Overview The DataRobot Automated Machine Learning product accelerates your AI success by combining cutting-edge machine learning technology with the team you have in place. The platform incorporates the knowledge, experience, and best practices of the world's leading data scientists, delivering unmatched levels of automation, accuracy, transparency, and collaboration to help your business become an AI-driven enterprise.

Use Case This guide will demonstrate the basics of how to build, select, deploy, and monitor a regression or classification model using the automated machine learning capabilities of DataRobot. The potential application of these capabilities spans many industries: banking, insurance, healthcare, retail, and many more. The use case highlighted throughout these examples comes from the healthcare industry.

Healthcare providers understand that high hospital readmission rates spell trouble for patient outcomes. But excessive rates may also threaten a hospital’s financial health, especially in a value-based reimbursement environment. Readmissions are already one of the costliest episodes to treat, with hospital costs reaching $41.3 billion for patients readmitted within 30 days of discharge, according to the Agency for Healthcare Research and Quality (AHRQ).

The training dataset used throughout this document is from a research study that can be found online at https://www.hindawi.com/journals/bmri/2014/781670/sup/. The resulting models predict the likelihood that a discharged hospital patient will be readmitted within 30 days of their discharge.
Automated Regression & Classification Modeling

STEP 1: Load and Profile Your Data

To get started with DataRobot, you will log in and load a prepared training dataset. DataRobot currently supports .csv, .tsv, .dsv, .xls, .xlsx, .sas7bdat, .bz2, .gz, .zip, .tar, and .tgz file types, plus Apache Avro, Parquet, and ORC (Figure 1). (Note: If you wish to use Avro or Parquet data, contact your DataRobot representative for access to the feature.)

These files can be uploaded locally, from a URL or Hadoop/HDFS, or read directly from a variety of enterprise databases via JDBC. Directly loading data from production databases for model building allows you to quickly train and retrain models, and eliminates the need to export data to a file for ingestion.

![Figure 1. Data Import](image)

DataRobot supports any database that provides a JDBC driver—meaning most databases in the market today can connect to DataRobot. Drivers for Postgres, Oracle, MySQL, Amazon Redshift, Microsoft SQL Server, and Hadoop Hive are most commonly used.

After you load your data, DataRobot performs exploratory data analysis (EDA which detects the data types and determines the number of unique, missing, mean, median, standard deviation, and minimum and
maximum values.) This information is helpful for getting a sense of the dataset shape and distribution. DataRobot also creates a Data Quality Assessment, which flags potential problems in your data.

Start Modeling

STEP 2: Select a Prediction Target

Next, select a prediction target (the name of the column in your data that captures what you are trying to predict) from the uploaded database (Figure 2). DataRobot will analyze your training dataset and automatically determine the type of analysis (in this case, classification).

DataRobot automatically partitions your data. If you want to customize the model building process, you can modify a variety of advanced parameters, optimization metrics, feature lists, transformations, partitioning, and sampling options. The default modeling mode is “Quick,” which employs an effective and efficient use of DataRobot’s automation capabilities. For more control over which algorithms DataRobot runs, there are Manual and full Autopilot options. You can even configure some of the default processes carried out by Autopilot.

STEP 3: Begin the Modeling Process

Click the Start button to begin training models. Once the modeling process begins, the platform further analyzes the training data to create an Importance column (Figure 3). This Importance grading provides a
quick cue to better understand the most influential variables for your chosen prediction target.

DataRobot Modeling Strategy

DataRobot supports popular advanced machine learning techniques and open source tools such as Apache Spark, H2O, Scala, Python, R, TensorFlow, Facebook Prophet, Keras, DeepAR, Eureqa, LightGBM and XGBoost. During the automated modeling process, it analyzes the characteristics of the training data and the selected prediction target and selects the most appropriate machine learning algorithms to apply. DataRobot optimizes data automatically for each algorithm, performing operations like one-hot encoding, missing value imputation, text mining, and standardization to transform features for optimal results.

DataRobot streamlines model development by automatically building and ranking models (or ensembles of models) based on the techniques advanced data scientists use, including boosting, bagging, random forests,
kernel-based methods, generalized linear models, deep learning, and many others. By cost-effectively evaluating a near-infinite combination of data transformations, features, algorithms, and tuning parameters in parallel across a large cluster of commodity servers, DataRobot delivers the best predictive model in the shortest amount of time.

**STEP 4: Evaluate the Results of Automated Modeling**

After automated modeling is complete, the Leaderboard will rank each machine learning model so you can evaluate and select the one you want to use (Figure 4). The model that is "Recommended for Deployment" is tagged so that you can begin your analysis there, or just move it forward to deployment.

![DataRobot Leaderboard](image)

**Figure 4. Leaderboard**

When you select a model, you see options for Evaluate, Understand, Describe, and Predict. To estimate possible model performance, the Evaluate options include industry standard Lift Chart, ROC Curve, Confusion Matrix, Feature Fit, and Advanced Tuning. You can even optimize your binary classification problems using a Profit Curve tool that allows you to add custom values to specific outcomes. There are also options for measuring models by Learning Curves, Speed versus Accuracy, and Comparisons. The interactive charts to evaluate models are very detailed, but don't require a background in data science in order to understand what they convey.

**Transparency**

**STEP 5: Review how your Chosen Model Works**

DataRobot offers superior transparency, interpretability, and explainability so you easily understand how
models were built, and have the confidence to explain why a model made the prediction it did.

In the **Describe** tab, you can view the end-to-end model blueprint containing details of the specific feature engineering tasks and algorithms DataRobot uses to run the model (Figure 5). You can also review the size of the model and how long it ran, which may be important if you need to do low-latency scoring.

![Figure 5. Blueprint](image)

In the **Understand** tab, popular exploratory capabilities include Feature Impact, Feature Effects, Prediction Explanations, and **Word Cloud**. These all help you understand what drives the model’s predictions.

**Interpreting Models: Global Impact**

Feature Impact measures how much each feature contributes to the overall accuracy of the model (Figure 6). For example, the reason why a patient was discharged from a hospital is directly related to the likelihood of a patient being readmitted. This insight can be invaluable for guiding your organization to focus on what matters most.
The Feature Effects chart displays model details on a per-feature basis (a feature's effect on the overall prediction), depicting how a model understands the relationship between each variable and the target (Figure 7). It provides specific values within each column that are likely large factors in determining whether someone will be readmitted to the hospital.

**Interpreting Models: Local Impact**

Prediction Explanations reveal the reasons why DataRobot generated a particular prediction for a data point so you can back up decisions with detailed reasoning (Figure 8). They provide a quantitative indicator of variable effect on individual predictions.
The Insights tab provides more graphical representations of your model. There are tree-based variable rankings, hotspots, and variable effects to illustrate the magnitude and direction of a feature’s effect on a model’s predictions, and also text mining charts, anomaly detection, and a word cloud of keyword relevancy.

Interpreting Text Features

The Word Cloud tab provides a graphic of the most relevant words and short phrases in a word cloud format (Figure 9). The tab is only available for models trained with data that contains unstructured text.
STEP 6: Generate Model Documentation

DataRobot can automatically generate model compliance documentation—a detailed report containing an overview of the model development process, with full insight into the model assumptions, limitations, performance, and validation detail. This feature is ideal for organizations in highly regulated industries that have compliance teams that need to review all aspects of a model before it can be put into production. Of course, having this degree of transparency into a model has clear benefits for organizations in any industry.

Making Predictions

STEP 7: Make Predictions

Every model built in DataRobot is immediately ready for deployment. You can:

A. Upload a new dataset to DataRobot to be scored in batch and downloaded (Figure 10).
B. Create a REST API endpoint to score data directly from applications (Figure 11). An independent prediction server is available to support low latency, high throughput prediction requirements.

C. Export the model for in-place scoring in Hadoop (Figure 12).
D. Download scoring code, either as editable source code or self-contained executables, to embed directly in applications to speed up computationally intensive operations (Figure 13).

**Monitor and Manage your models**

**STEP 8: Monitor and Manage Deployed Models**

With DataRobot you can proactively monitor and manage all deployed machine learning models (including models created outside of DataRobot) to maintain peak prediction performance. This ensures that the machine learning models driving your business are accurate and consistent throughout changing market conditions.
At a glance you can view a summary of metrics from all models in production, including the number of requests (predictions) and key health statistics:

- **Service Health** looks at core performance metrics from an operations or engineering perspective: latency, throughput, errors, and usage (Figure 14).

![Service Health](image1.png)

**Figure 14. Service Health**

- **Data Drift** proactively looks for changes in the data characteristics over time to let you know if there are trends that could impact model reliability (Figure 15).

You can also analyze data drift to assess if the model is reliable, even before you get the actual values back. You’re essentially analyzing how the data you’ve scored this model on differs from the data the model was trained on. DataRobot compares the most important features in the model (as measured by its *Feature Impact* score) and how different each feature’s distribution is from the training data.

Green dots indicate features that haven't changed much. Yellow dots indicate features that have changed but aren’t very important. You should examine these, but changes with these features don’t necessarily mandate action, especially if you have lots of models. Red dots indicate important features that have drifted. The more red dots you have, the greater the likelihood that your model needs to be replaced.
Figure 15. Data Drift

- **Accuracy** compares actual values (or ground truth) corresponding to our predictions so you can assess model performance using standard machine learning metrics (Figure 16).

Figure 16. Accuracy

From here you can apply “embedded DataRobot data science” expertise to review model performance and
detect model decay. By clicking on a model you can see how the predictions the model has made have changed over time. Dramatic changes here can indicate that your model has gone off track.

**Replacing a Model**

If you decide to replace a model that’s drifted, simply paste the URL from a re-trained DataRobot model (a model trained on more recent data from the same data source), or from one that has compatible features. After DataRobot validates that the model matches, you can select a reason why you made the replacement for a permanent archive. From this point forward, new prediction requests will go against the new model with no impact to downstream processes. If you ever decide to restore the previous model, you can easily do that through the same process.

**Prediction Applications**

Once you have deployed a model you can launch a prediction application. Simply go to the Applications tab and select an application (Figure 17).

This example shows how to launch the Predictor application. The first step is to click **Launch** in the Applications Gallery and fill out the fields below. Click **Model from Deployment** and indicate the deployment form which you want to make predictions (Figure 18). Then click **Launch**. You can also create and evaluate different scenarios based on your model with the What-If application.
Once the launch is complete you will be taken to your **Current Applications** page. You will see your application listed; this may take a moment to finish. Now you can open the application and make predictions by filling out the relevant fields (Figure 19).
Figure 19. Create New Record

**Conclusion**

DataRobot’s regression and classification capabilities are available as a fully-managed software service (SaaS), or in several Enterprise configurations to match your business needs and IT requirements. All configurations feature a constantly expanding set of diverse, best-in-class algorithms from R, Python, H2O, Spark, and other sources, giving you the best set of tools for your machine learning and AI challenges.