning frameworks use. Although Databricks has a powerful workspace UI, there are problems for machine learning professionals and enterprise AI developers who rely on programming environments such as notebooks, workflows, which can be implemented through Databricks notebooks or workspaces and notebooks as the primary development environment. Databricks provides a simplified collaborative development environment of a notebook called a Databricks notebook, which is compatible with such big data processing and machine learning workloads, helping data teams to improve efficiency and predict problems.



Databricks notebooks' ability to provide a collaborative workspace is an enabling developer for organizations. It allows collaborative teams and deep understanding of deviated workflows to be involved in data engineering and machine learning efforts. It uses notebooks that allow users to fully automate data processing and predict results of the data with multiple languages like Python, R, SQL, Scala, and other frameworks around the big data ecosystem. It eliminates time for model iteration and increases productivity as data scientists may utilize data, test the model, and even run search experiments on the same platform. It also makes it easy to share findings and models for use with teammates, business stakeholders, or potential customers.

3.1. Data Preparation and Exploration

3.1.1 Data Preparation

One of the most important stages of machine learning is data preparation. This stage includes tasks such as data cleaning, transformation, and integration. The necessity of this stage is understandable. The better prepared the data, the more accurate the models built using that data. Data preparation also has a cost. In many projects, 80-90% of the time is allocated to data preparation. Databricks plays a key role in this area. Databricks not only provides a robust infrastructure for large volume and noisy data but also includes features for automating the process. In this section, we introduce some of the key features for data preparation.

3.1.2 Data Access

One of the key features of Databricks is its ability to access, integrate, and analyze data from a large variety of sources. This includes RDBMS databases, flat files, operational systems, and IoT data. Thanks to its robust infrastructure, Databricks can easily integrate diverse sources of existing data in terms of velocity, variety, and also veracity of the data. Large volume and velocity of data is not adequate by itself in AI and machine learning projects. Quality and feature selection are also key factors to consider. Data

Licensed Under Creative Commons Attribution CC BY

collection is often difficult and expensive, and if it is not managed correctly, data can lead to various degrees of degraded model performance.

4. Advanced Techniques and Models in Databricks

The machine learning and deep learning trainings have demonstrated how to implement some of the most common conventional and state-of-the-art models in Databricks.The performance characteristics of the model, such as accuracy, can tell you about a model's predictive capacity. However, in some cases, you would like to know what built a model to predict service outcomes like revenue modification from future positioning of engaging market items. This information helps you to map the inventory and business strategies that can increase long-term outcomes like gross margin and customer satisfaction. The model's interpretation function, which resembles function features, represent the facts that were characteristic for the improvement of a model. After you have created your model, you will call it on a variety of test data points to describe how it will work on a large population of search queries not as seen in training. The uplift model can be represented as a separate data point by creating a dataframe that includes estimated treatment outcome and expected control outcome for all your examples. Then, you'll plot the immediate uplift and return. The equilibrium uplift models will take the form of a plane.

4.1. Deep Learning and Neural Networks

The purpose of deep learning and neural networks is to establish a closer processing correlation with data and the model, not only an algorithmic. Basically, the correlation becomes learning with the purpose of executing simulations of interactions among neurons and synapses, guiding the biological reality. There are two approaches in neural networks: supervised and unsupervised. In supervised mode, the network model response is predetermined and in the unsupervised process the model determines the response. The unsupervised mode is used in pattern recognition, complex spatial organization, and logical association.

Considering the purpose of deep learning, where large quantities of data are handled for generating rich data presentations and high-level features, additional layers of clusters influence neural networks development for enhancing their computational capacity in operations involving analysis and imagination. The unique platform determining the execution of large-scale deep learning calculations is the neural flow operation, which provides operations that construct neural networks focused on various parallelization and execution strategies. At the beginning, TensorFlow was used to describe neural networks and generate symbolic expressions at each instruction level, enhancing the model. TensorFlow is not yet considered to be a released version, however, it presented successful partnerships and explorations, excellent visualization and tuning tools. Spark aims to become more dynamic for easy deployment of artificial intelligence and deep learning, and it is expected that Tensor Spark Server can overcome existing limitations and consolidation benchmarks for large-scale deep learning calculations.

5. Practical Applications and Case Studies

At Unilever, they developed a next-best-action ML framework for multi-channel marketing optimization, which was required to share data and collaborate with business units on an ongoing basis. In addition, the ML models also needed to be served for predictions/training to support business processes across different geographies. Data Lakehouse was used for model version management, and Databricks MLflow was used to automate model deployment. The online on-demand traversal processing was performed using the Delta Lake's merge streaming capability with Databricks's Structured Streaming on Spark. The batch processing has used periodically scheduled jobs. Python was the primarily used programming language for modeling and development. Python-based pandas and scikit-learn libraries were used for the data modeling and processing. Performance-critical operations were migrated to PySpark to leverage Apache Spark's scale-out big data processing capability. MLflow's Model Registry stored all the versions of models and their respective metadata. The real-time serving of transactions was carried out by a PySpark job that deployed the accepted model version using Spark UDFs and was pushed to the target data. Eurocontrol, the organization providing air navigation services throughout Europe, has developed a scalable ML toolkit to improve aviation and safety with less complexity.

5.1. Real-world Use Cases in Big Data

In order to grasp the applications of ML and AI, one must first understand that these solutions are designed to address business challenges. There are tangible real-world scenarios to consider, some of which may evolve into fundamental use cases for the big data community. When it comes to cloud deployment, Databricks stands out as a frontrunner in the realm of big data. With its advantages as a hosted Spark offering, Databricks has the potential to excel in numerous use cases, with a particular emphasis on iterative machine learning and artificial intelligence applications. These encompass a variety of scenarios such as financial fraud detection, retail supply chain optimization, data governance, network intrusion detection, and predictive maintenance. These use cases entail a combination of feature selection, visualizations, numerical operations, data sampling, statistics, and graph algorithms as part of the analytical workflow. [3][4][5][6][7][8]

A high-level solution for such use cases involves the following steps: ingest the data into Databricks to create ML and AI model training, feature engineering, model training and evaluation, model deployment, and feature transformation. Given the narrative methodology, it is possible to identify and provide guidance on how to create Glue, which is used to ingest data, while Databricks SQL is used to transform the ingested data (in Advent). Concerning training and evaluating models, several machine learning libraries are made available as part of the initial environment configuration. Additionally, to support the iterative nature of model training, model results are published directly to

Licensed Under Creative Commons Attribution CC BY

access for monitoring purposes. To trace who did what and why, the workspace feature is turned on, which requires a storage account setup and linked for auditing purposes. Models are published as APIs as well as Azure functions. Furthermore, Glue in combination with Databricks utilities such as JDBC can be implemented for data preparation, feature integration, and model deployment and in life cycle utilization respectively.

6. Conclusion

Integrating machine learning and AI within Databricks offers a robust and scalable solution for harnessing big data's potential. This paper has explored the synergies between Databricks and advanced analytics, demonstrating how the platform's unified data analytics environment simplifies the development, deployment, and management of machine learning models. Key benefits include the seamless collaboration between data engineers, data scientists, and business analysts, enabled by Databricks' collaborative workspaces and integrated tools.

The integration of Apache Spark in Databricks enhances data processing capabilities, allowing for efficient handling of large datasets and accelerating machine learning workflows. Furthermore, the platform's compatibility with popular machine learning libraries and frameworks such as TensorFlow, PyTorch, and Scikit-learn provides flexibility and broadens the scope of potential applications.

Databricks' robust support for MLOps, through features like MLflow, ensures that models can be easily tracked, versioned, and deployed, promoting a streamlined transition from development to production. This end-to-end management capability is crucial for maintaining model performance and reliability in real-world applications.

Security and compliance are also addressed effectively within Databricks, with enterprise-grade security features ensuring data privacy and integrity. The platform's ability to integrate with various data sources, both on-premises and cloud-based, further enhances its versatility and appeal for organizations looking to leverage big data for AI-driven insights.

In conclusion, Databricks stands out as a powerful platform for integrating machine learning and AI into big data workflows. Its comprehensive suite of tools and features not only simplifies the complexities of big data management but also empowers organizations to unlock valuable insights and drive innovation. As the landscape of big data continues to evolve, platforms like Databricks will play a pivotal role in shaping the future of data analytics and AI integration.

References

[1] J. Dai, Y. Wang, X. Qiu, D. Ding et al., "BigDL: A Distributed Deep Learning Framework for Big Data," 2018.
[2] M. Ali Mohamed, I. Mahmoud El-henawy, and A. Salah, "Usages of Spark Framework with Different Machine Learning Algorithms," 2021. ncbi.nlm.nih.gov
[3] S. K. Pala, "Databricks Analytics: Empowering Data Processing, Machine Learning and Real-Time Analytics," Machine Learning, 2021. researchgate.net
[4] S. Tang, B. He, C. Yu, Y. Li, K. Li, "A survey on spark ecosystem: Big data processing infrastructure, machine learning, and applications," in IEEE Transactions on Knowledge and Data Engineering, 2020.
[5] A. Kumar, S. Nakandala, Y. Zhang, S. Li, et al., "Cerebro: A layered data platform for scalable deep learning," in Proc. Conf. on Innovative Data Systems Research, 2021. nsf.gov
[6] R. Ilijason, "Beginning Apache Spark Using Azure Databricks: Unleashing Large Cluster Analytics in the Cloud," 2020.
[7] A. M. Fernández, D. Gutiérrez-Avilés, A. Troncoso, et al., "Automated deployment of a spark cluster with machine learning algorithm integration," Big Data Research, vol. 2020, Elsevier, 2020. us.es
[8] A. Cakir, Ö Akın, H. F. Deniz, and A. Yılmaz, "Enabling real time big data solutions for manufacturing at scale," journal of Big Data, 2022. springer.com

Licensed Under Creative Commons Attribution CC BY