AutoML

September 2021

Contents

1 Introduction

This is a project-based lesson on DataRobot's AutoML platform. We balance depth with breadth in order to provide a working understanding of AutoML. Hopefully, you will a feature that inspires you to dive deeply into that topic and learn more about it using other resources like the Platform Docs, accessible through the DataRobot Platform. While DataRobot offers the flexibility of Python and R APIs, this lesson focuses on applying interacting with DataRobot through a GUI so you can see how easy it is.

1.1 Binary Classification with the Lending Club Dataset

We will work through a binary classification problem to predict a variable called `is_bad`, using real data from Lending Club, a platform for peer-to-peer lending. To motivate this self-paced lesson: Banks invest significant resources in vetting their borrowers and assessing the lending risk. DataRobot can help you do the same by building a model to predict default risk, i.e. the probability that a borrower will fail to make the remaining payments on their loan, starting at any point during the life of the loan.

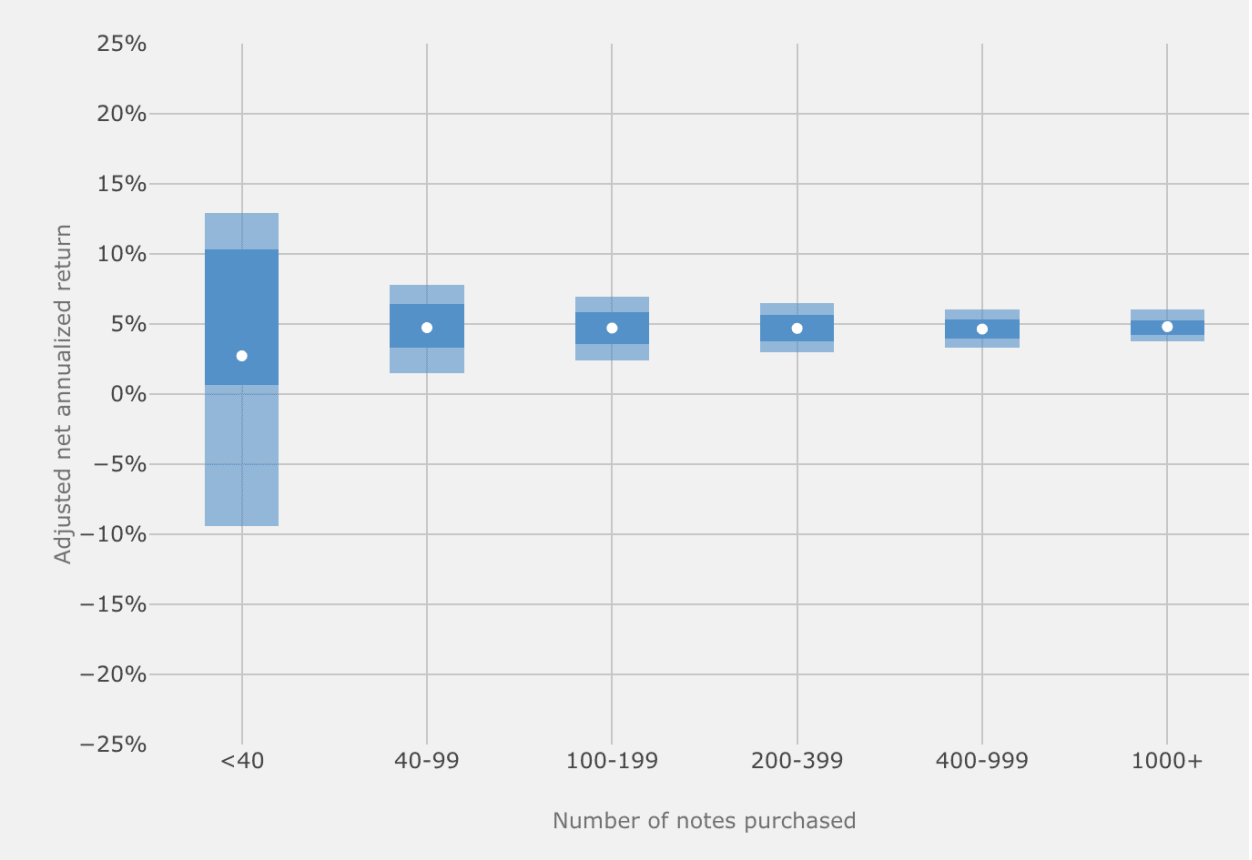


Figure 1: Investors who successfully pick the right loans to fund, taking into account default risk, are able to attain a ~10% annualized return on their investment even after expenses. We can also note that investors in this category make fewer investments and tend to have a more dispersed range of outcomes in their net annualized returns on investments

1.2 Starting a Project and Importing Data

First, you need a DataRobot account, which may be an enterprise or trial account available on:

- our public cloud https://app.datarobot.com (if located in the US)
- our public cloud https://app2.datarobot.com (if located in Europe)
- your company's internal instance of DataRobot

If you have not registered for a trial account, please do so using your work email address.

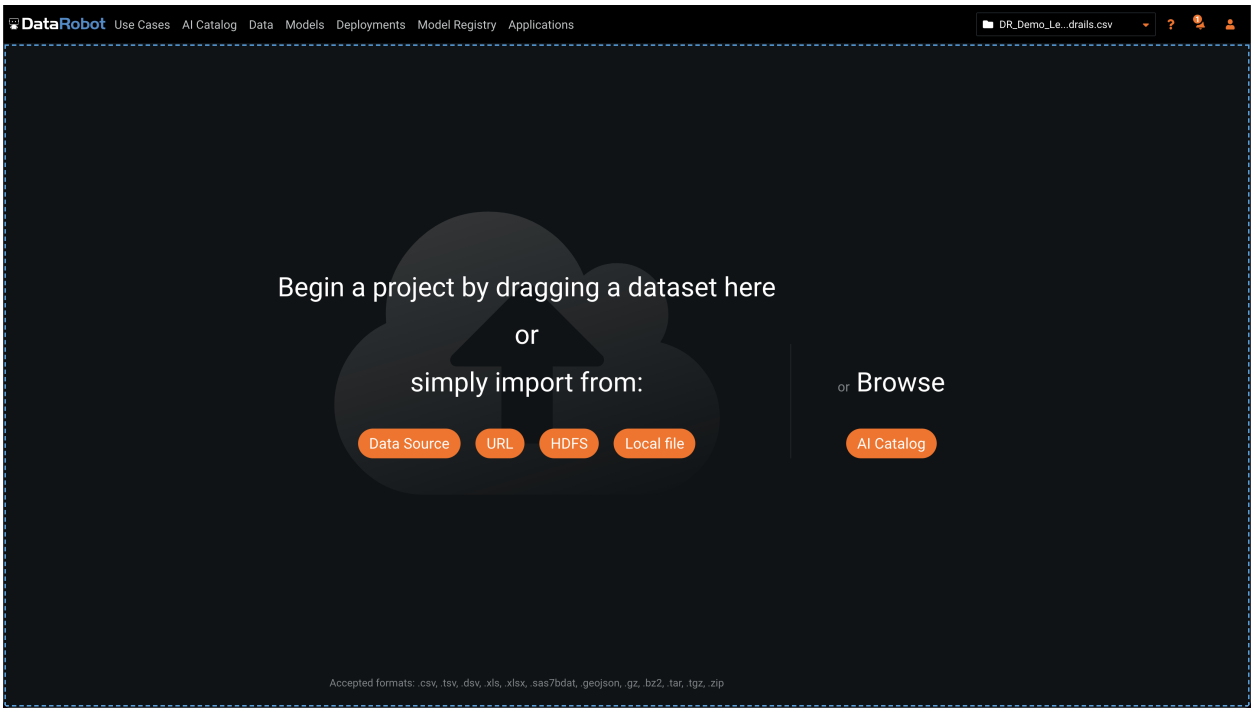


Figure 2: When you log in to DataRobot for the first time, the new project page is shown If you have built a project, you may see a different page. If so, click the DataRobot logo in the top left corner to display the new project page. At the bottom, you can see the many possible file types that DataRobot can handle. "Data Source" lets you create a connection to a database (e.g. Snowflake, .db files), "URL" allows connections to data in an AWS S3 bucket, Google Drive, or any other location accessible via a URL. "HDFS" connects data stored in distributed file systems like Hadoop. "Local files" allows uploading of locally stored `.csv`, `.tsv`, `.dsv`, `.xls`, `.xlsx`, `sas/bdat`, `.parquet`, `.avro` and several others.

1.3 Importing Data

There are several ways to upload your data:

- the DataRobot UI
- AI Catalog. The AI catalog is a great way to store, track and share pre or post-processed datasets.
- DataRobot DataPrep
- the API for R (CRAN) or Python (PyPi)

1.4 AI Catalog

The AI Catalog is a centralized collaboration hub for working with data and related assets. The DataRobot landing page provides the option to start a project via the legacy method or by using the AI Catalog.

The catalog enables seamless finding, sharing, tagging, and reusing data, decreasing time to production and increasing collaboration. Rather than work with local data files or poorly documented files in AWS S3 buckets, you upload your dataset once, have an automatically pre-populated data catalogue to start with, summary statistics and version control. It provides easy access to the data needed to answer a business problem while ensuring security, compliance, and consistency. The AI catalogue helps you:

- execute simple data preparation, leveraging SQL scripts for pinpointed results
- create datasets without the full commitment of creating projects
- find, access, delete, and reuse the assets you need
- share data without sharing projects, decreasing risks and costs around data duplication
- dramatically improve time-to-prediction through direct use of prepared and featurized datasets, which are then available to DataRobot's Batch Prediction functionality
- support data security and governance, which reduces friction and speeds up model adoption, through selective addition to the catalog, role-based sharing, and an audit trail

You can read more about loading data into the AI Catalog (here).

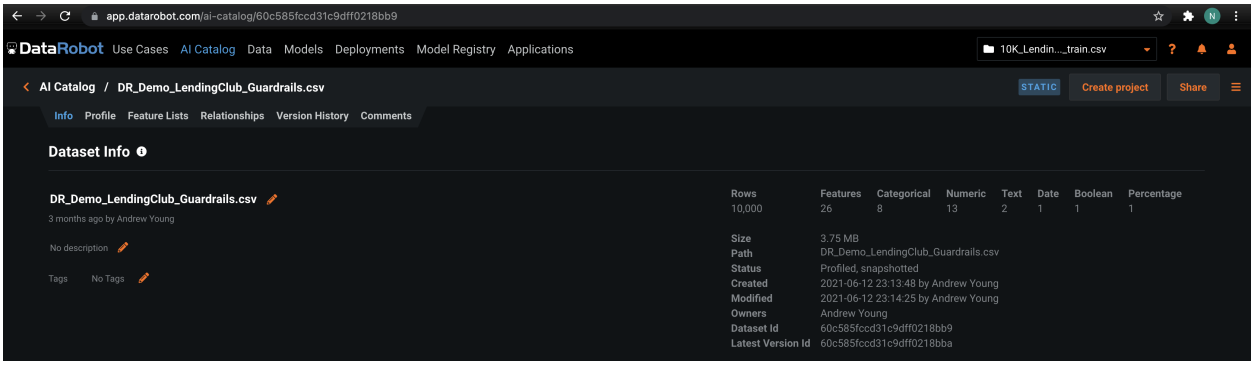


Figure 3: You can rename the dataset. I kept the default data set name derived from the file name, "DR_Demo_LendingClub_Guardrails.csv." In the top right, you have the option to "Create (a) project," using this dataset. A project is just an attempt to find the best model to predict a selected target variable. You also have the ability to share this dataset easily with colleagues, or collaborate on it. This is an improvement over the esoteric AWS S3 bucket interface.

1.5 Data Types

In addition to data consisting of dates, numerics, categoricals, summarized categoricals and percentages, your DataRobot models can process text, images, and spatial features. For each of these, DataRobot performs the appropriate preprocessing and state-of-the-art feature engineering. To elaborate, DataRobot is capable of many pre-processing operations and feature engineering transformations like missing value imputation, differencing, ratios, one-hot encoding, transformations, standardization, NLP, image featurizers, lagged features (numeric and spatial) and more.

2 Activity

You can import data into DataRobot in two ways:

- Load a local file into the AI catalog.
- Use the browser start page to import a local file directly into a new project.

2.1 Get Started

1. Using Google Chrome, download these datasets from the right-hand side of the MindTickle screen:
   - Lending Club Training Data: `DR_Demo_LendingClub_Guardrails.csv`
   - Lending Club Scoring/Test Data: `DR_Demo_LendingClub_Guardrails_SCORING.csv`
2. Login to the DataRobot platform.

2.2 Import Data

Import the dataset `DR_Demo_LendingClub_Guardrails.csv` into DataRobot using two approaches: the AI Catalog, and the new project page.

2.2.1 Part I. Load the data into the AI catalog.

1. Go to the AI Catalog tab.
2. Click Add to catalog and upload the local file. Notice the summary with the dataset ID, the dataset name, and created and modified dates.
3. Click the tabs to see the information contained in the catalog.

2.2.2 Part II. Launch a new project (click the DataRobot logo at the top left).

1. Import the training data `DR_Demo_LendingClub_Guardrails.csv` as a local file by dragging and dropping the file or uploading it using the graphical user interface (GUI).

3 Using the APIs with DataRobot

Some people prefer to start a project using a script. DataRobot allows you to interact through a REST API. There are both R and Python packages that you can install to work programmatically with the platform. Or, you can interact directly with the REST API using Curl at the terminal.

Start a Classification Project

```
In [ ]: !pip install datarobot.
```

Install Libraries and Connect to DataRobot.

```
In [1]: import datarobot as dr
        import pandas as pd
        import numpy as np

        dr.Client(token='YOUR_TOKEN',
                  endpoint= 'https://app.datarobot.com/api/v2')
Out[1]: <datarobot.rest.RESTClientObject at 0x117972e90>
```

Upload CSV file

```
In [2]: df_path = '/YOUR_FILEPATH.csv'
        df = pd.read_csv(df_path)
```

Create Project

Make Sure to change the pathname

```
In [75]: project = dr.Project.create(df_path,
                                     project_name='New Project Name')
```

Set Target and Start Autopilot

The next chunk of code sets your target and settings and starts autopilot. I have worker_count set to -1, this uses the max amount possible.

```
In [78]: project.set_target(target='readmitted',
                            metric='LogLoss',
                            mode=dr.AUTOPILOT_MODE.FULL_AUTO,
                            advanced_options= advanced_options,
                            worker_count = -1)
```

Figure 4: An example Jupyter notebook with a Python kernel. First, we `pip` install datarobot then import it and create a project using a dummy csv for a data set. You can see more example Jupyter notebooks utilizing DataRobot for predicting bad loans, modeling airline delays, etc. on https://datarobot-public-api-client.readthedocs-hosted.com/en/v2.14.0/examples/

Use our Community Github to find examples and tutorial scripts for Python. There is also a page focused on the DataRobot API at developers.datarobot.com.

4 Prepare and Explore Data

DataRobot is designed to work with tabular datasets containing most all data types suitable for this data structure. Let's look at how DataRobot prepares the data for modeling with automated feature engineering and adding additional datasets for feature discovery.

Upon uploading a data set, selecting one from an external data lake like AWS, Snowflake, etc., or choosing one uploaded to DataRobot's AI Catalogue by yourself or a colleague and added to the current project, DataRobot automatically embarks upon an exploratory data analysis.

4.1 Feature Lists

Feature Lists enable the platform to remove and add features from the modeling without altering your data. DataRobot automatically creates some Feature Lists for you depending on the type of problem and the properties of the data.

To see the list of features, select the "Data" tab to ensure you can see the "Start" button, then scroll down.

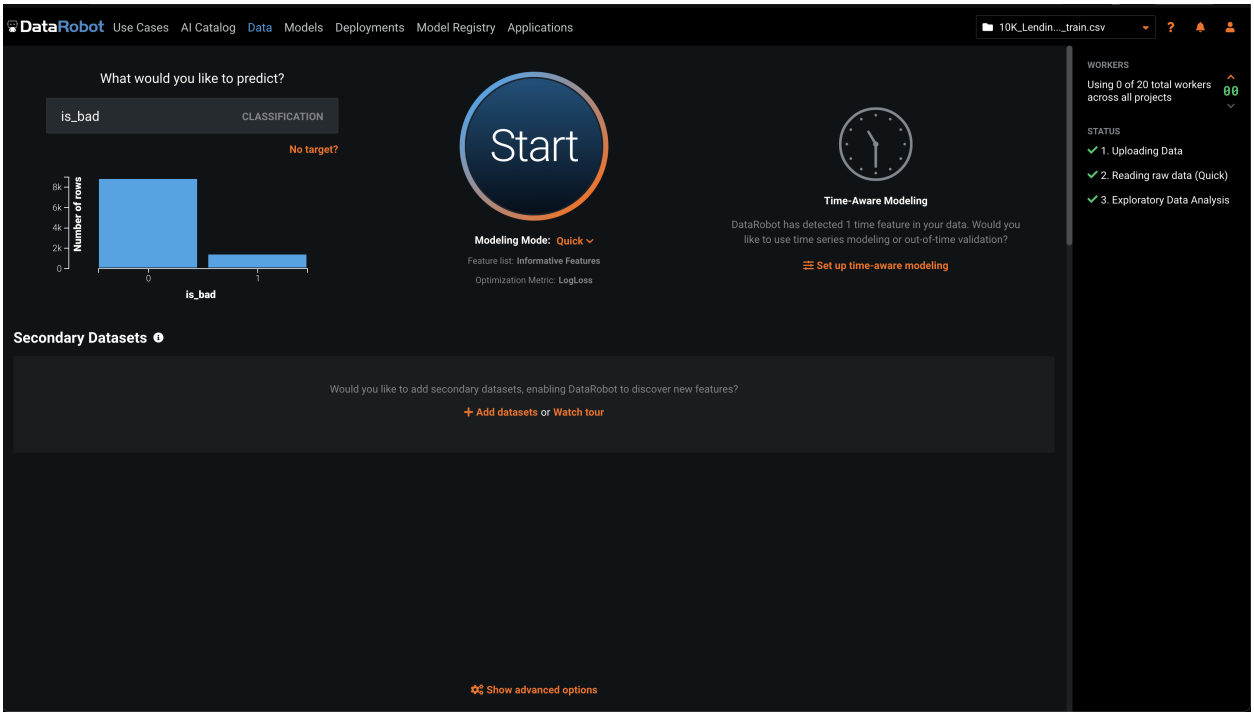


Figure 5: While looking at this view, scroll down to view the list of features, their detected data type and other summary statistics.

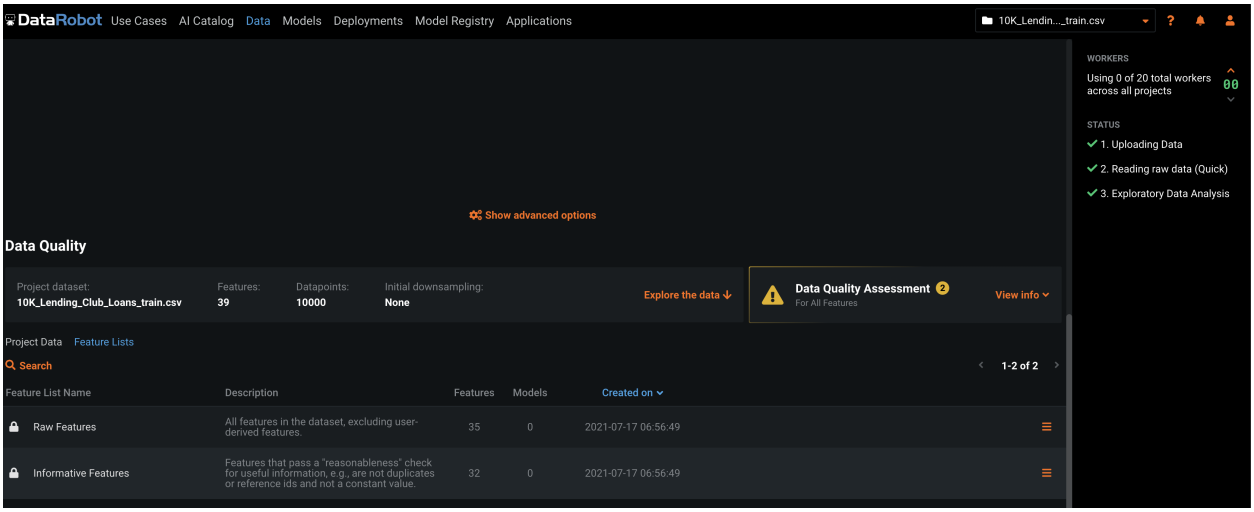


Figure 6: DataRobot automatically creates some Feature Lists for you depending on the type of problem and the properties of the data. In this screenshot, there are only two Feature lists since we haven't clicked "Start" yet.

- **Raw features**: All features in the dataset, excluding user-derived features and including those excluded from the Informative Features list (e.g., duplicates, high missing values).
- **Informative features**: This is the default feature list if DataRobot does not detect target leakage. This list includes features that pass a "reasonableness" check that determines whether they contain information useful for building a generalizable model. For example, DataRobot excludes features it determines are low information or redundant, such as duplicate columns, a column containing all ones or reference IDs, a feature with too few values, and others.
- **Informative Features - Leakage Removed**: The default feature list if DataRobot detects target leakage. This list starts with the Informative Features list and then excludes feature(s) that are at risk of causing target leakage. To determine what was removed, you can see these features labeled in the Data table with All Features selected.
- **Univariate Selections**: Features that meet a certain threshold for non-linear correlation with the selected target. DataRobot calculates, for each entry in the Informative Features list, the feature's individual relationship against the target. This list is not available until EDA2 completes.

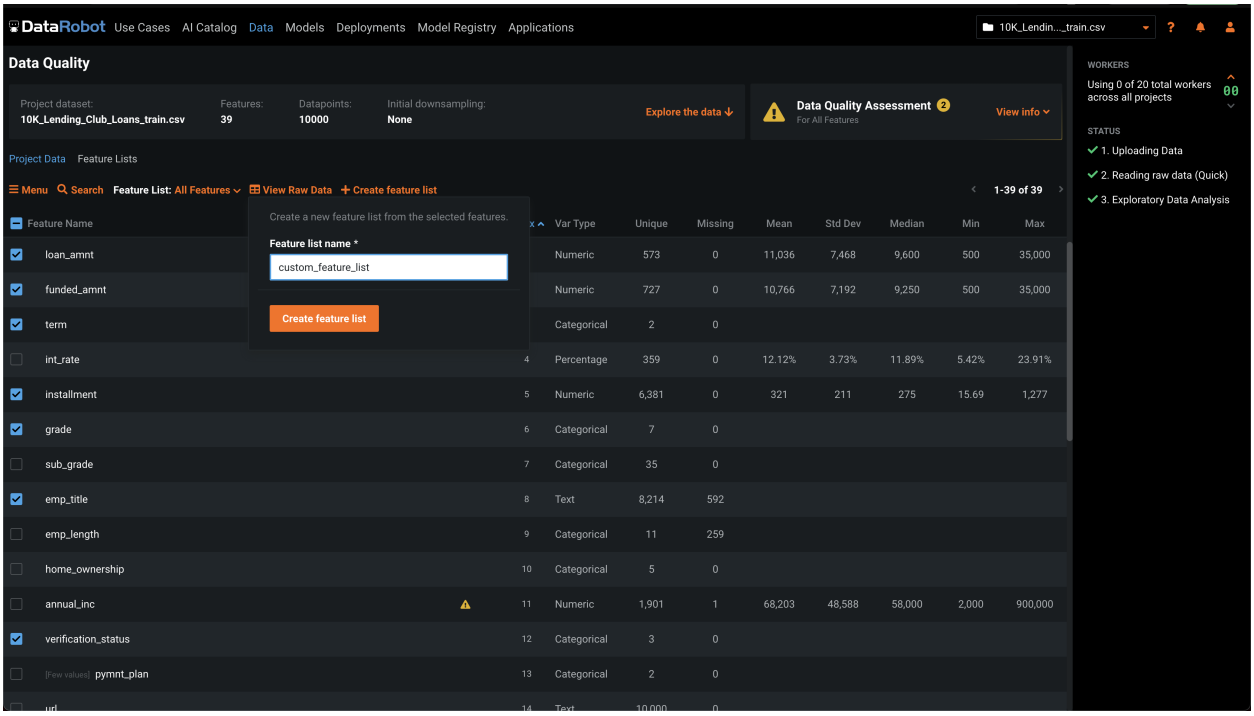


Figure 7: To create your own Custom Feature Lists select the check boxes next to the features to include. Then click “+ Create feature list.”

4.2 Exploratory Data Analysis 1 (“EDA1”)

As your data is being imported, DataRobot not only makes feature lists using subsets of the columns but also begins the first round of Exploratory Data Analysis (EDA1) process on each variable in your data. DataRobot detects the data types for each variable and shows summary statistics. After uploading `DR_Demo_LendingClub_Guardrails.csv`, you’ll notice that EDA1 has been completed and you can see the following:

- Number of unique and missing values
- Mean, median, standard deviation
- Minimum, and maximum values

DataRobot also computes a number of summary statistics for numeric variables, performs a count of terms in your text variables, and creates graphs that represent each variable’s distribution. Click any of the features to see any of these distributions.

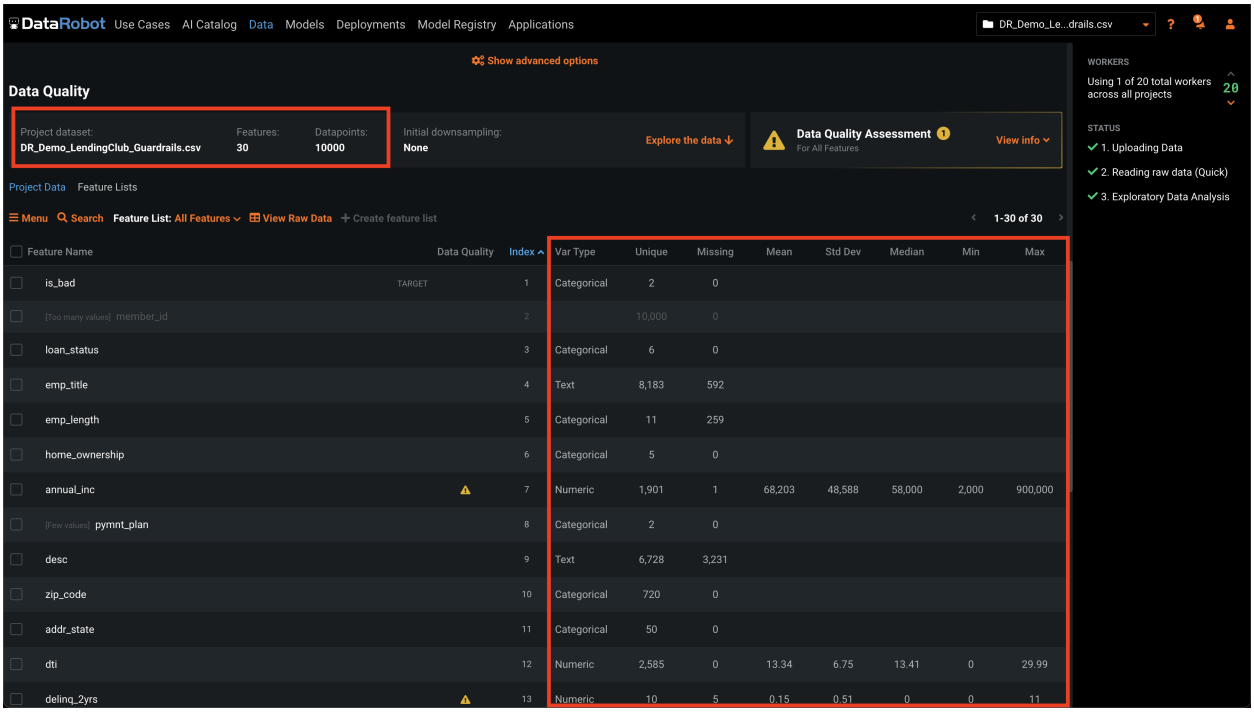


Figure 8: Without any additional action on your part, you can see the data file being examined, the shape of the data (box 1), and univariate-level feature analysis that gives the user a better understanding of each feature (box 2).

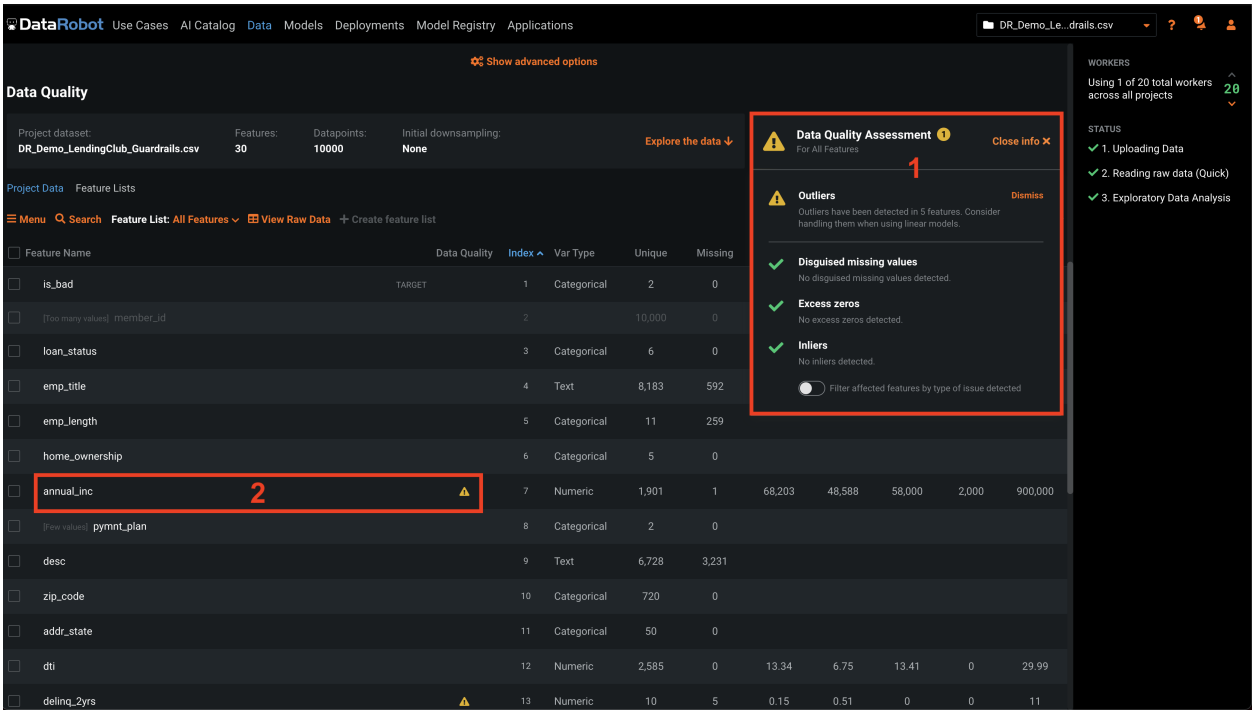


Figure 9: A Data Quality Assessment is also visible.

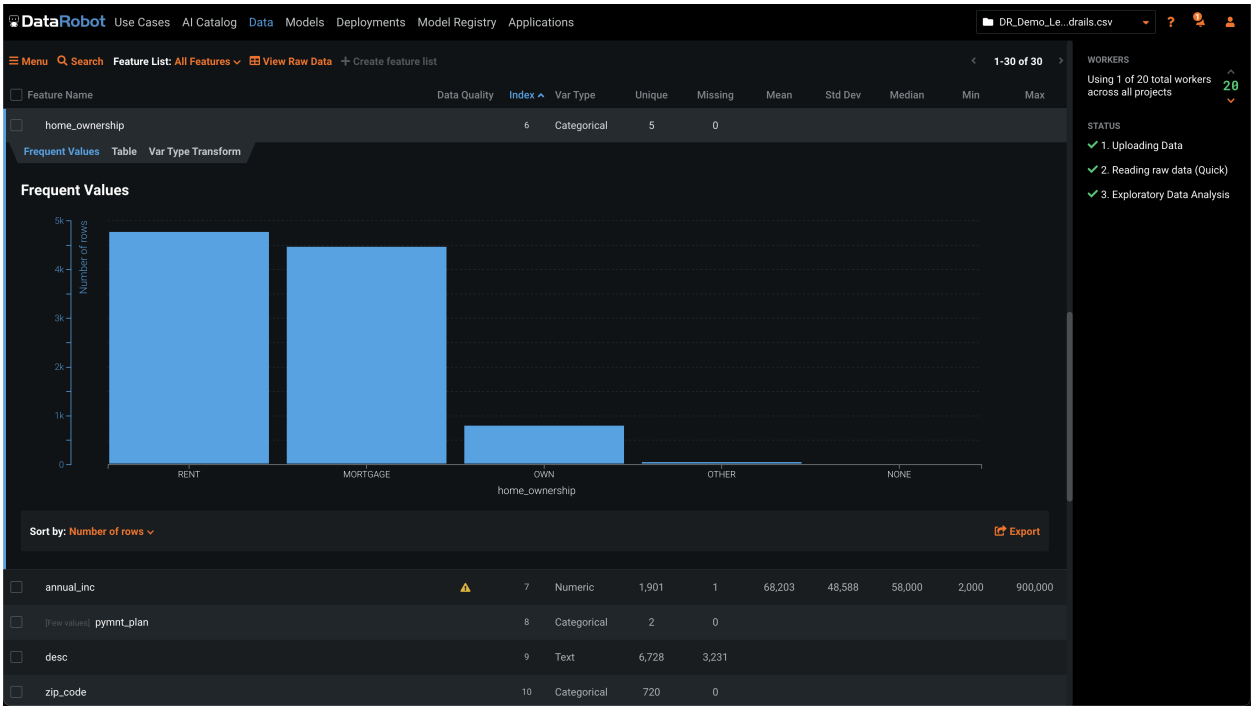


Figure 10: Additionally, you can delve deeper into any single feature. Let’s look at the `home_ownership` feature and notice the cardinality and spread of values.

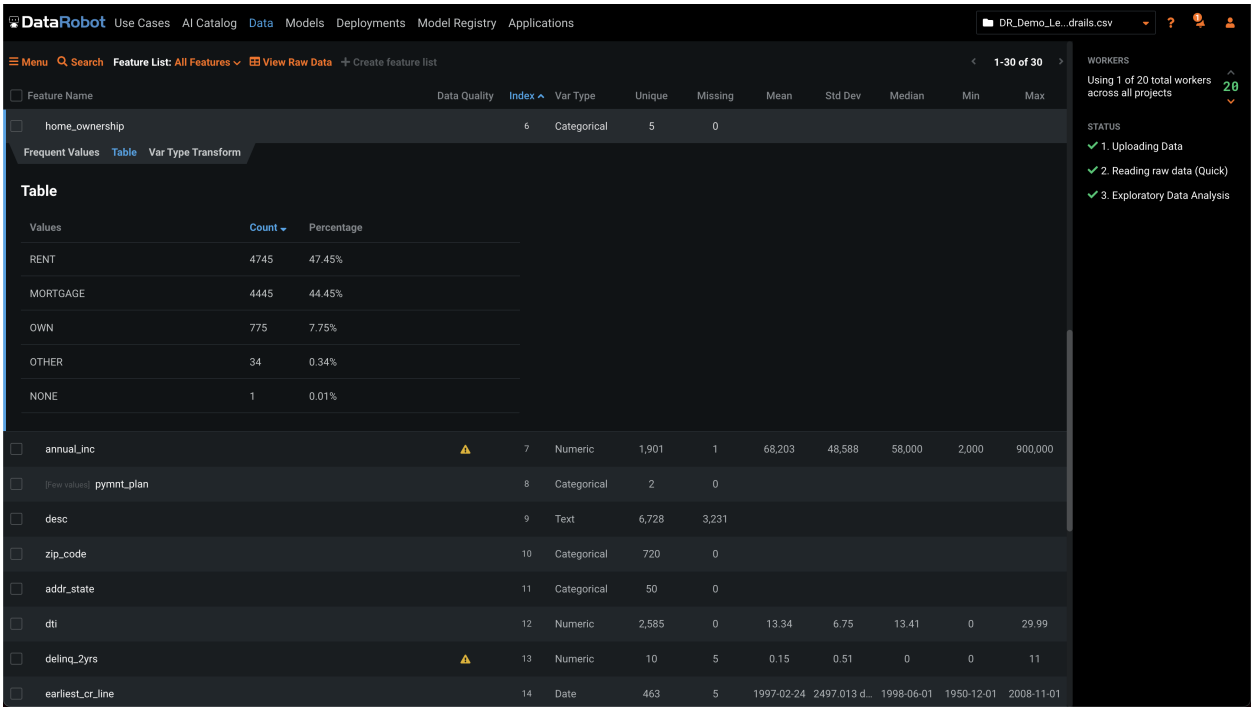


Figure 11: The `home_ownership` feature’s cardinality and spread of values is also available in table form.

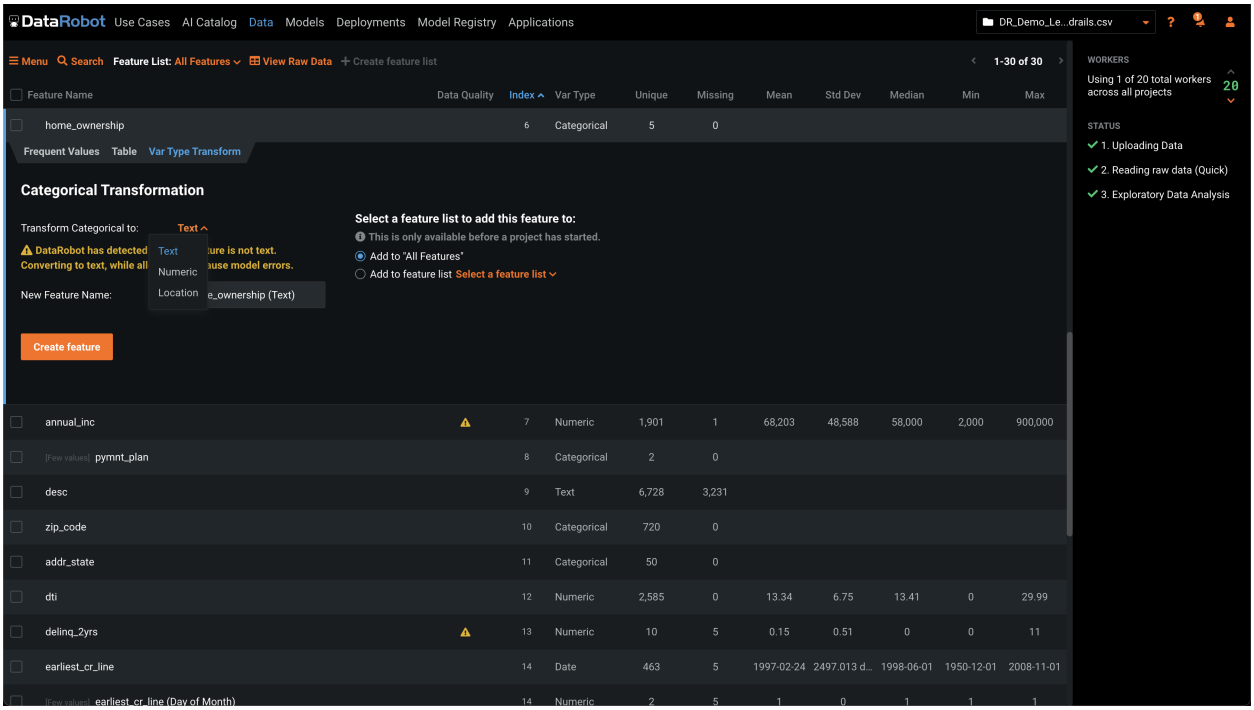


Figure 12: We can also find options to make transformations to the `home_ownership` variable. In this case, these aren't useful. In other cases, you can make useful transformations like taking the logarithm or polynomial of a numeric feature, for example.

While DataRobot automatically identifies the feature types for you and does so with great fidelity, it's good practice to ensure the variable types have been correctly identified when working with unusual datasets and before clicking "Start."

4.3 Data Quality Assessment

DataRobot has several guardrails in place to maximize success. One of these guardrails involves a data quality assessment that identifies potential data quality issues and flags:

- Target leakage (training your model on a dataset that includes information that would not be available at the time of prediction)
- Outliers
- Inliers (an inlier is a data value that lies in the interior of a statistical distribution and is in error. Inliers are difficult to distinguish from good data values so are sometimes difficult to find and correct.)
- Missing values
- Excess zeros, leading zeros, and trailing zeros
- Inconsistent gaps in time for time series projects
- Missing images and broken links for visual AI projects.

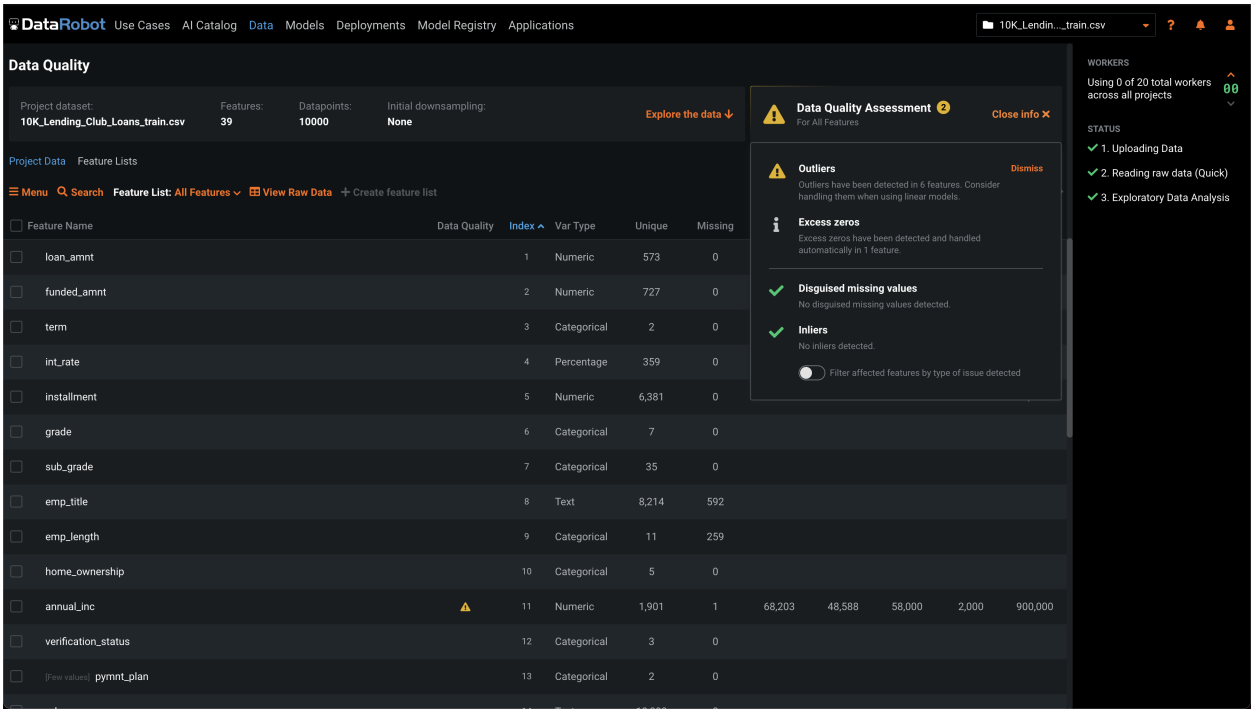


Figure 13: You can toggle a filter at the bottom of the assessment to focus on the features with potential issues.

DataRobot has strategies that can handle these data quality issues:

- Some models natively handle missing values so that no special preprocessing is needed.
- For linear models (such as linear regression or an SVM), DataRobot's handling depends on the case:
  - median imputation: DataRobot imputes missing values, using the median of the non-missing training data. This effectively handles data that are missing-at-random.
  - missing value flag: DataRobot adds a binary "missing value flag" for each variable with any missing values, allowing the model to recognize the pattern in structurally missing values and learn from it. This effectively handles data that are missing-not-at-random.
- For tree-based models, DataRobot imputes with an arbitrary value (e.g., -9999) rather than the median. This method is faster and gives just as accurate a result.
- For categorical variables in all models, DataRobot treats missing values as another level in the categories.

You can read more about DataRobot's comprehensive missing value approach and other strategies for dealing with poor data quality at the DataRobot Platform Docs.

After selecting a target and clicking Start, target leakage is assessed during the second round of EDA (EDA2) and features detected as leakage are excluded from the "Informative Features - Leakage Removed" list.

4.4 Automated Feature Engineering (AFE)

After your data has been successfully imported, DataRobot performs a number of operations on each variable in the input table to prepare it for modeling including encoding categorical variables, cleaning up missing values, transforming features, identifying potential target leakage, searching for interactions, identifying non-linearities, and so forth.

In the project from the prior section, you can scroll down to see how the features were imported into DataRobot.

For example, the earliest_cr_line feature was detected as a Date variable type, and DataRobot automatically derived features for the Month, Day of Week, and Year.

4.5 Automated Feature Discovery

Once you identify the target, you can add secondary datasets to your model. Using Automated Feature Discovery, DataRobot automatically derives a rich set of new relevant features across your project datasets to train more accurate models. This saves time when you have multiple, disparate datasets because it eliminates the need for manual feature engineering.

If you have another dataset you can add it here. In our Lending Club example, you might want to enrich the dataset to make your model as accurate as possible. Once imported into DataRobot the platform will automatically explore and derive features from that new dataset.

You can even track the feature lineage of the discovered features on the `Data` tab.

5 Build and Evaluate Models

Let's start the modeling process by selecting the target, the appropriate modeling mode, and viewing advanced options. First, enter the target `is_bad` as shown

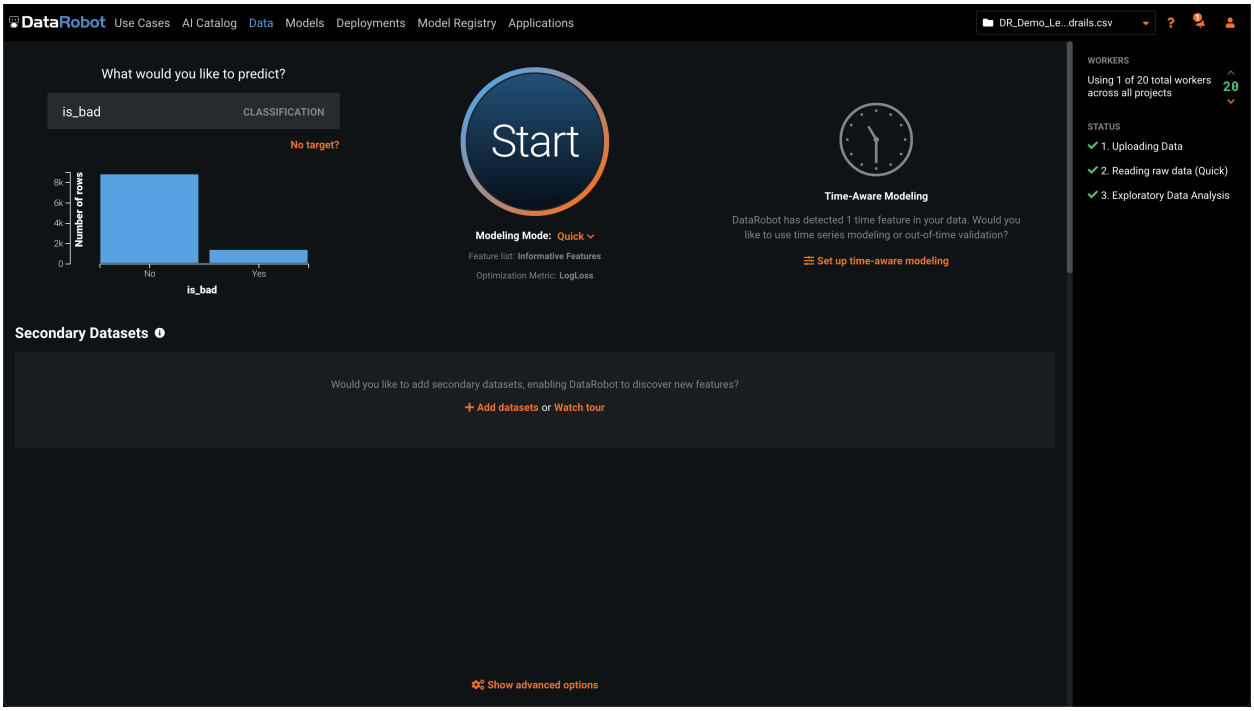


Figure 14: Enter `is_bad` in the text field on the left side.

5.1 Modeling Setup and Advanced Options

After you enter the target variable, you can set the modeling options to your desired setup. By default, DataRobot incorporates many default settings optimized through millions of datapoints from our users and experts, but you can override them to suit your needs or for the sake of experimentation.

One example of intelligent automatic settings is how DataRobot analyzed the distribution of the data, considered that we are predicting `is_bad`, and selected the `LogLoss` optimization metric as a result. In our dataset, DataRobot chose *LogLoss* because the target is a binary classification target with a highly imbalanced distribution, namely many fewer loans that eventually default ($`is_bad` = 1$).

You can customize options in the `Advanced Options` > `Additional tab`. In fact, many Autopilot settings can be altered from defaults like: whether new features should be generated from interactions found in the data, enabling SHAP blueprints, and creating "blenders" or "blends" of top performing modeling approaches. You can also set an upper bound on run time for model blueprint building, set manual weights, offsets, exposure for projects with a positive numeric target with cardinality greater than 2 and log-link metrics, and of course, random seeds for reproducibility.

5.1.1 Partitioning

In the **Advanced Options > Partitioning** tab, you have the freedom to select from traditional Partitioning strategies and have the ability to customize partitioning hyperparameters.

You can use the following partitioning methods:

- Random (Rows for each partition are selected at random, without taking target values into account)
- Partition Feature (A partition is created for each unique value of the selected feature.)
- Group (All rows with the same single value of the selected feature are in the same partition. Each partition can contain multiple values of the feature.)
- Date/Time (Partitions are ordered by time period, and there is no temporal overlap between partitions)
- Stratified (For stratified partitioning, each partition (T, V, H, or each CV fold) has a similar proportion of positive and negative target examples, unlike the previous example with random partitioning)

By default, DataRobot uses 5-fold cross-validation and 20% holdout to control for sampling bias and overfitting. You can customize these settings by changing the number of partitions, holdout size and the partitioning method. It is also worth noting that for datasets > 800MB, k-fold CV is prohibited due to computational cost and DataRobot defaults to using a Train-Validation-Holdout partitioning strategy.

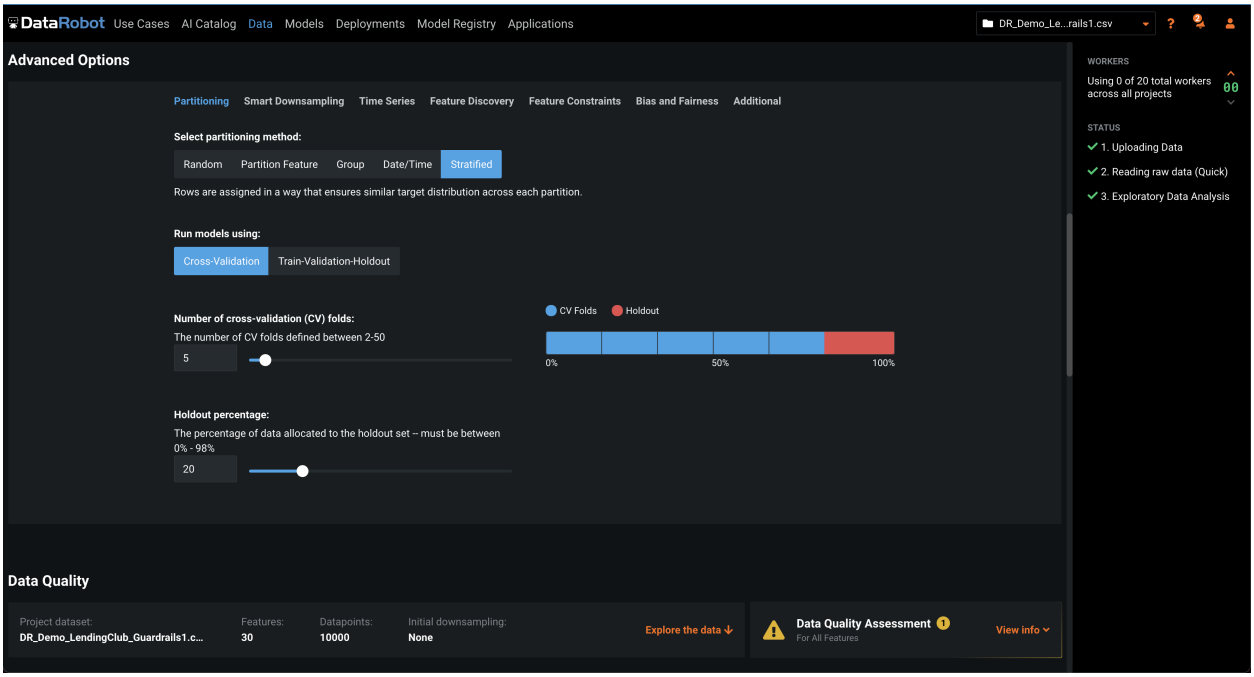


Figure 15: Advanced options for modeling gives the user the ability to customize partitioning, smart downsampling, advanced time series modeling hyperparameters, feature discovery & constraints, run bias & fairness detection for sensitive use-cases and many more.

5.1.2 Downsampling

DataRobot also offers smart downsampling is a technique to reduce the total size of the dataset by reducing the size of the majority class, enabling you to build models faster without sacrificing accuracy. This is useful for imbalanced classification problems or zero-inflated regression problems. When enabled, all analysis and model building is based on the new dataset size after smart downsampling.

As a general practice, we don't recommend downsampling to address class imbalance. There are other ways that DataRobot can overcome this obstacle that allow you to keep all of your data for training. Techniques such as using LogLoss as the optimization metric and tree-based methods can produce models that are robust in these scenarios.

5.1.3 Feature Discovery

Feature Discovery enables DataRobot to discard low importance features prior to modeling. This results in better runtime and interpretability for some models, with similar accuracy and also obeys the conventional wisdom of building parsimonious models. You can turn this off by unchecking the box.

5.1.4 Feature Constraints

Feature Constraints allow you to set up monotonic constraints on your model. Monotonic constraints (described below) control the influence, both up and down, between variables and the target. Monotonicity forces a directional relationship between the feature and the target. You can also customize the positive class assignment and enable pairwise interactions in Generalized Additive Model (`GAM`) models.

All of these modeling options are available within `Advanced Options` before you click the `Start` button. After you click the `Start` button, these settings cannot be changed for the project.

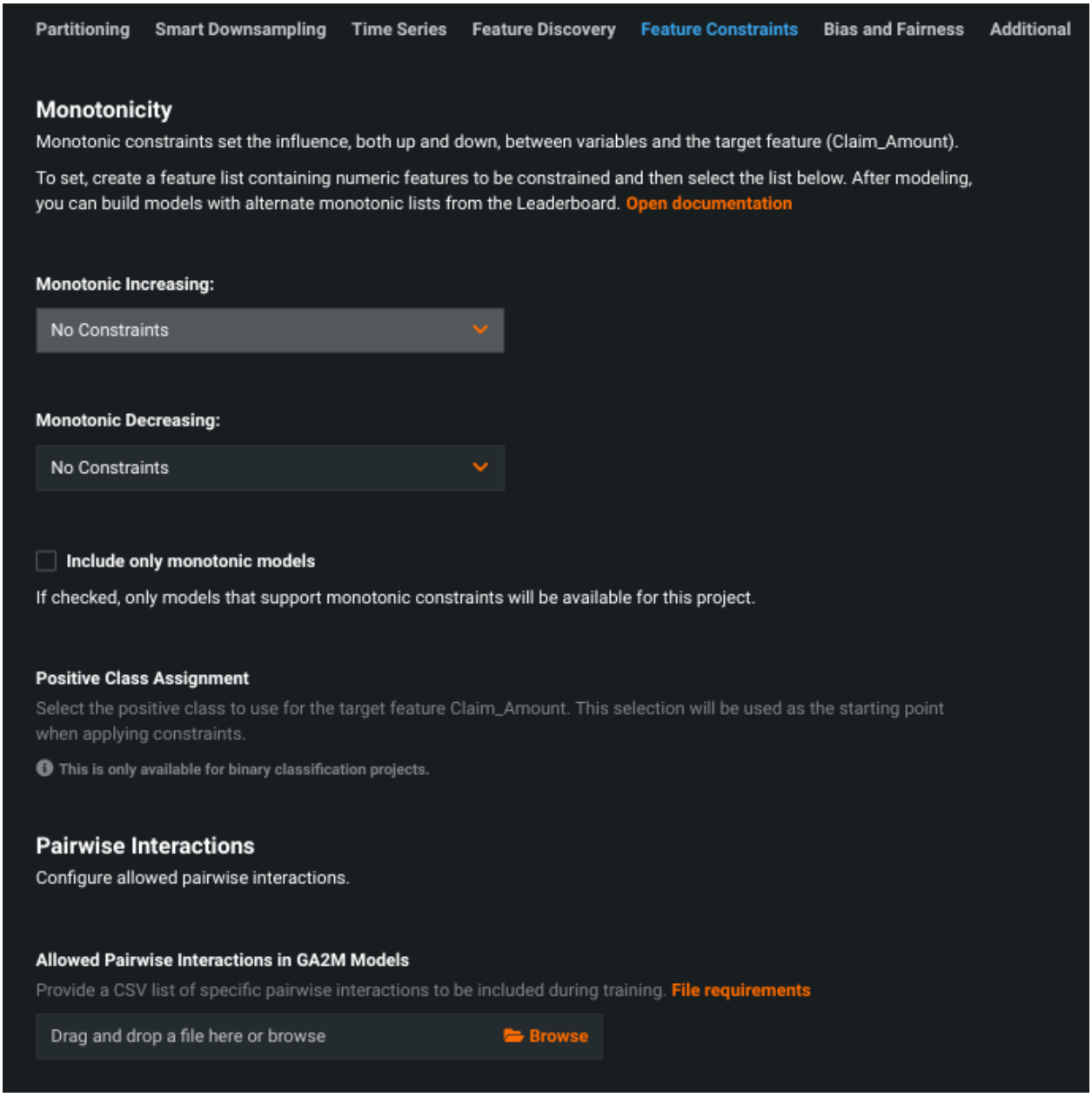


Figure 16: Set constraints on features to force directional relationships and to allow pairwise interactions for Generalized Additive Models.

5.2 Modeling Modes

The following sub-sections discuss 4 modeling modes to choose from: `Autopilot`, `Quick`, `Manual` and `Comprehensive`.

5.2.1 Autopilot

Autopilot mode automatically selects the most promising modeling approaches for the project. By default, it runs on the Informative Features list. This is the most common mode to start modeling.

The strategy for Autopilot is to try many different algorithm types using 16% of the data. The models that perform well go to the next round of modeling and are rerun on 32% of the data. The top 8 models that perform well from that round are re-attempted on 64% of the data. By default, Autopilot runs on the Informative Features feature list. Autopilot is useful when you want to generate a more diverse group of models for your specific use case (results in slower runtimes).

There is an initial Validation that occurs (Folds 1-4 vs Validation Fold). For models that make it further in the pipeline, DataRobot calculates Cross-Validation using the different folds. You can unlock the Holdout partition manually after modeling completes.

This process produces a pipeline of modeling that focuses on the most effective modeling approaches. You can retrain any model on any sample size.

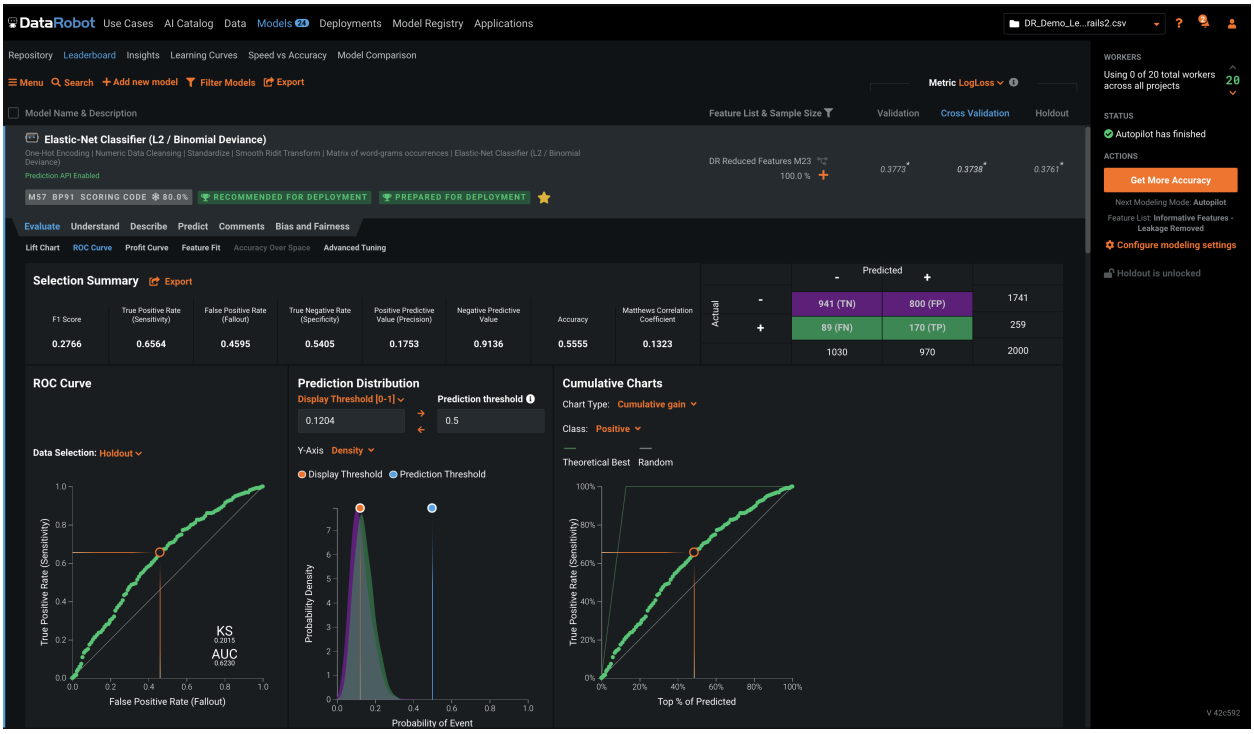


Figure 17: Once the modeling completes, you can see Lift, ROC, Profit and many other curves for eah of the models built by clicking on the model, then the `Evaluate` tab.

5.2.2 Quick

Quick mode uses sample sizes of 32% then 64% on a subset of the modeling approaches that would have run in full Autopilot mode. The intent is to quickly generate a set of models. DataRobot still uses the specified target and optimization metric to quickly provide a base set of models and insights. If the dataset contains < 5000 rows, DataRobot only performs a single run using 64% of the data.

5.2.3 Manual

Manual mode gives you full control over which models to build. For example, you can choose a specific modeling blueprint from the Repository instead of running the default choices.

5.2.4 Comprehensive

Comprehensive mode runs all Repository blueprints on the maximum Autopilot sample size to ensure more accuracy. This results in extended model build time.

5.2.5 Summary

Mode	Best used:
Autopilot	When you want to generate a more diverse group of models for your specific use case (results in slower runtimes).
Quick	When you want to generate the best models for your specific use case using a balance of speed and accuracy.
Manual	When you already know which kind of modeling approach you want to use.
Comprehensive	When you want to find the most accurate model for your use case, regardless of time. Note that this mode can result in significantly longer build times.

6 Project Checkpoint

At this point, you should already have the training dataset DR_Demo_LendingClub_Guardrails.csv data uploaded into DataRobot.

1. Enter the target is_bad (as shown earlier).
2. Scroll down and click Advanced Options > Additional tab.
3. View the different optimization metrics that you can choose from.
4. Click the Partitioning tab and view the options that you can customize.
5. Scroll back up to the top, select Quick mode under the Start button, and click Start.

7 Interpret Models

DataRobot offers many tools for model interpretability, including: Feature Impact data, Feature Effects data, Prediction Explanations, Word Clouds, Hotspots, Image embeddings, and Activation maps. These tools are located in the **Understand** tab under every model on the Leaderboard and the **Models > Insights** tab.

Before you get started, click on the Understand tab and kick off the computation for Feature Impact. This is computed for your top model, but you'll also want it for the starred 80% model.

7.1 Feature Impact

Feature Impact measures how much each feature contributes to the overall accuracy of the model.

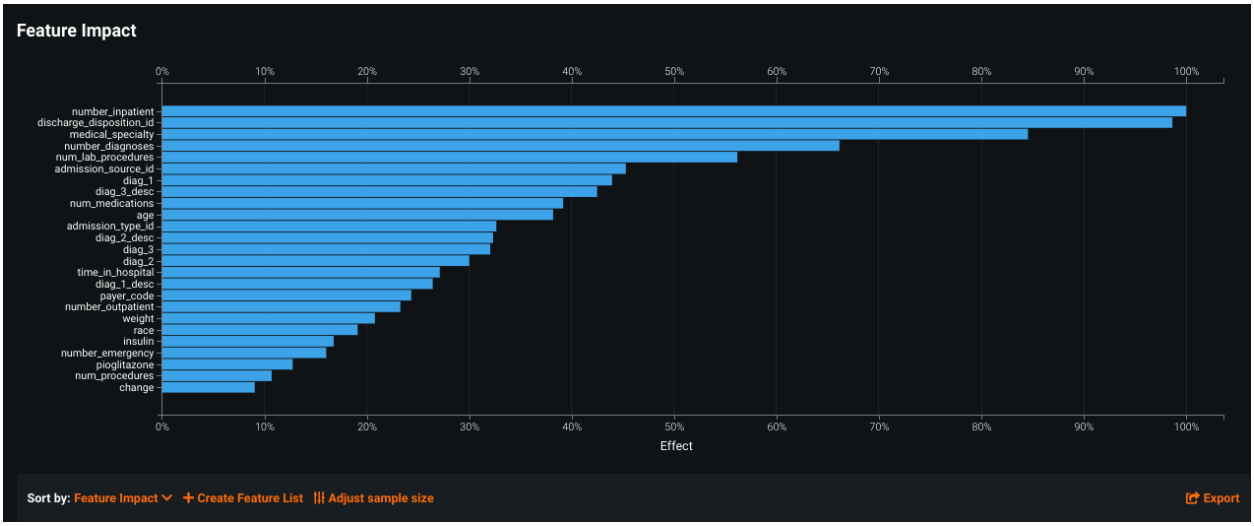


Figure 18: Here is an example feature impact chart.

DataRobot allows you to choose from 3 available methodologies for calculating Feature Impact:

- Permutation-based Feature Impact
- SHAP-based Feature Impact
- Tree-based variable importance

7.1.1 Permutation-based Feature Impact

```
Permutation-based Feature Impact measures a drop in model accuracy when feature values are shu

1. Makes predictions on a sample of training records.
2. Alters the training data (shuffes values in a column).
3. Makes predictions on the new (shuffled) training data and computes a drop in accuracy that
4. Computes the average drop.
5. Repeats steps 2-4 for each feature.
6. Normalizes the results (i.e., the top feature has an impact of 100%).
```

7.1.2 SHAP-based Feature Impact

```
SHAP-based Feature Impact measures how much, on average, each feature affects training data pr

1. Takes a sample of 5000 records from the training data.
2. Computes SHAP values for each record in the sample, generating the local importance of each
3. Computes global importance by taking the average of abs(SHAP values) for each feature in th
4. Normalizes the results (i.e., the top feature has an impact of 100%).
```

7.1.3 Tree-based variable importance

```
Tree-based variable importance uses node impurity measures (gini, entropy) to show how much ga
```

During project setup, you can configure Autopilot to run **SHAP** instead. You can also find tree-based importance measures on the `Models > Insights` tab. In this course, you'll use permutation importance rather than SHAP.

7.2 Insights

DataRobot Insights provides graphical representations of model details. Some representations are model agnostic and applicable to any model or the data as a whole, while others are representations of model details that apply to a particular model that you select.

7.2.1 Word Cloud

The Word Cloud provides a visualization of the text features and their relationship to the target. This image is the result of an NLP Autotuned Word n-Gram Text model that used a preprocessing step in the model you are evaluating. Use the dropdown to select which NLP model will use. The default should show you the 80% version of the model that you are evaluating.

- The size of a word indicates its frequency in the dataset, where larger words appear more frequently than smaller ones.
- The color indicates how it's related to the target.
- Words closer to the red end of the spectrum are associated with a higher target value, such as a 1 vs. 0 in a binary classification problem, or a larger numeric value in a regression problem.
- Words closer to the blue end of the spectrum are associated with a lower target value, such as a 0 vs. 1 in a binary classification problem, or a lower numeric value in a regression problem.

Click the `Export` button to export the word cloud, either as an image or a CSV that contains the raw values. The exported CSV gives you a number of useful fields for each text feature in your dataset, including the word, (the feature name, the strength of the correlation, and the frequency as a proportion of the total words and absolute measure.

7.2.2 Feature Effects

Feature Effects shows how a given feature impacts the overall predictive capability of the model. This is achieved using another model agnostic approach called partial dependence. You can click to compute Feature Effects in your own project now.

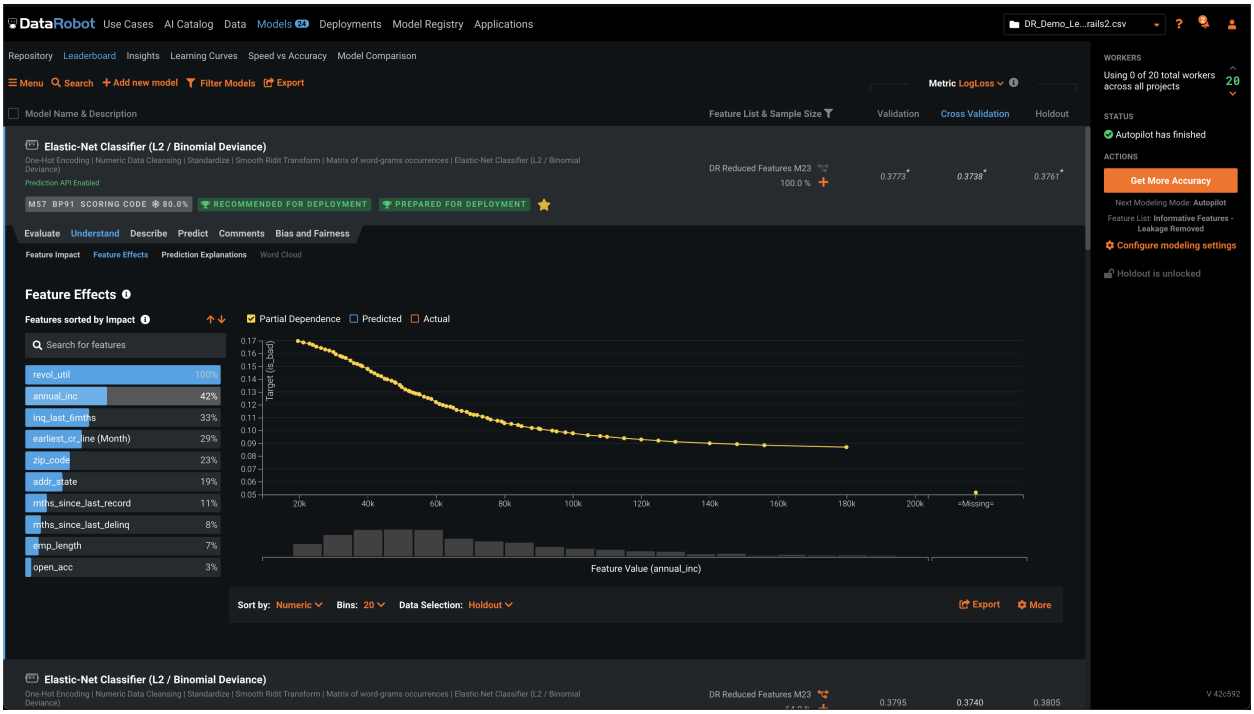


Figure 19: Here is a `Feature Effects` plot showcasing partial dependence to measure the effect of the `annual_inc` feature. Notice that as `annual_inc` increases, the likelihood of the loan defaulting decreases. This helps you understand why your features are predictive and can give you deep insights into the patterns of your data. For example, now that you know `annual_inc` is inversely related loan default, you can engineer and try to find other features that also might be related to employment or income.

7.2.3 Prediction Explanations

Feature Impact and **Feature Effects** tell you globally how your features are impacting your predictions. Prediction Explanations tell you locally how your features are impacting your predictions on a row-by-row basis. This is achieved using a proprietary model agnostic method called **XEMP**, which relies on similar strategies to permutation importance, just at a local level. During the initial project setup you could have chosen to enable SHAP based prediction explanations instead, but this course will focus on XEMP.

Click the `Understand > Prediction Explanations` tab to see a summary of the prediction explanations. A sample of three rows from the high and low end of the prediction distribution is shown. However, you can compute and download them for every row in your dataset. You can get up to 10 prediction explanations per row. Each row in our example is a loan application.

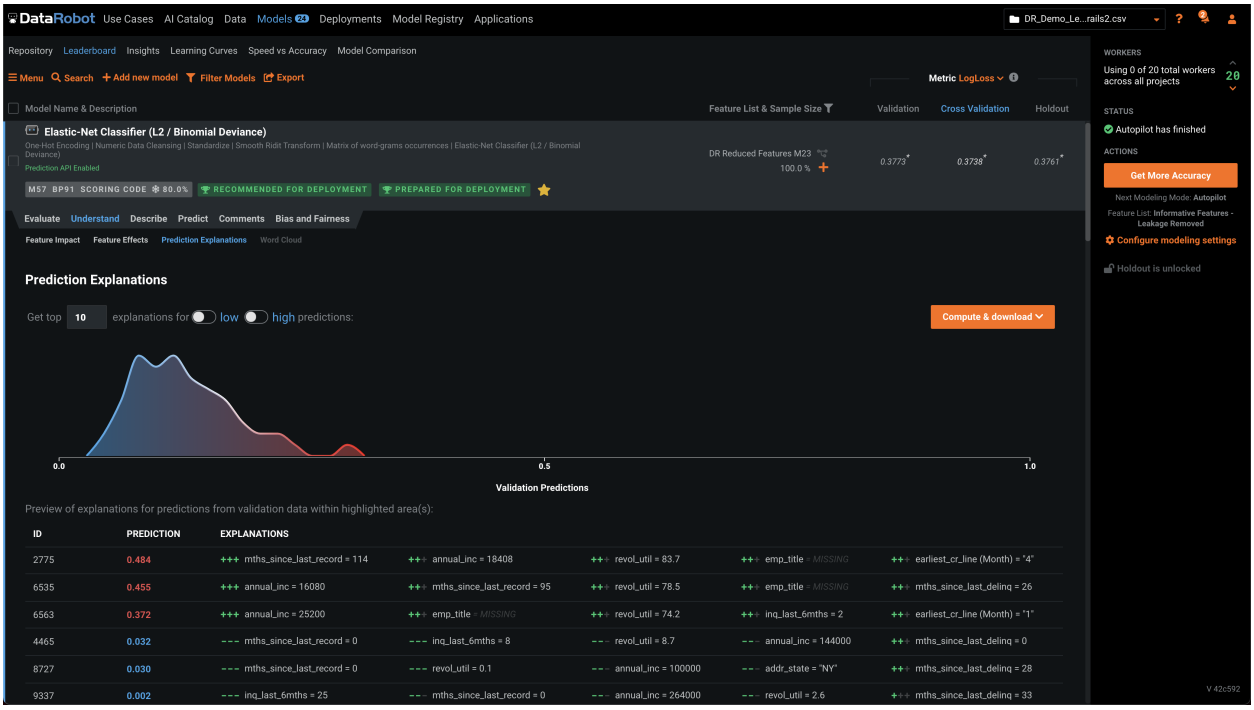


Figure 20: This screenshot shows 6 predictions with decimal numbers ranging from 0 through 1 in red or blue text. The red colored predictions sit at the higher-end of the prediction distribution, namely, closer to 1. The blue colored prediction values sit on the lower-end of the distribution, closer to 0. We can see the variables most responsible for the prediction and the values they took on. This helps build intuition on the selected model's predictions as well as trust that post-prediction analysis is always possible.

The +++ and — indicate the strength and direction of the relationship. In the first row, `annual_inc` is quite low and it is strongly pushing the prediction probability up (+++). In the last row, the feature inq_last_6mths is pushing the prediction probability down (—).

Prediction Explanations are especially useful for communicating results with non-data scientists. For example, if you have a loan officer that doesn't know anything about machine learning, you can show the prediction explanations to explain why the features are making the applications high or low risk. Because the loan officer is the end-user, the modeling scores will impact this person's day-to-day decision-making.

7.3 What's Next

Learn how to deploy a model, generate predictions in the platform, and unlock the holdout score. The Holdout column displays an evaluation metric that measures a model's accuracy against unseen ("new") data. Holdout is calculated using the trained model's predictions on the holdout partition. DataRobot reserves a portion of your data to use as holdout (20% by default); it does not train models using this data but instead validates the quality of your models once they have been trained.

8 Deploy Models and Make Predictions

You have a variety of options for model deployment. We'll focus on three:

- Generating predictions using a local file directly in the UI.
- Downloading the scoring code to score your models outside of DataRobot.
- Creating a deployment to request predictions via a REST API.

`Note: Model deployment options depend on the DataRobot installation you are using. Not all functionality is available on all installations of DataRobot.`

8.1 Model Deployment

You can make predictions directly in the UI using the `Predict` tab of a model. You can upload a local file for input to the leading ML blueprint and then download the predictions. This is typically used for ad hoc batch predictions.

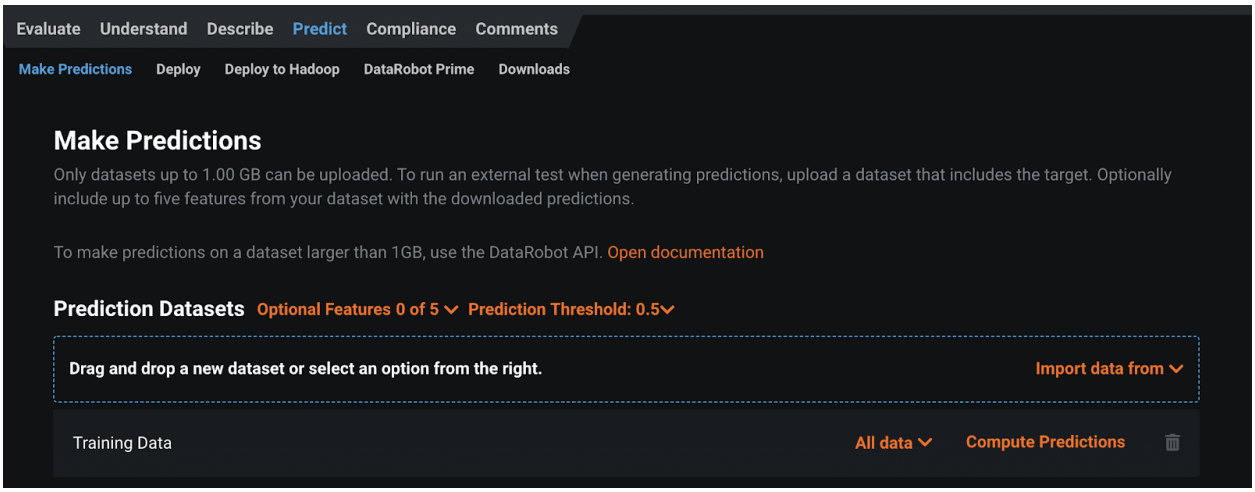
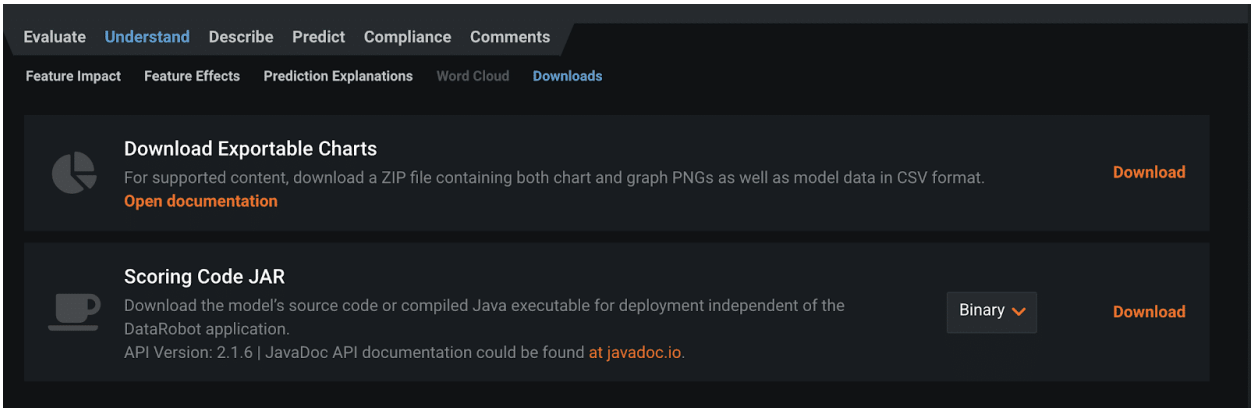


Figure 21: Make batch predictions on uploaded/imported data.

If you have actual values for the target, you can also run this as an external test set and review your prediction results within the **Evaluate** and **Understand** tabs. This enables you to use tools like the **Profit Curve** on external data for your model which is useful when you want to set penalties specific to your use case (e.g. assigning a custom amount for loss from a loan default).

8.1.1 Downloading Scoring Code

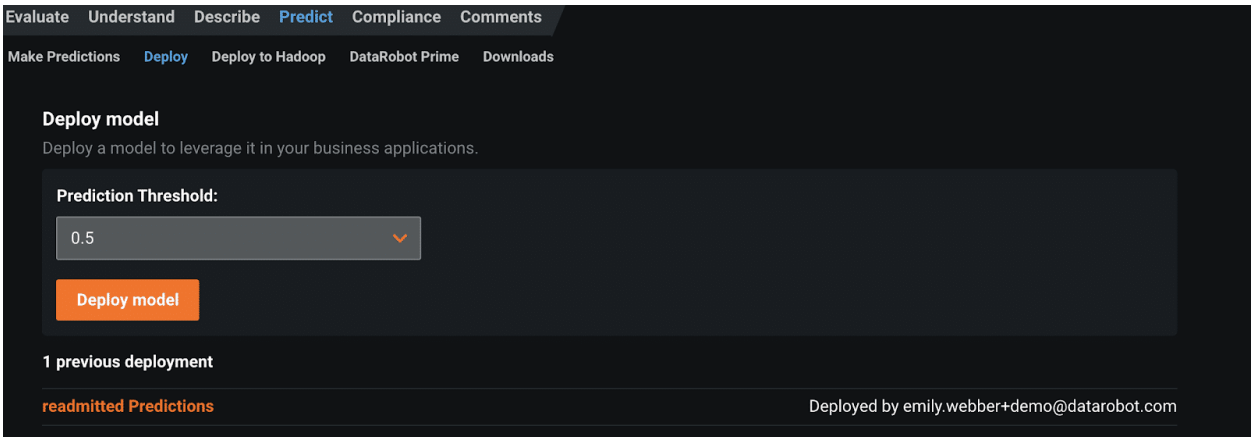
You can use the code to score the data outside of DataRobot. This is useful when you want to score your data off of a network or at a very low latency. To export scoring code in Java or Python using Codegen, use the `Understand > Downloads` tab.



8.1.2 Creating a Deployments Object

Creating a Deployment object is the most common way to set up your prediction workflow. It provides a fast way to get models into production. This enables you to deploy to an API endpoint. You can get this REST endpoint as a Docker container that you host or use a DataRobot dedicated prediction server. With either approach, you get a deployment object and can track things like Service Health, and Data Drift.

Click the `Predict > Deploy` tab to create a Deployment.



8.1.3 Unlocking the Holdout

Now, it's time to make predictions on your model. You used the 80% version of the model to evaluate the fitness and understand how it was making predictions. If you feel pretty confident about your model, you can unlock the holdout by clicking the **Unlock project Holdout** for all models link. Note that this cannot be undone, nor can you alter the models afterward.

Once the holdout is unlocked, you can see how your model performed on sample data.

In this example, the model performed similarly across the **validation, cross-validation, and holdout partitions**.

It's critical that you use the holdout only to check that the model performs well out of sample. You should NOT unlock in the beginning of your analysis to determine which model to use.

8.1.3.1 Activity

```
1. Unlock the Holdout and see if the score is consistent with the other partitions.  If it is,
2. Click the `Recommended for Deployment` model to make predictions using the model that was t
3. Click the Predict tab for the top model.
4. Upload the scoring file `DR_Demo_LendingClub_Guardrails_SCORING.csv`
5. Compute the predictions and download them.
6. Review the CSV file to determine what is included. You can add columns to the download by u
7. You can also set the Prediction Threshold with the dropdown as well. You can keep this at t
8. We are done working with the Lending Club dataset. We will take the remainder of our time t
```

8.2 Monitoring and Replacing Models

When you create a Deployment object you unlock the functionality of DataRobot MLOps. MLOps allows you to monitor and replace your deployments from the Deployments tab. Here you can monitor the number of deployments you have, as well as the number of predictions you are making. You can even monitor models you built outside of DataRobot.

8.2.1 Deployments Tab

The Deployments tab gives an overview of the models that you have deployed. At the top of this tab you can see the number of active Deployments and the number of predictions you are making. You also have a summary of **Service Health, Data Drift, and Accuracy**.

Click the any Deployment to see their details. This displays an overview page that gives you a summary, the content, and the version history of the Deployment. In each Deployment, you can also make predictions using the **Predictions** tab.

Once you've made predictions, you can monitor the Service Health, Data Drift, and Accuracy of the deployment. You can also set up notifications that tell you when your deployment needs attention and add governance roles to control access.

8.2.2 Model Replacement

To replace a model due to data drift or poor accuracy performance, go to the model's `Overview` tab. On the Menu dropdown, select `Replace Model`. Enter the URL of the new model you want to use and click `Accept and Replace`. The History block indicates a new model deployment and reason for replacement.

9 Conclusion

Congratulations on learning more about DataRobot's AutoML and working with a dataset. You have worked through a dataset using the DataRobot Platform and addressed a critical business need. However, this is hopefully just the beginning of your DataRobot journey!