# Automate Machine Learning with H2O Driverless AI on Dell Infrastructure

## Dell Validated Design for AI

July 2022

H19252

White Paper

### Abstract

This technical white paper discusses the benefits of automated machine learning and the challenges of non-automated model development that it overcomes. The paper presents an overview of the H2O Driverless AI product from H2O.ai, along with a solution architecture for H2O Driverless AI built on the Dell Validated Design for AI. It also provides several validated use cases using the solution.

Dell Technologies Solutions

Dell Validated Design

H2O.ai

DELL Technologies

## Copyright

# Contents

# Introduction

## Executive summary

Artificial intelligence (AI) and machine learning have revolutionized how organizations are using their data. Automated machine learning (AutoML) facilitates and improves the end-to-end data science process. This process includes everything from preprocessing and cleaning the data, selecting and engineering appropriate features, tuning and optimizing the model, analyzing results, explaining and documenting the model, and of course, deploying it into production.

AutoML accelerates your AI initiatives by providing methods and processes to make machine learning accessible to both experts and nonexperts alike. Organizations looking to apply machine learning quickly and accurately without employing large numbers of data scientists can benefit from AutoML capabilities. For organizations that have data scientists, AutoML equips and empowers them to create more robust models with accuracy, speed, and transparency to deliver better performance and outcomes. In all cases, AutoML helps organizations quickly discover business value hidden inside their data and easily use that data to address complex problems.

H2O Driverless AI is a comprehensive automated machine learning product that uses AI to do AI, optimizing data science workflows to increase both the quantity and quality of data science projects delivered to business stakeholders. It empowers data scientists to work on projects faster and more efficiently by using automation to accomplish key machine learning tasks in minutes or hours, not months.

H2O Driverless AI provides capabilities such as:

- Exploratory data analysis (AutoViz)
- Automatic feature engineering
- Model building and validation
- Automatic model documentation (AutoDoc)
- Model selection and deployment
- Machine learning interpretability (MLI)

AutoML does not replace machine learning operations (MLOps). AutoML focuses on automating and accelerating the model development portion of the ML pipeline, while MLOps provides an overall life cycle management framework for data preparation, model development, and coding. AutoML complements MLOps and can run successfully and efficiently with various MLOps frameworks such as cnvrg.io. MLOps provides an overall life cycle management framework for data preparation, model development, and coding.

With H2O Driverless AI bring-your-own recipes, and time series and automatic pipeline generation for model scoring, H2O Driverless AI provides companies with an extensible and customizable data science platform that addresses the needs of various use cases for every enterprise in every industry.

### Document purpose

This white paper discusses AutoML, including its benefits and the challenges of more traditional model development processes that it overcomes. The white paper provides an overview of the H2O Driverless AI product, presents a solution architecture for H2O Driverless AI built on the Dell Validated Design for AI with VMware, and describes several validated use cases using the solution. By deploying this solution, data scientists and IT professionals can move machine learning models out of the lab and into production faster and more easily, thus bringing a better return on investment (ROI) for an organization's machine learning investments.

### Audience

This white paper is intended for data scientists, solution architects, system administrators, and others developing and supporting AI and machine learning applications.

# The challenges of AI adoption

### Machine learning challenges

As organizations streamline decision making and improve customer experiences with AI, they are running into three core challenges: talent, time, and trust. First, there is not enough data science talent to build models for every use case by hand. Even with the right people, hand-coding takes too much time and is prone to errors. Then, the business must explain and validate each model so that users can trust the decisions that the model supports. The key to breaking through the talent, time, and trust barriers is the automation of advanced machine learning techniques with H2O Driverless AI.

#### Talent

Data scientists are in short supply for all but the largest technology companies. With H2O Driverless AI, both expert and novice data scientists can automatically build highly and transparent accurate models quickly. H2O Driverless AI is an award-winning AutoML product that embeds data science best practices from the world's leading experts in engineering and data science, including the world's top Kaggle Grandmasters. It uses a unique genetic algorithm that determines the best combination of features, models, and tuning parameters for each use case. Integrated best practices and guardrails ensure that models do not overfit the data and help with other common issues with which novice data scientists might need assistance. H2O Driverless AI enables companies to undertake more use cases with the talent that they already have or can easily find.

#### Time

Reducing the time to develop accurate, production-ready models is critical to delivering AI at scale. H2O Driverless AI automates time-consuming data science tasks such as advanced feature engineering, model selection, hyperparameter tuning, model stacking, and creation of an easy-to-deploy, low-latency scoring pipeline. With high-performance computing using both CPUs and GPUs, H2O Driverless AI compares thousands of combinations and iterations to find the best model in minutes or hours. Even experienced data scientists can use H2O Driverless AI to explore more techniques, feature combinations, and tuning parameters. H2O Driverless AI also streamlines model deployment that includes everything needed to run the model in production, taking the process time from experimentation to production from months to days.

### Trust

For organizations to adopt AI at scale, data teams, business leaders, and regulators must be able to explain, interpret, and trust AI results. H2O Driverless AI delivers industry-leading capabilities for understanding, debugging, and sharing model results, including an extensive machine learning interpretability (MLI) toolkit, fairness dashboards, automated model documentation, and reason codes for each prediction for service representatives and customers. With H2O Driverless AI, data teams have everything they need to build trust with business stakeholders and regulators.

## Overview of AutoML and H2O Driverless AI

H2O Driverless AI delivers enterprise-ready, scalable, and secure AutoML that can run on any cloud platform or in on-premises environments, using the architecture that this document describes. With an on-premises environment, you do not need to move your data to the cloud; you can perform AutoML securely wherever your data resides.

H2O Driverless AI enables data scientists to work on projects faster and more efficiently by using automation to perform key machine learning tasks in minutes or hours, not months.

H2O Driverless AI increases the productivity of data practitioners by automating data processing, feature engineering, model building, and hyper parameter tuning. It is a stand-alone platform that can be applied for use cases such as Natural Language Processing (NLP), time series forecasting, and image classifications. Enterprises can choose to deploy an MLOPs platform to enable cross-functional collaboration and to manage the end-to-end life cycle of their AI applications. In those cases, users can integrate H2O Driverless AI with their MLOps platform such as cnvrg.io (see Invoking H2O Driverless AI from cnvrg.io MLOps Platform).

### AutoML workflow with H2O Driverless AI

The following figure shows the steps in a typical AutoML workflow and how H2O Driverless AI enables these steps:

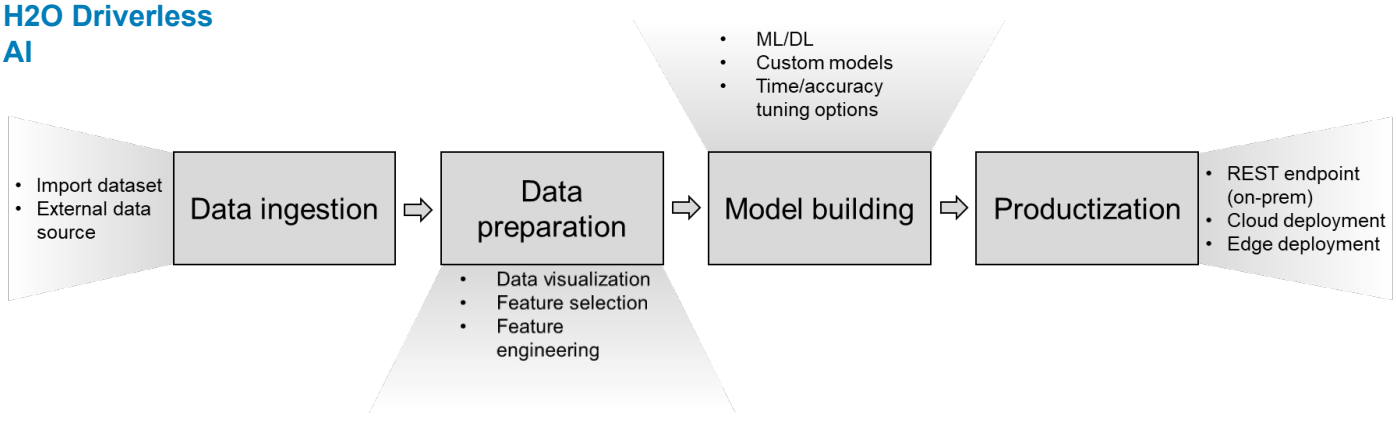


**Figure 1. AutoML workflow in H2O Driverless AI**

**Data ingestion**—The workflow begins with the data. Data ingestion consists of importing and obtaining data to perform analysis and training.

H2O Driverless AI can ingest data from datasets in various formats and file systems including Hadoop HDFS, Amazon S3 compatible storage, Azure Blob Storage, Google BigQuery, Google Cloud Storage, Apache Hive, JDBC, kdb+, MinIO, Snowflake, Data Recipe, Data Recipe File, and NFS. For larger datasets that are already available in PowerScale storage, H2O Driverless AI provides data connectors for accessing and ingesting data.

**Data preparation**—When the data is defined, the next step is data preparation. The dataset can be divided into training, test, and validation datasets. Data scientists can interactively model the data for exploration, analysis, and visualization using data plots and statistics. AutoML tools automatically perform feature engineering by extracting features (domain-specific attributes) from raw data and data transformations to suite ML algorithms.

H2O Driverless AI determines the best pipeline for a dataset, including automatic data transformation and feature engineering. Data scientists can control the number of original features used in model building by selecting or excluding columns in the dataset. H2O Driverless AI uses a unique genetic algorithm to automatically find new, high-value features and feature combinations for a specific dataset that are virtually impossible to find manually. The interface includes an easy-to-read variable importance chart that shows the significance of original and newly engineered features.

Automatic visualizations (AutoViz) in H2O Driverless AI provide robust exploratory data analysis capabilities by automatically selecting data plots based on the most relevant data statistics that are based on the data shape. In specific cases, AutoViz can suggest statistical transformation for some data. Experienced users can also customize visualizations to meet their needs. AutoViz helps users discover trends and issues such as large numbers of missing values or significant outliers that can impact modeling results.

**Model building**—When the data is prepared, the next step is model building. Automatic model building includes data transformations and hyperparameter tuning for the various models available in the AutoML product. It automatically trains several in-built models and selects the best model or a final ensemble of models based on user-defined parameters such as model accuracy.

Automatic model development in H2O Driverless AI is accomplished by running experiments. H2O Driverless AI trains multiple models and incorporates model hyperparameter tuning, scoring, and ensembling. Data scientists can configure parameters such as the accuracy, time, loss function, and interpretability for a specific experiment. This preview is automatically updated when any of the experiment's settings change (including the knobs). Users can also run multiple diverse experiments that provide an overview of the dataset. This feature provides data scientists with relevant information for determining complexity, accuracy, size, and time tradeoffs when putting models into production. H2O Driverless AI uses a genetic algorithm that incorporates a 'survival of the fittest' concept to determine the best model for specific dataset and configured options automatically.

**Productization**—When the experiment is completed, you can make new predictions and push the model for production, either in the cloud, on-premises, or at the edge.

H2O Driverless AI offers convenient options for deploying machine learning models, depending on where the AI application is run:

- Download the model and build your own container.
- Download a scoring pipeline.

  When the experiment (model building step) is complete, H2O Driverless AI can build a scoring pipeline that can be deployed to production. A scoring pipeline is a packaged experiment which includes artifacts necessary for model deployment, including model binary, runtime, readme, example, scripts, and so on. You can download two different types of scoring pipelines:

  - Python Scoring Pipeline
  - MOJO Scoring Pipeline, which is available with both Java and C++ backends

  The decision about which type of pipeline to use comes from various factors including the type of model being built in the experiment, use case, latency requirements, and so on. In general, MOJO Scoring Pipelines are faster but might require additional setup, while Python Scoring Pipelines are built into a `.whl` file, which easily installable in Python. H2O Driverless AI also allows you to visualize the scoring pipeline as a directional graph, as shown in the following figure:

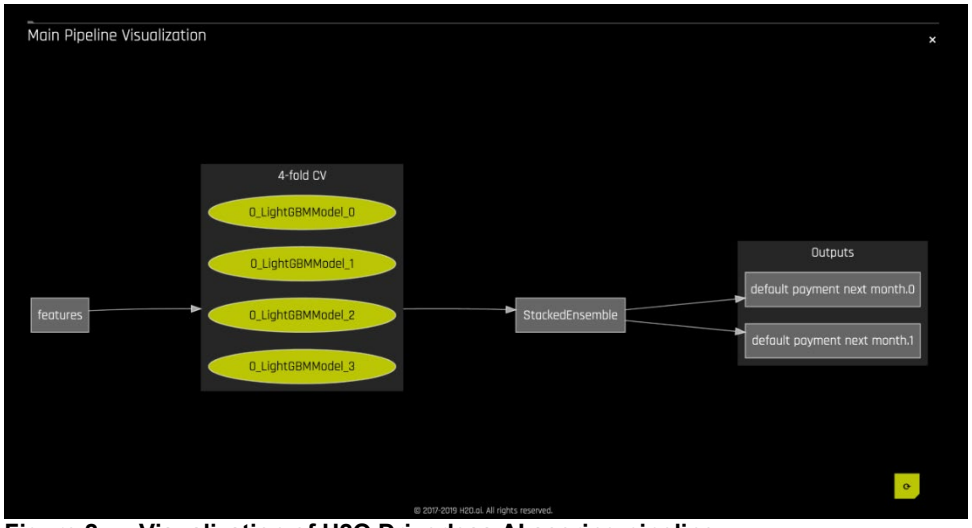


**Figure 2. Visualization of H2O Driverless AI scoring pipeline**

- Deploy the model directly in a cloud service.
- Configure the model to run on a local REST server with a couple of clicks.

## Key features

The H2O Driverless AI platform enables the following elements of AutoML:

- **Support for NVIDIA GPUs**—AI models are exploding in complexity, and automated data transformation and deep learning require massive compute power and scalability. H2O Driverless AI supports the latest NVIDIA GPUs to accelerate feature engineering and training of neural networks. NVIDIA's Multi-Instance GPU (MIG) feature can be used to partition the GPUs, increase overall GPU utilization, and support several types of use cases and deployments with guaranteed quality of service.
- **Integrated catalog of recipes and models**—H2O Driverless AI offers a rich catalog of AI models, transformers, and scorers for automatic feature engineering and model building.
- **Machine learning and deep learning**—H2O Driverless AI includes leading open-source transformers, embeddings, and frameworks for machine learning and deep learning techniques to handle various data science use cases. With H2O Driverless AI, you can automatically build models for Independent and Identically Distributed (IID) data, images, text, and more. For example, H2O Driverless AI includes TensorFlow CNNs for image modeling and NLP libraries from PyTorch, including BERT and other state-of-the-art techniques.
- **Machine Learning Interpretability (MLI)**—H2O Driverless AI provides robust explainability and fairness analysis for machine learning models and helps explore and demystify modeling results. It includes straightforward disparate impact analysis to test for model bias and provides reason codes for every prediction. Maximum transparency and minimal disparate impact are crucial differentiators if you must justify your models to business stakeholders and regulators.
- **Automatic model documentation (AutoDoc)**—Data scientists must document the data, algorithms, and processes used to create machine learning models for business users and regulators. H2O Driverless AI automatic model documentation relieves you from the time-consuming task of recording and summarizing your workflow while building machine learning models. The documentation includes details about the data used, the validation schema selected, model and feature tuning, MLI, and the final model created. AutoDoc saves data scientists time and removes tedious work so that they can spend more time practicing data science and drive more value for the business.
- **Bring-Your-Own Recipes**—Experienced data scientists can easily extend H2O Driverless AI with customizations that run within the H2O Driverless AI platform, including data preparation, models, transformers, and scorers. These customizations, called recipes, are Python code snippets that can be uploaded into H2O Driverless AI at runtime, like plugins. H2O Driverless AI can consume recipes with multiple convenient options: uploading from a local machine, consuming from published code in a source control hub (Bitbucket) and linking to a recipe raw code. You can check the GitHub repository for the available and optimized H2O.ai recipes. During training of a supervised machine learning modeling pipeline, H2O Driverless AI can use these recipes as building blocks with or instead of all integrated code pieces. They are used in the automatic machine learning optimization process, eventually creating the winning model. Data science teams can develop customizations specific to their use-cases, industry, or business.

# Solution architecture for AutoML

H2O Driverless AI provides an enterprise-ready AutoML product for data scientists and machine learning engineers to develop and publish AI applications. It can be deployed either in Kubernetes as pods or as a stand-alone container.

## Kubernetes-based deployment using Enterprise Steam

Enterprise Steam from H2O.ai is a service for securely managing and deploying infrastructure for H2O Driverless AI on Kubernetes. Enterprise Steam offers security, access control, resource control, and resource monitoring out of the box so that organizations can focus on the core of their data science practice. It enables secure, streamlined adoption of H2O Driverless AI and other H2O.ai products that complies with company policies.

For data scientists, Enterprise Steam provides Python, R, and web clients for managing clusters and instances. It allows data scientists to practice data science in their own H2O Driverless AI instance. For administrators, Enterprise Steam controls which product versions and compute resources are available.

Enterprise steam is a single pod that is deployed using Helm. When Enterprise Steam is deployed, you can launch a new H2O Driverless AI instance and manage existing instances.

You can use each instance for model building for a specific project. In the following figure, we show three instances of H2O Driverless AI deployed for automated model building for three different use cases: NLP, time series forecasting, and image classification.

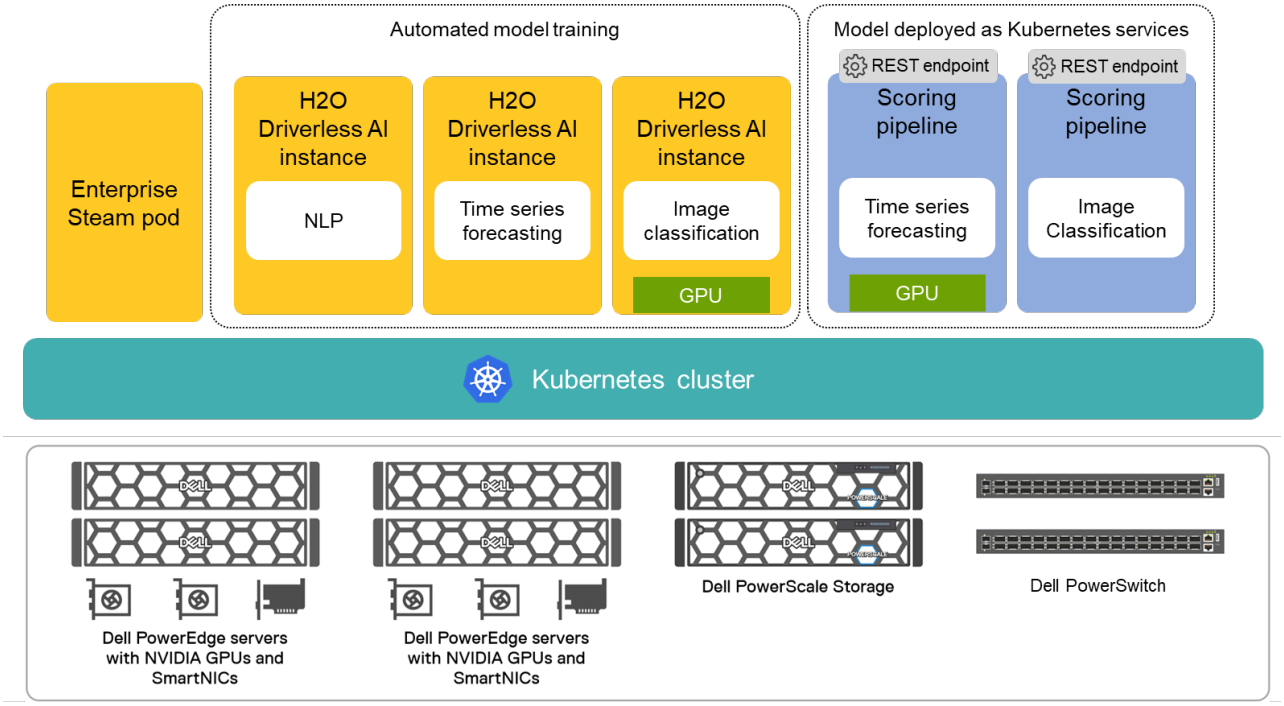


**Figure 3. Solution architecture for Kubernetes-based Driverless AI deployment**

Datasets are made available to the instance either by downloading them into the container or through several of the data connectors, as explained in the following sections. Data visualization, feature engineering, and model development are performed

on this instance. H2O Driverless AI supports NVIDIA GPU acceleration and some use cases such as image classification can benefit from GPU resources. For these use cases, GPUs are configured and made available to the container.

After the model is trained, you can download the Python or MOJO Scoring Pipeline and build a Docker container. You can deploy this Docker container outside of the Kubernetes environment or as pod exposed as a Kubernetes service.

H2O Driverless AI can also be deployed as a stand-alone container either on bare metal or virtual machines. This deployment option is outside the scope of this validated design. See the H2O Driverless AI documentation for more information.

### Docker image

H2O Driverless AI Docker images are available through Enterprise Steam. The Docker images come with all the required libraries and software installed, including libraries for the GPU.

### Security

Enterprise Steam provides access control. Users can be created with different roles, and resources can be allocated to each user. H2O Driverless AI supports client certificate, LDAP, and other authentication options. These options can be configured by specifying the environment variables when starting the H2O Driverless AI Docker image or by specifying the appropriate options in the configuration file. See the H2O Driverless AI documentation for more information.

### GPU support

H2O Driverless AI can run on machines with only CPUs or machines with CPUs and GPUs. H2O Driverless AI supports NVIDIA A100 and A30 GPUs. Only one GPU is supported per instance. Image and NLP use cases in H2O Driverless AI benefit significantly from GPU usage. Model building algorithms such as XGBoost (GBM/DART/RF/GLM), LightGBM (GBM/DART/RF), PyTorch (BERT models), and TensorFlow (CNN/BiGRU/ImageNet) models use GPU.

NVIDIA's Multi-Instance GPU (MIG) feature can be used to partition the GPUs, increase overall GPU utilization, and support several types of use cases and deployments with guaranteed quality of service. For more information about GPU partitioning recommendations, see to the NVIDIA Multi-Instance GPU and NVIDIA Technical Brief.

Image and NLP use cases in H2O Driverless AI benefit significantly from GPU usage. Model building algorithms such as XGBoost (GBM/DART/RF/GLM), LightGBM (GBM/DART/RF), PyTorch (BERT models), and TensorFlow (CNN/BiGRU/ImageNet) models use GPU.

### Storage and network configuration

H2O Driverless AI requires no special network considerations. The Kubernetes-based deployment uses ingress control and load balancers to govern access to the deployment.

H2O Driverless AI uses persistent volumes to save the required data and to connect to external data sources such as NFS.

### Licensing

H2O Driverless AI is licensed per user. Each user can deploy an instance of H2O Driverless AI. H2O Driverless AI manages the GPUs in the deployment. It ensures that different experiments by different users can run safely simultaneously and do not interfere with each other. No special licensing is required for GPU support.

Enterprise Steam is licensed separately. Users require one license per Enterprise Steam deployment.

## Invoking H2O Driverless AI from cnvrg.io MLOps Platform

As shown in Figure 1, AutoML enables automatic model building. However, it does not offer the complete life cycle for a machine learning application. Also, AutoML automated model building does not support all scenarios and use cases. For example, AutoML supports training only for supervised data and unsupervised learning. It does not support reinforcement learning.

For building models for such complex use cases and to maintain a complete life cycle of AI models, enterprises rely on an MLOPs platform. MLOps is a defined process and life cycle for machine learning data, models, and coding. The MLOps life cycle begins with data extraction and preparation as the dataset is massaged into a structure that can effectively feed the model. MLOps platforms provide constant monitoring to ensure that the process is running smoothly. MLOps enables data scientists to build complex pipelines that allow for continuous learning. Automatic retraining can be implemented to help adjust the deployed process and improve the accuracy with each iteration.

Enterprises that have multiple ongoing AI projects to support progress towards their business intelligence goals can use both MLOps and AutoML platforms to their respective strengths. Dell Technologies has worked closely with cnvrg.io to deliver MLOps for AI and machine learning adopters through a jointly engineered and tested solution to help organizations capitalize on the benefits of MLOps for machine learning and AI workloads. The *Optimize Machine Learning Through MLOps with Dell Technologies and cnvrg.io* White Paper and Design Guide provide guidance for architecting, deploying, and operating MLOps in the data center.

Data scientists can use H2O Driverless AI's Python API to access its capabilities from a MLOps platform, like cnvrg.io, as shown in the following figure. Both cnvrg.io and H2O Driverless AI are deployed on Kubernetes. The cnvrg.io platform consists of control plane pods and worker pods. The control plane pods consist of all the pods in the cnvrg.io management plane, such as application server and Sidekiq. Machine learning workloads run as worker pods (or workers). A cnvrg.io workspace is an interactive environment for developing and running code. You can run popular notebooks, interactive development environments, Python scripts, and more. From this workspace, you can invoke Python APIs to access H2O Driverless AI.

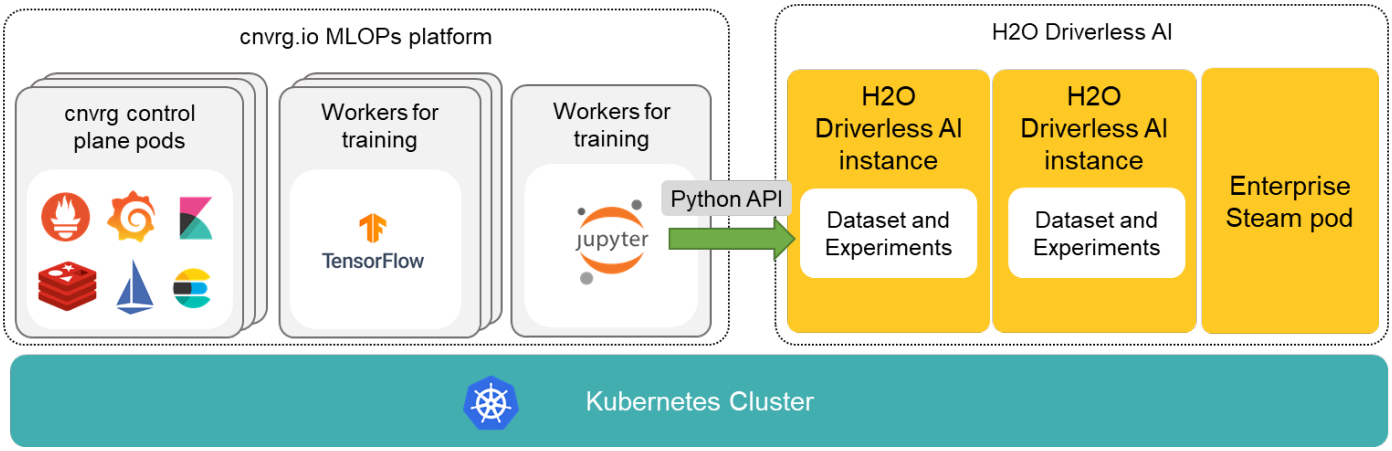


**Figure 4. Invoking a H2O Driverless AI instance from a cnvrg.io workspace**

Data scientists can perform complex data cleaning and preparation in cnvrg.io. The data can then be made available to a H2O Driverless AI instance through an NFS share. Using the APIs, data scientists can perform automated feature engineering and model development. They can evaluate the performance generated by H2O Driverless AI. The data scientists can proceed with developing their models if the performance does not satisfy them or H2O Driverless AI does not support their use case.

Data practitioners can also use cnvrg.io to push a model generated by H2O Driverless AI to production in cnvrg.io. Using the APIs, they can download all the artifacts created by the experiment. Artifacts include scoring pipelines, autodocumentation, prediction CSVs, and logs. The scoring pipelines can then be packaged as a Docker container and can be hosted in the cnvrg.io platform. The container exposes a REST API that can be used for predictions. For instructions, see Appendix A.

**Note**: H2O Driverless AI cannot further optimize models developed manually. Similarly, models selected by H2O Driverless AI cannot be further tuned and optimized manually in cnvrg.io.

Now, automated feature engineering, model development, and productizing a model from H2O Driverless AI can be part of complex machine learning pipeline in cnvrg.io allowing enterprises to build continuous learning to strengthen their AI models.

# AutoML on an optimized Dell infrastructure

Dell Technologies has worked closely with H2O.ai to deliver AutoML for AI and machine learning adopters through a jointly engineered and validated solution to help organizations capitalize on the benefits of AutoML for machine learning and AI workloads, as shown in the following figure:

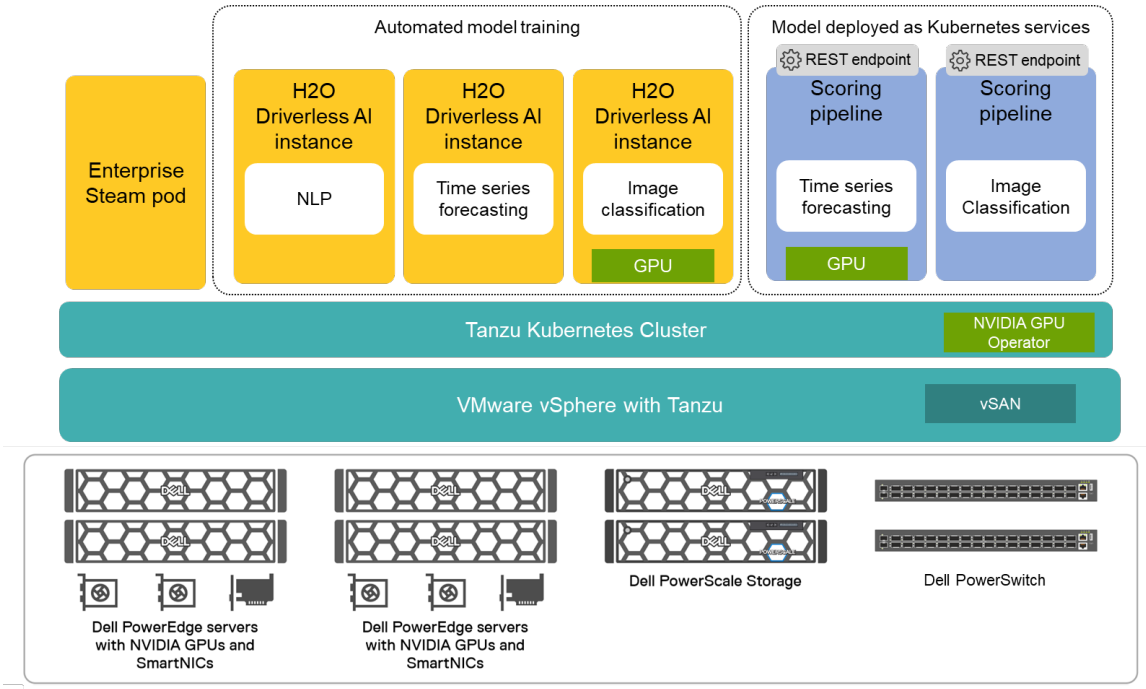


**Figure 5. Components of the Dell Validated Design for AI with VMware**

This design for AutoML with the H2O Driverless AI platform has been validated with the Dell Validated Design for AI with VMware, which is described in the Virtualizing GPUs for AI with VMware and NVIDIA Design Guide. This platform uses VMware vSphere with Tanzu for the Kubernetes layer. It uses NVIDIA AI Enterprise software for additional applications, frameworks, and tools that researchers, data scientists, and developers can use to build machine learning models and analyze data. Powered with Dell PowerEdge servers for compute (with optional NVIDIA GPUs) and coupled with Dell PowerScale storage, the solution provides the analytics performance and concurrency at scale that is critical to consistently feed the most data-hungry machine learning and AI algorithms.

**Note:** Other Dell Validated Designs for container platforms, including Red Hat OpenShift and others, can be used with similar results. However, Dell Technologies has not validated these container platforms for AutoML as of the publication of this white paper.

# Sizing of AutoML infrastructure

This section provides sizing recommendations for H2O Driverless AI platforms. Data scientists deploy individual instances for each experiment and each instance can be sized separately.

Consider several factors for sizing H2O Driverless AI deployment, such as the number of projects, the type of use cases, the number of concurrent users, and the number of active tasks. The size of the dataset drives storage and memory requirements. With its modular and container design, a H2O Driverless AI deployment can be scaled easily depending on resource use.

**CPU and GPU recommendations**—H2O Driverless AI containers with CPUs only can be used for exploratory data analysis and classical machine algorithms that do not require GPU acceleration. For example, data scientists use these templates to work on problems related to classification, regression or clustering. H2O Driverless AI benefits from multi-core CPUs with sufficient system memory and GPUs with sufficient RAM, as it can schedule and run experiments in parallel.

H2O Driverless AI containers with GPUs can be useful for building and training deep learning models. Some of the deep learning problem types include image classification and NLP. Ampere-based NVIDIA GPUs are also supported on x86 processors, as H2O Driverless AI ships with the NVIDIA CUDA 11.2.2 toolkit. NVIDIA MIG capability also enables effective partitioning of GPUs for various use cases and increases GPU use.

**Memory recommendations**—As guidance, the memory requirement per experiment is approximately five to 10 times the size of the dataset. Dataset size can be estimated as the number of `rows x columns x 4 bytes`; if text is present in the data, more bytes per element are needed.

H2O Driverless AI supports automatic queuing experiments to avoid system overload. You can launch multiple experiments simultaneously, which are automatically queued and run when the necessary resources become available.

We recommend three sizing deployments for our validated design:

- **Minimum production deployment**—The minimum size production-ready deployment is recommended for organizations that are starting their AI journey. It can support up to five projects and workspaces that perform classic machine learning or statistical modeling. This deployment assumes that these use cases do not require GPU acceleration.
- **Mainstream deployment**—The mainstream deployment is recommended for organizations that want to kickstart AI projects that might require GPU acceleration. It can support 10 projects (five of which require GPU acceleration for model building).
- **High-performance deployment**—The high-performance deployment is recommended for organizations that want to train and deploy AI models at scale. It can support up to 20 active projects (10 of which require GPU acceleration for model building).

The following table describes the recommended sizing:

**Table 1. Recommended sizing**

| Deployment | Number of H2O Driverless AI instances and total resource requirements | Recommended storage |
|---|---|---|
| Minimum production | 5 H2O Driverless AI instances with total resource recommendation:<br>• 40 CPU cores<br>• 3.2 TB of memory | 5 TB |
| Mainstream | 10 H2O Driverless AI instances with total resource recommendation:<br>• 100 CPU cores<br>• 8 TB of memory<br>• 5 A100 GPUs | 10 TB |
| High performance | 20 H2O Driverless AI instances with total resource recommendation:<br>• 200 CPU cores<br>• 16 TB of memory<br>• 10 A100 GPUs | > 20 TB |

# Validated use cases for AutoML

We validated the design presented in this white paper through two use cases: sentiment analysis and image classification. We used the workflow described in AutoML workflow with Driverless AI for these two use cases.

## Sentiment analysis with NLP

NLP is one of the key use cases supported by H2O Driverless AI. The H2O Driverless AI platform can support both stand-alone text and text with other column types as predictive features. TensorFlow-based and PyTorch Transformer Architectures (for example, BERT) are used for feature engineering and model building. H2O.ai has also validated this use case and made results available in their documentation.

We used a classic example of sentiment analysis of tweets using the US Airline Sentiment dataset. The sentiment of each tweet was labeled in advance and our model was used to label new tweets. The following table shows some samples from the dataset:

**Table 2. Example rows in dataset**

| Tweet | Sentiment |
|---|---|
| @<Airline1> it was amazing, and arrived an hour early. You're too good to me. | Positive |
| @<Airline2> hey me and my family have questions on a trip to Disney world that we are doing in April. Please follow me so I could dm you guys | Neutral |
| @<Airline2>thank you for not even coming with a solution. Great service I might say...as a TrueBlue member I am totally dissatisfied...thanks | Negative |

For the use case, we deployed H2O Driverless AI on Dell Validated Design for AI with VMware. We deployed the following components:

- VMware vSphere with Tanzu to build Tanzu Kubernetes Cluster
- NVIDIA AI Enterprise to deploy GPU operators on the cluster
- H2O Driverless AI on the Tanzu Kubernetes Cluster using Enterprise Steam

The following table describes the hardware and software configuration used for validating this design:

**Table 3. Validation setup**

| Category | Components |
|---|---|
| Servers | 3 x PowerEdge R750xa servers, each with 2 x NVIDIA A100 80 GB GPUs<br>**Note**: This validation does not use all servers or GPUs. The minimum number of servers for vSphere with Tanzu and vSAN is 3. |
| Virtualization and container orchestration | VMware vSphere 7.0U3 and VMware vSphere with Tanzu |
| AI Enterprise software | NVIDIA AI Enterprise 1.1 |
| Storage | vSAN |
| Network switches | Dell S5248F-ON (for workload and management)<br>Dell S4148T-ON OOB |
| AutoML platform | Driverless AI (1.10.2)<br>Enterprise Steam (1.8.12) |
| H2O Driverless AI instance configuration | Number of CPUs: 16<br>Number of GPUs: 1<br>Memory: 64 GB<br>Storage: 512 GB |

**Data ingestion**—The US Airline Sentiment dataset is available in Kaggle. We uploaded the dataset to the H2O Driverless AI local repository.

**Data preparation**—Using H2O Driverless AI, we split the dataset into training, validation, and test sets. We used the tweets in the 'text' column and the sentiment (positive, negative, or neutral) in the 'airline_sentiment' column for this demonstration. Because we did not want to use other columns in the dataset, we clicked **Dropped Cols**, and then excluded everything. We also used the visualization feature for a better understanding of the dataset, as shown in the following figure:

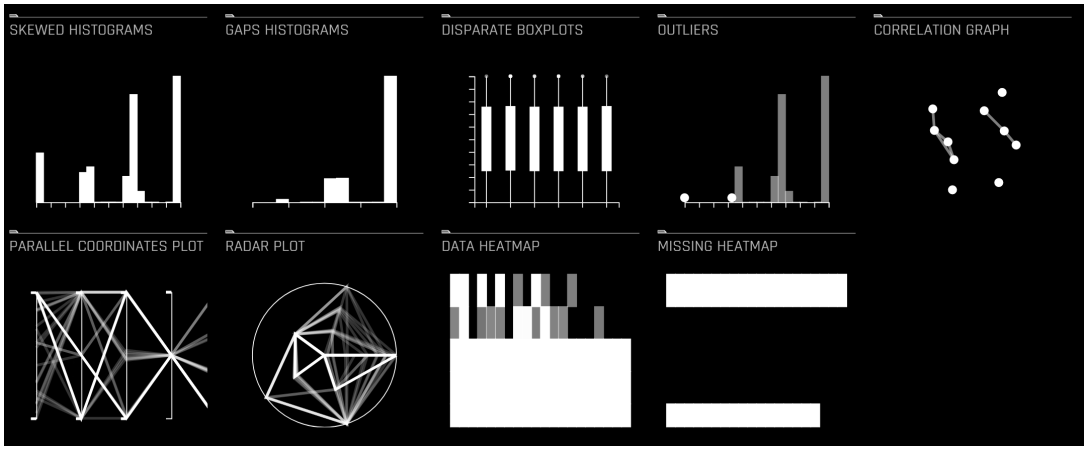


**Figure 6. Data visualization options in H2O Driverless AI**

**Model Building**—We configured the experiment using the settings shown in the following figure and then ran the experiment. Text features were automatically generated and evaluated during the feature engineering process. Some features such as TextCNN rely on TensorFlow models can be accelerated through GPUs. We verified the use of GPUs using the `nvidia-smi` command.

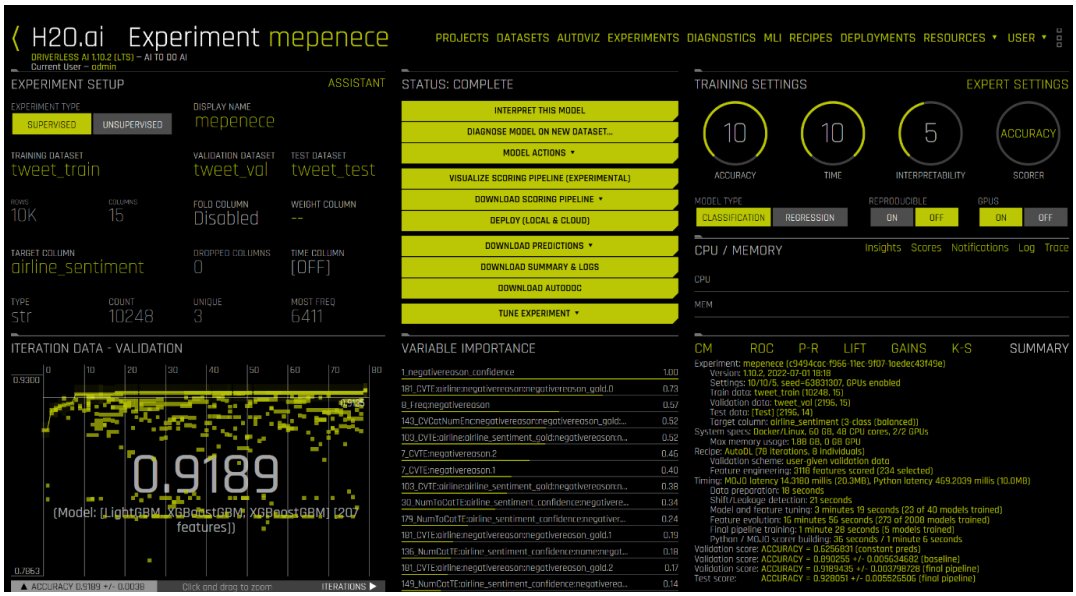


**Figure 7. Completed experiment for sentiment analysis**

**Productization**—When the experiment is completed, you can make new predictions and download the scoring pipeline as with any other H2O Driverless AI experiments. The model is then pushed to production and made available for prediction through a REST end point. We downloaded the MOJO Scoring Pipeline and built a container for it. The instructions for building the container are available in H2O's GitHub.

We used the following Dockerfile to build the container:

```
FROM ubuntu:18.04

# Install necessary dependencies ls -al
RUN apt-get update && \
apt-get -y install --no-install-recommends \
build-essential \
ca-certificates \
openjdk-11-jdk \
&& rm -rf /var/lib/apt/lists/*

# Install H2O.ai Inference server
RUN mkdir -p /opt/H2O
WORKDIR /opt/H2O
COPY /opt/H2O/DAI-Mojo2-RestServer-1.0.jar ./
COPY H2OaiRestServer.properties ./
COPY license.sig ./
COPY pipeline.mojo ./
COPY goRestServerStart.sh ./
Run /bin/sh -c 'chmod a+x goRestServerStart.sh'
ENTRYPOINT ["/bin/sh","./goRestServerStart.sh"]
```

We then built a Kubernetes service to deploy the container. Using the external IP address configured for the service, we can access the REST end point. We tested the end point with the following curl command and received the following output:

```
$ curl -X POST -d '{"fields": ["text"], "rows": [["@Airlines You are
horrible. Really bad food"]]}' http://<external-ip>/model/score -H
"Content-Type: application/json"
{"id":"92fcc278-e8e5-11ec-9e4d-
06f524030811","fields":["airline_sentiment.negative","airline_sentiment.n
eutral","airline_sentiment.positive"],"score":[["0.14810233","0.13797826"
,"0.7139194"]]}
```

## Image classification

In this validation use case, we performed image classification. The dataset consists of 1,310 images with each image belonging to one of five vehicle type categories (sedan, hatchback, pick-up, SUV, and van). Based on the images trained, the model can predict a type of the vehicle.

**Data ingestion—**We used the publicly available VTID1 dataset. We uploaded the dataset to the H2O Driverless AI local repository.

**Data preparation**—The dataset is organized in directories with directory names that indicate the label associated with the image. We split the dataset into train, test, and

validation and used it in H2O Driverless AI. We selected the target column or label (directory name) during the experiment creation process. Based on the data, H2O Driverless AI uses ImageVectorizer features for the model.

**Model building**—The following figure shows the model configuration. Because the objective is to obtain the best available model, we set the accuracy and time to 10 and interpretability to 1 in the training settings. As this use case addresses an image classification problem in which neural networks perform better in general cases, we explored the expert experiment settings to filter and use TensorFlow- and PyTorch-based models.

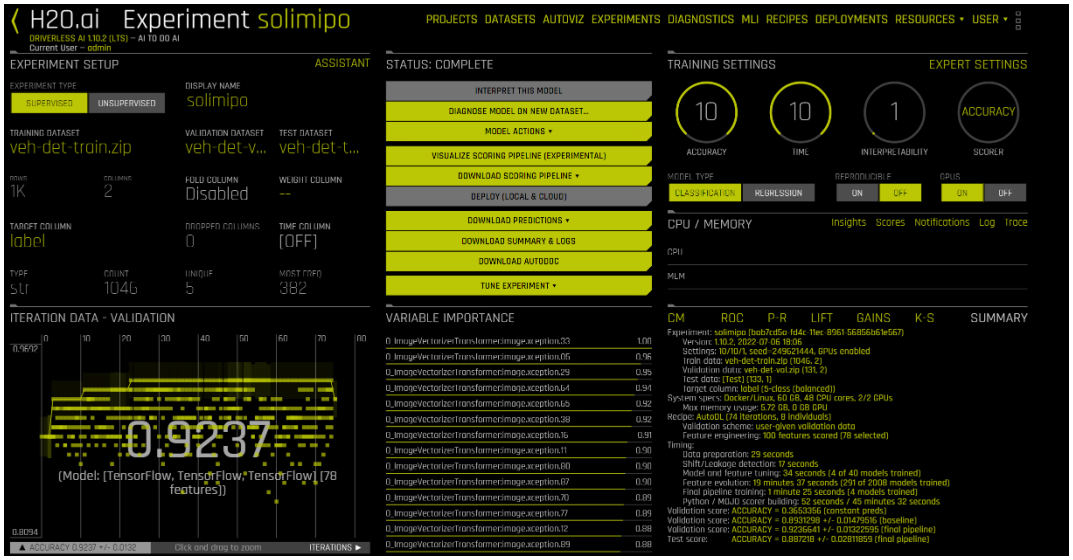


**Figure 8. Model building for image classification use case**

**Model deployment** – When the experiment is completed, you can download the scoring pipeline. A container is built similarly to the sentiment analysis use case. An API request can be sent to REST endpoint with the image array when you need to predict the vehicle type for new data.

# Dell Technologies services and support

## Deployment and support

Dell Technologies is ready to support this joint solution at every step, linking people, processes, and technology to accelerate innovation and enable optimal business outcomes.

- Consulting Services help you create a competitive advantage for your business. Our expert consultants work with companies at all stages of data analytics to help you plan, implement, and optimize solutions that enable you to unlock your data capital and support advanced techniques, such as AI.
- Deployment Services help you streamline complexity and bring new IT investments online as quickly as possible. Leverage our 30 plus years of experience for efficient and reliable solution deployment to accelerate adoption and ROI while freeing IT staff for more strategic work.

- Support Services driven by AI and deep learning will change the way you think about support with smart, groundbreaking technology backed by experts to help you maximize productivity, uptime, and convenience. Experience more than fast problem resolution—our AI engine proactively detects and prevents issues before they impact performance. Select ProSupport Plus for a single point of contact for software and hardware support.
- Payment Solutions from Dell Financial Services help you maximize your IT budget and get the technology you need today. Our portfolio includes traditional leasing and financing options, as well as advanced flexible consumption products.
- Managed Services can help reduce the cost, complexity, and risk of managing IT so you can focus your resources on digital innovation and transformation while our experts help optimize your IT operations and investment.
- Residency Services provide the expertise needed to drive effective IT transformation and keep IT infrastructure running at its peak. Resident experts work tirelessly to address challenges and requirements, with the ability to adjust as priorities shift.

# The Dell Technologies Customer Solutions Center

The Dell Technologies Customer Solution Center helps you plan and achieve your business goals to accelerate your digital future:

- **Proof of Concept**—Validate that your preferred solution meets your needs with a custom Proof of Concept. Dell Technologies solution architects enable practical, hands-on implementation based on your test cases.
- **Design Session**—Collaborate with Dell Technologies experts to design a solution framework. Brainstorm with our experts to explore your current IT environment, your future objectives, and potential solutions.
- **Technical Deep Dive**—Dive into the technical solution details that you are considering for your organization. Learn from live product demonstrations and solution-focused discussions with Dell Technologies subject matter experts.

Contact your Dell Technologies Sales Representative today to schedule a customized briefing or solutions engagement for this or any other Dell Validated Design for AI.

# Conclusion

AutoML accelerates AI initiatives by providing methods and processes to make machine learning available for both experts and nonexperts alike, enhancing an organization's ability to gain valuable insights from their data.

This technical white paper discusses automated machine learning, including the challenges of nonautomated AI model development and the benefits of automated machine learning. It provides an overview of the H2O Driverless AI product and presents a validated solution architecture for Driverless AI built on the Dell Validated Design for AI with VMware as the underlying infrastructure, including sizing recommendations for the infrastructure. It further describes how H2O Driverless AI for AutoML integrates with MLOps with cnvrg.io.

A practical real-world use case for sentiment analysis with NLP was developed and validated for this solution. The sentiment analysis is based on the analysis of tweets from the US Airline Sentiment dataset that assess positive, negative, and neutral comments from the public in real time. We also validated the solution using an image classification problem. These use cases are examples of how AutoML can be used to gain business value from a data stream.

AutoML with H2O Driverless AI on Dell infrastructure provides companies with an extensible and customizable data science platform that addresses the needs of various use cases for every enterprise in every industry.

## We value your feedback

Dell Technologies and the authors of this document welcome your feedback on the solution and the solution documentation. Contact the Dell Technologies Solutions team by email.

# References

## Dell Technologies documentation

The following Dell Technologies documentation provides additional and relevant information.

- *Dell Technologies Info Hub for Artificial Intelligence*
- *Design Guide—Optimizing Machine Learning Through MLOps with Dell and cnvrg.io*
- *Design Guide—Virtualizing GPUs for AI with VMware and NVIDIA*
- *Implementation Guide - Virtualizing GPUs for AI with VMware and NVIDIA*
- *Design Guide—Red Hat OpenShift Container Platform 4.6 on Dell EMC Infrastructure*
- *Implementation Guide—Red Hat OpenShift Container Platform 4.6 on Dell EMC Infrastructure*

## H2O.ai documentation

The following H2O.ai documentation provides additional and relevant information.

- *H2O Driverless AI*
- *Driverless AI Documentation*
- *Driverless AI NLP Demo - Airline Sentiment Dataset*
- *Driverless AI Python API Documentation*
- *H2O Enterprise Steam Documentation*

## NVIDIA documentation

The following NVIDIA documentation provides additional and relevant information.

- *NVIDIA Multi-Instance GPU and NVIDIA Virtual Compute Server*

# Appendix A – Model serving in cnvrg.io

The following steps show how to download a model training in H2O Driverless AI and push it to production in cnvrg.io:

1. Create a Docker container based on Ubuntu installed with Python 3.8.
2. Launch the Docker container using the `docker run` command.
3. In your workstation, install the H2O Driverless AI library using the following command:

   ```
   $ pip install driverlessai
   ```

4. Download the Python Scoring Pipeline using the following Python code:

   ```
   import driverlessai
   dai = driverlessai.Client(address='<IP address of Driverless
   AI instance', username='<username>', password='<password>')
   ex = dai.experiments.list()[<Experiment number>]
   artifacts = ex.artifacts.download(overwrite=True)
   ```

5. Extract the scoring pipeline ZIP file.

   The ZIP file includes all the artifacts required for model inference. The scoring pipeline includes the `run_example.sh` and `common-functions.sh` files, which installs all the required libraries in a Python virtual environment that is required for model serving.

6. Because cnvrg model serving does not support containers with a virtual environment, we need to modify these files using the following command:

   ```
   $ patch run_example.sh patchfile1
   ```

The following are the contents of the `patchfile1` file:

```
21,22c21,22
<             virtualenv -p python3.8 --system-site-packages --never-
download --copies --app-data env_app_data_dir --setuptools embed --pip
embed --wheel embed env
<             source env/bin/activate
---
>             #virtualenv -p python3.8 --system-site-packages --never-
download --copies --app-data env_app_data_dir --setuptools embed --pip
embed --wheel embed env
>             #source env/bin/activate
30,31c30,31
<             virtualenv -p python3.8 --never-download --copies --app-
data env_app_data_dir --setuptools embed --pip embed --wheel embed env ||
virtualenv -p python3.8 --never-download env
<             source env/bin/activate
---
>             #virtualenv -p python3.8 --never-download --copies --app-
data env_app_data_dir --setuptools embed --pip embed --wheel embed env ||
virtualenv -p python3.8 --never-download env
>             #source env/bin/activate
45,47c45,47
<             mv $spackagespath/xgboost $spackagespath/xgboost_h2o4gpu
<             mv $spackagespath/lightgbm_gpu
$spackagespath/lightgbm_gpu_h2o4gpu
<             mv $spackagespath/lightgbm_cpu
$spackagespath/lightgbm_cpu_h2o4gpu
---
>             #mv $spackagespath/xgboost $spackagespath/xgboost_h2o4gpu
>             #mv $spackagespath/lightgbm_gpu
$spackagespath/lightgbm_gpu_h2o4gpu
>             #mv $spackagespath/lightgbm_cpu
$spackagespath/lightgbm_cpu_h2o4gpu
61c61,62
<         source env/bin/activate
---
>         #source env/bin/activate
>         echo "just pass"
114c115
<             set_PYTHON
---
>             #set_PYTHON
169c170
< source "./common-functions.sh"
---
> source "./modified-common-functions.sh"
172a174
>
```

7. Update the `common-functions.sh` file using the following command:

```
$ patch common-functions.sh patchfile2
```

The following are the contents of the `patchfile2` file:

```
<     PYTHON=`realpath env/bin/python`
---
>     #PYTHON=`realpath env/bin/python`
>     PYTHON=`realpath /usr/bin/python3`
76c77,78
<     spackagespath=`$PYTHON -c "from sysconfig import get_paths ;
info = get_paths() ; print(info['purelib'])"`
---
>     #spackagespath=`$PYTHON -c "from sysconfig import get_paths
; info = get_paths() ; print(info['purelib'])"`
>     spackagespath="/usr/local/lib/python3.8/dist-packages"
133c135,136
<     export PYTHON=`realpath env/bin/python`
---
>     #export PYTHON=`realpath env/bin/python`
>     export PYTHON=`realpath /usr/bin/python3`
135c138,139
<     export spackagespath=`$PYTHON -c "from sysconfig import
get_paths ; info = get_paths() ; print(info['purelib'])"`
---
>     #export spackagespath=`$PYTHON -c "from sysconfig import
get_paths ; info = get_paths() ; print(info['purelib'])"`
>     export spackagespath="/usr/local/lib/python3.8/dist-
packages"
140a145
>
```

8. Run the following command to install the required libraries in the container:

```
$ apt-get update && apt install -y build-essential
libmagic-dev libopenblas-dev  git  locales  unzip  wget
$ /usr/bin/python3.8 -m pip install --upgrade pip
$ cd /driverless/scoring-pipeline
$ ./ run_example.sh
```

9. Commit the Docker container and push it to a private repository.

   This Docker container can be used in cnvrg.io for model serving.

10. Implement the model prediction as a Python wrapper function.

To push a model to production, cnvrg requires a *predict* function to implement the model prediction. H2O Driverless AI implements model prediction using a *scorer* function. Use the following example Python predict wrapper function:

```
import os
import pandas as pd
import numpy as np
from numpy import nan
```

```
from scipy.special._ufuncs import expit
# import scoring_pipeline
from <scoring experiment module> import Scorer

os.environ['DRIVERLESS_AI_LICENSE_KEY']="<License key>"

def predict(*input_text):
    """
    Function to read input texts and predict the sentiment
associated
    """
    # create scorer instance
    scorer = Scorer()
    model_input = input_text
    # predict using score using the instance
    prediction = scorer.score(model_input)
    scorer.finish()
return prediction
```

11. Follow the instructions in cvnrg.io to publish a model for serving using the Docker container and the preceding Python code.