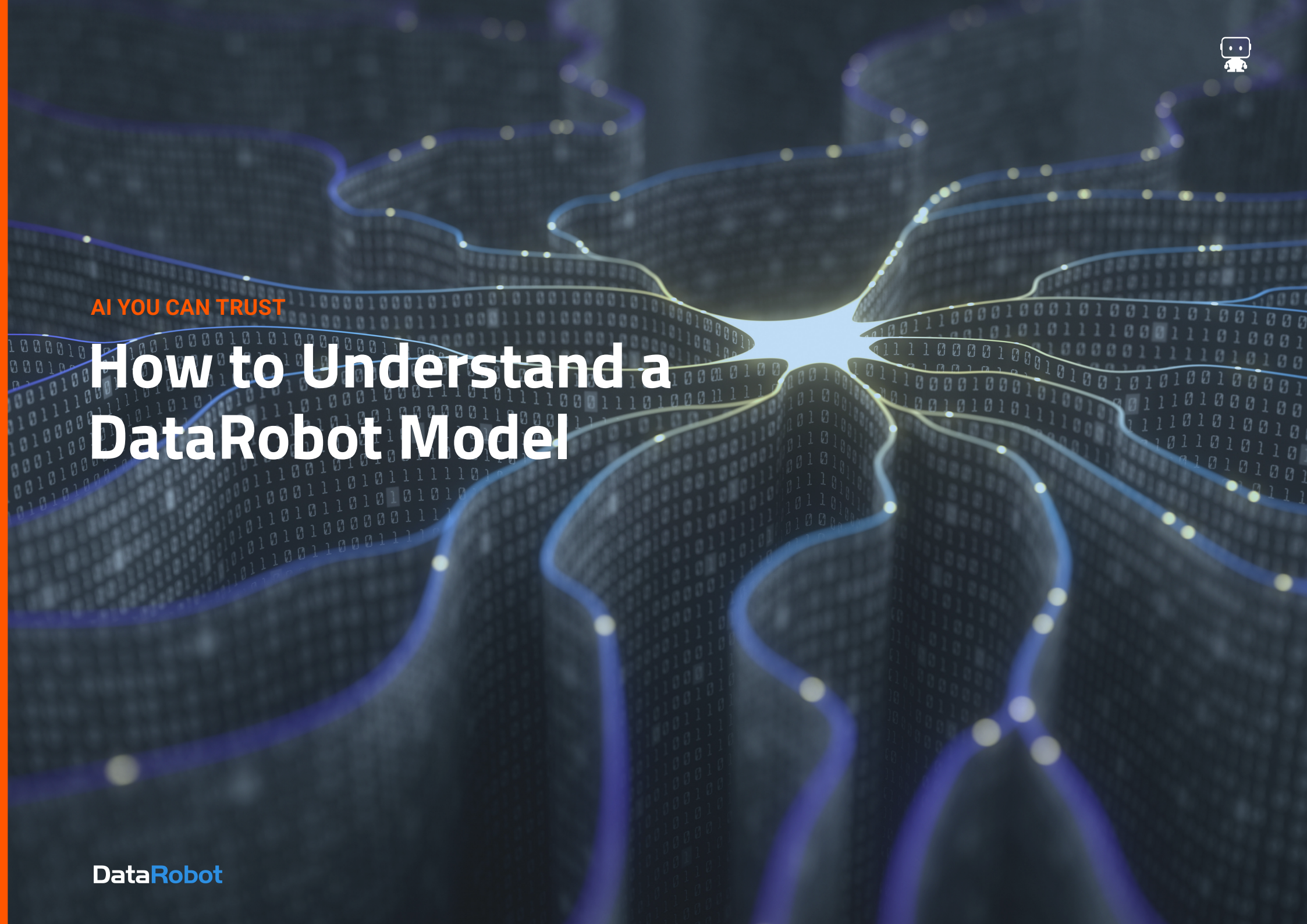
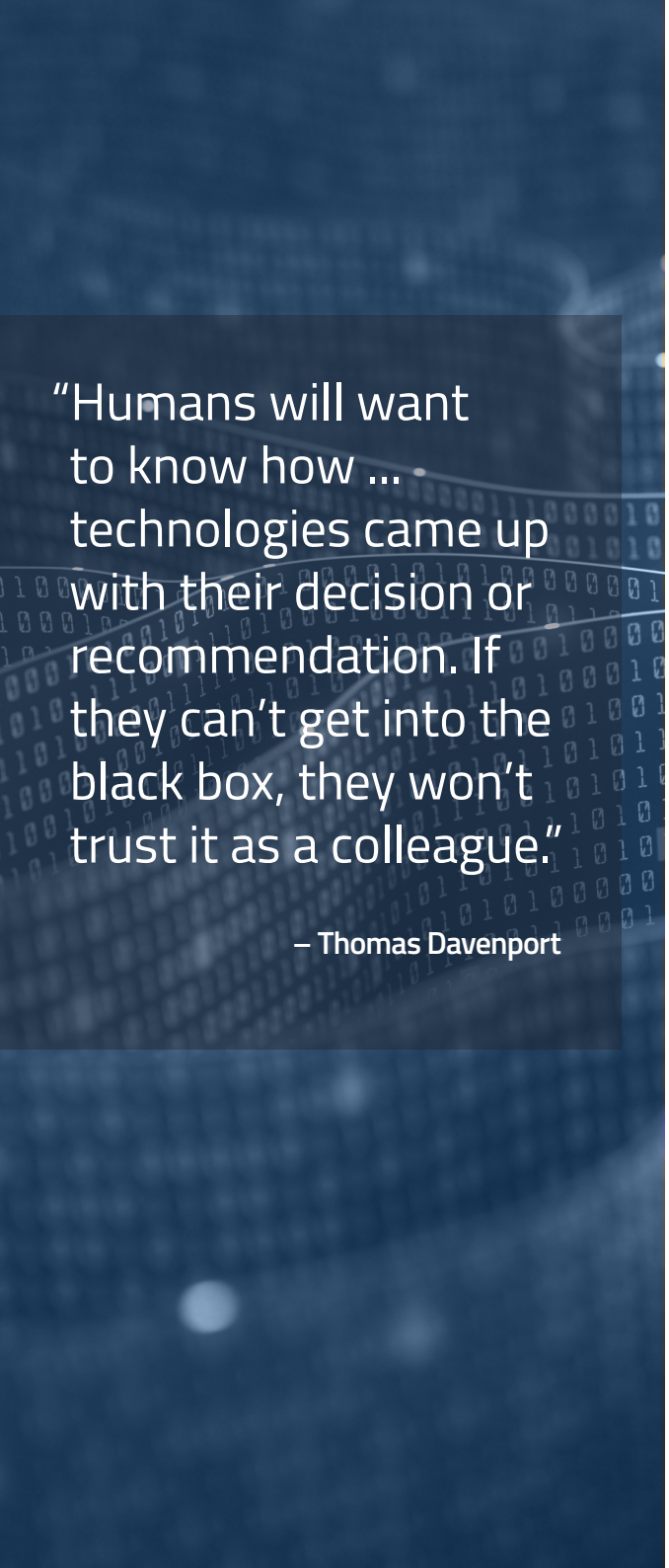


AI YOU CAN TRUST

How to Understand a DataRobot Model

DataRobot

Contents



Trust v. Understanding

In 2016, when AI was dominated by black box technologies, thought leader Thomas Davenport predicted that: “Humans will want to know how ... technologies came up with their decision or recommendation. If they can’t get into the black box, they won’t trust it as a colleague.”

At DataRobot, we also saw this need. So, we built algorithms with human-friendly explanations that can be understood by ordinary business people, and we’ve created a cheat sheet that shows you how to quickly understand a DataRobot model.

Trusting AI is a matter of understanding how well it does its job and whether it undertakes that job in a sensible manner. But understanding an AI can be as complex as understanding a human. Just as you would ask questions to a human to learn more about them, DataRobot recommends that you ask questions of your AI to better understand it.

These questions can be summarized as:

- How accurate is it? When is it most accurate and when is it not so accurate?
- What process or pipeline did it follow?
- Which data was important?
- What patterns were found in the data?
- Why did the AI make a particular decision?

The answer to each question involves a different type of explanation. Do you want to understand the model or an individual prediction that it made? These are two fundamentally different questions:

Understanding a model is about seeing whether it is accurate, and seeing what patterns the model derived from the data.

Understanding an individual prediction is about seeing why a particular data point resulted in a particular decision.

Comparing Models for Accuracy

When you want to put an artificial intelligence (AI) into production, you shouldn't just build a single algorithm. Instead, you should train multiple algorithms, and find the one that performs best on your data.

Historically, if you wanted a new AI, the first step would be to build a machine learning algorithm (machine learning algorithms power most AIs). So, a data scientist would choose a single machine learning algorithm and train it for you. Why only one algorithm? Well, that's because of the time and effort it took to manually build, train and deploy these algorithms. And since you only had a single algorithm to choose from, you weren't guaranteed to get the best.

But the days of manually building AIs are ending. Automated machine learning makes it possible to quickly and efficiently build and train dozens or even hundreds of algorithms for you to choose from.

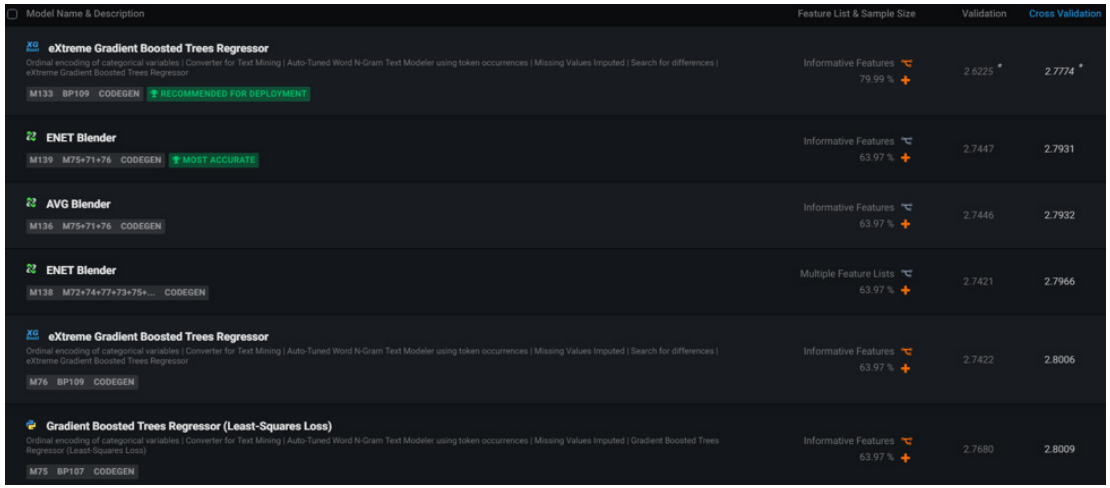
The days of manually building AIs are ending.

How to Interpret
The Leaderboard

1. DataRobot has flagged the ENET Blender as the most accurate model on your data. The ENET Blender was trained on 64% of the rows and used the default list of input features (Informative Features).

2. DataRobot has flagged the eXtreme Gradient Boosted Trees Regressor as recommended for deployment, because it has accuracy almost equal to the most accurate model but runs much faster. This model was retrained on 80% of the rows in order to give it an extra boost of accuracy, and it used default list of input features (Informative Features).

DataRobot Leaderboard



In the same way that competition is good for people, it is also good for AI and machine learning algorithms. According to the "no free lunch" theorem, no algorithm is always going to be the best. You may have your favorites, but step aside and let competition find the most accurate. DataRobot's leaderboard does just that. The most accurate models are listed at the top, the least accurate at the bottom.

The above screenshot shows the top six models from a leaderboard. The full leaderboard for this project contained 90 models, but the actual count of leaderboard models will vary from project to project.

On the left of the leaderboard, each trained model is listed, along with its description. You can click on the model to expand it and learn more details. To the right are details about which features or input variables the model used, how much data it was trained on, and its accuracy scores. All models have a validation accuracy.

To make it even easier for you to choose a model, DataRobot automatically flags which model is most accurate, and which model is best suited for going into deployment. But you may have more considerations than just accuracy. So, you can choose to deploy any model from the leaderboard. You can even choose more than one.

How to Interpret
The Lift Chart

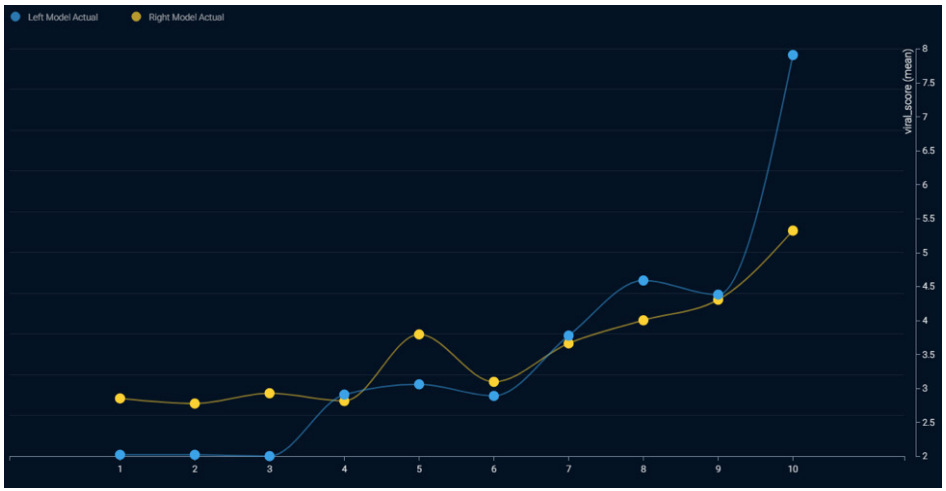
1. The blue line has a steeper slope than the yellow line. Therefore, the model represented by the blue line is more accurate than the model represented by the yellow line.

2. The blue line is particularly lower for the first 30% and higher for the last 10% of rows ranked by predictions. This implies the the model represented by the blue line is particularly stronger at predicting lower values, and at predicting the small number of extremely high values.

Model Comparison

If you're more of a visual person than a numbers person, then model comparison plots are for you. These plots compare any two models of your choosing from the leaderboard, and visually compare their accuracy.

Model Comparison - Lift Charts



The first comparison we can make between any two models is by plotting the lift charts for each. Lift charts communicate accuracy by displaying how well a model can separate high values (e.g., finding those customers most likely to purchase your product) from low values (e.g., finding those customers not suited to a product offering). Here the blue line represents one model, and the yellow line represents a different model, both selected by the user.

DataRobot makes predictions for each row, and then ranks each row from the lowest prediction to the highest prediction. Finally, it plots average results from the rows with the lowest predictions (on the left) up to the rows with the highest predictions (on the right). The more accurate model is the one where the line has the steepest slope, and the widest vertical range, because that means the model is correctly separating high values from low values in its predictions.

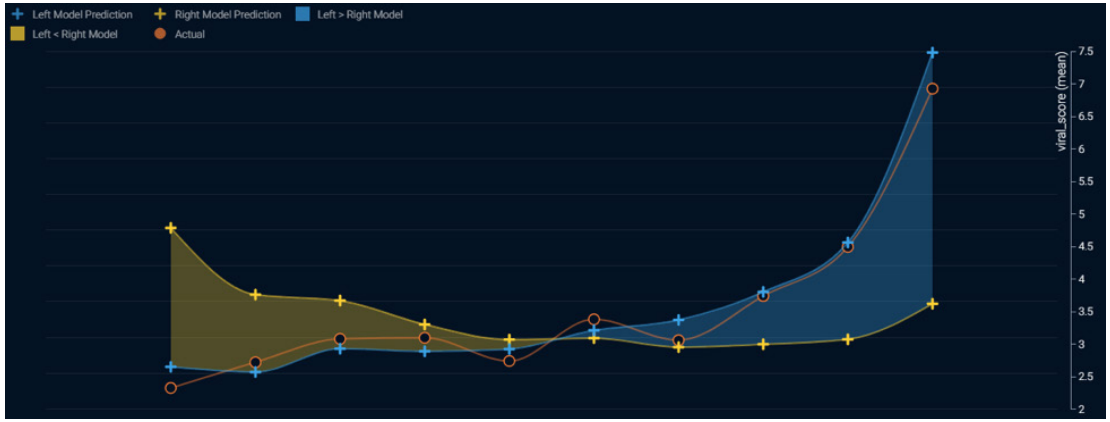
How to Interpret
The Dual-Lift Chart

1. The two lines vary by up to 4.0 (the raw data values range from 1 to 10), indicating that the two models can have significantly divergent predictions.
2. Whenever the models have divergent predictions, the model represented by the blue line is typically closer to the actual results. In fact, the more the two models diverge, the more accurate is the model that is represented by the blue line. This means that the model represented by the blue line is more accurate than the model represented by the yellow line.
3. Advanced: You may be able to get even more accuracy by combining these two models. Using a weighted average of the two models, mostly weighted towards the blue model, may give marginally higher accuracy because when the two lines diverge, the true result lies part-way between the two lines but usually closer to the blue line.

Model Comparison - Dual Lift Charts

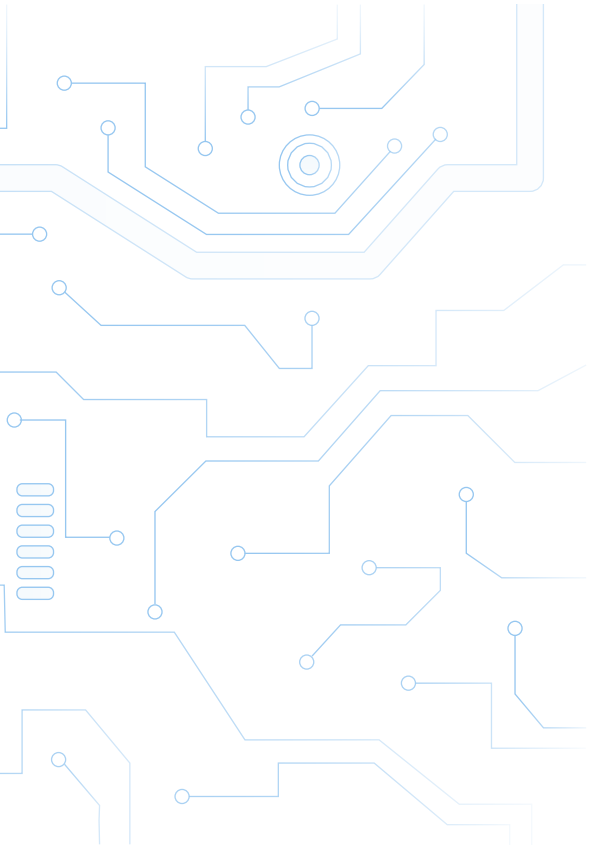
Sometimes, instead of comparing the overall accuracy of two models across all of the data, we want to understand how the predictions are different, and when the models have different predictions, which model is more accurate. Dual lift charts communicate accuracy by showing us how different the model predictions are, and which model is more correct when the predictions diverge. This can be important for certain types of modeling use cases where I want to know how much different models can diverge, in order to assess the risk of winner's curse.

Once again, the blue line represents one model, and the yellow line represents a different model, both selected by the user. This time there is an extra line, an orange line that represents the actual values for the target variable.



To do this we collect pairs of predictions, from the two chosen models, ranking the data rows by how much the first model's prediction exceeds the second model. Then, we plot the predictions and actual results.

On the left side of the plot are the rows where the yellow model has predictions that most exceed those of the blue model. On the right side of the plot are the rows where the blue model has predictions that most exceed those of the yellow model. The more accurate model is the one whose line is closest to the orange line.



Drilling Down into Model Accuracy

With AI, a good overall performance may not be enough. Even if your AI has high accuracy scores, maybe it has a weakness that is important to you. You also want to know when the AI is unsure about what to do. In such situations, you want the AI to triage the decision to a human who can investigate and apply general knowledge and common sense.

"We should be as cautious of AI explanations as we are of each other's—no matter how clever a machine seems. If it can't do better than us at explaining what it's doing … then don't trust it."

MIT Technology Review

Historically, the more complex an algorithm, the more inscrutable it was. The more conservative organizations would only deploy simple algorithms that were easy to understand, but this choice was at the cost of accuracy, which sometimes proved to be very expensive. Other more competitive organizations deployed models into production without understanding their strengths and weaknesses, but this choice was at the cost of unexpected behavior, which sometimes damaged their reputation and brand value.

But the days of choosing between inaccuracy or inscrutability are coming to an end. Automated machine learning makes it possible to quickly and easily discover when the AI can be trusted to make a decision, versus when there is a difficult case that needs a friendly helping hand from humans, no matter how simple or complex the algorithm that has been used.

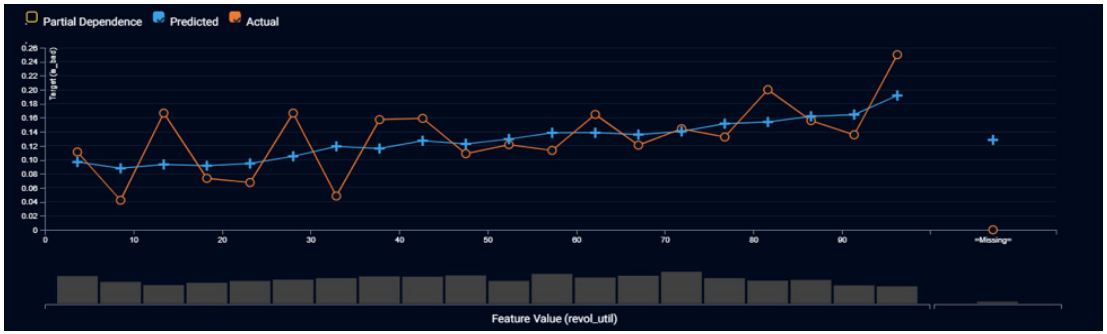
How to Interpret
The Feature Fit Chart

1. The orange and blue lines cross over frequently and the blue line is smoother. So, we have confidence that the model is removing luck, keeping the underlying pattern, and not generally overestimating or underestimating the outcomes.

2. The gap between the orange and blue lines is larger for lower values of line of credit usage, meaning that the model is less sure about outcomes for loan applicants who haven't used as much of their line of credit usage than it is for loan applicants who are close to maxing out their line of credit.

Feature Fit

Even if your overall model accuracy is great, your model may have a blind spot. It may be more accurate for some input values than for others, (i.e., it may be more sure about some decisions than others). Some decisions may be more important to you than others (e.g., decisions about VIP customers). You will want to prevent your AI from accidentally becoming biased, (e.g., treating females and males differently). And if you understand where the model is more uncertain, that may give you ideas about what extra data it needs to make better decisions and improve its accuracy. To achieve these aims, you will want to drill down on the accuracy by input feature value.



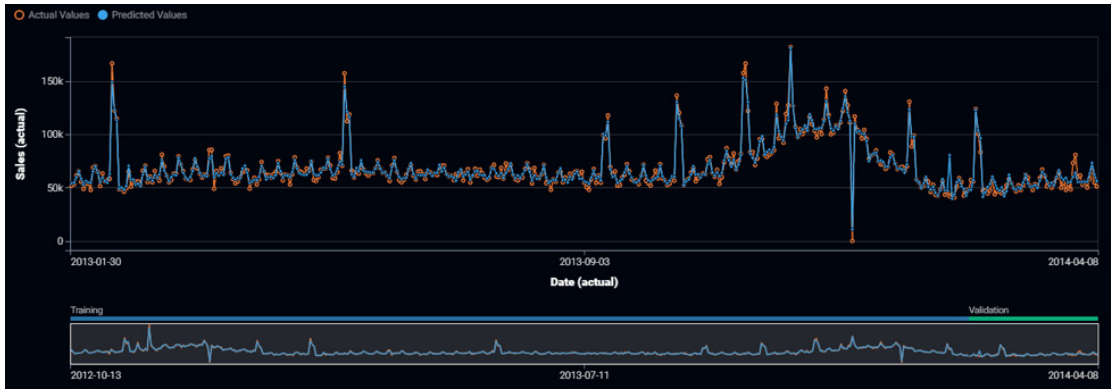
Above is a screenshot of a feature fit for a model that predicts the probability of a personal loan going bad, drilling down by the applicant's percentage usage of their existing line of credit. The orange line is the average proportion of loans that went bad across a range of values for line of credit usage. The blue line is the comparable average predicted probability of a loan going bad. A model is more accurate when the orange and blue lines are closer. We also prefer where the orange and blue lines cross each other frequently, and the blue line is smoother, because that means the model is capturing the underlying patterns (or signal) and ignoring mere luck (or noise).

How to Interpret
The Accuracy Over Time Plot

1. The blue line follows the orange line fairly closely, following the weekly cycle and correctly capturing the highest peaks and troughs in the data. We can be comfortable that the model is correctly capturing seasonal effects and major events (such as the store being closed on Christmas Day 2013).
2. The blue line follows the orange line across the full range of dates. So, we can conclude that the model accuracy is stable across time.
3. The blue line has a spike in early February 2014 that doesn't match the data. We should investigate whether there was a holiday incorrectly specified in our data.
4. The orange line has a spike in late March 2014 that isn't matched by the blue line. We should investigate whether there was a special event or marketing activities that weren't included in our usual data sources.

Accuracy Over Time

Sometimes the process you are modeling changes over time. For example, people's behavior is constantly changing, as is the competitive environment within which you operate. The model may be accurate for a while, but then something changes, and the model is no longer as accurate. Or maybe the model accuracy is seasonal. For example, the model may be more accurate during the winter than the summer. To check the stability and accuracy of the model over time, you want to plot the accuracy across a time period.



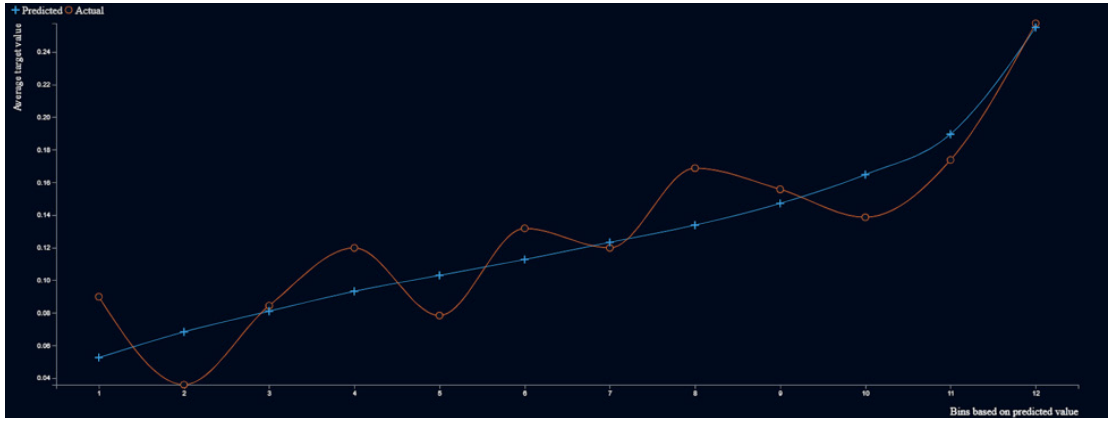
The plot above shows the accuracy over time for daily sales of a retailer. The orange line is the actual sales for each day, while the blue line is the predicted sales. A model is more accurate when the orange and blue lines are closer. We also prefer where the blue lines capture the seasonal effects evident in the data.

How to Interpret
The Lift Chart

1. The orange and blue lines cross over many times, indicating that the model does not consistently overestimate or underestimate.
2. The orange and blue lines typically diverge by 0.03, and the orange line sometimes slopes downward, indicating that even though the model is predictive, it is not highly accurate. This may be because so many unexpected events can occur during the several years in which a personal loan is being paid off.
3. Both the blue and orange lines gradually slope upwards, with many values in the middle height range, and a maximum prediction of 0.27. This indicates that there are few clear yes or no predictions in these circumstances. The model may be useful for ranking the credit quality of a loan applicant, but it is not as strong at predicting which specific loans will go bad.

Lift Chart

For a model to be accurate, it must be good at predicting the highs and the lows, not just the average values. Lift charts communicate accuracy by displaying how well a model can separate high values (e.g. finding those customers most likely to purchase your product) from low values (e.g., finding those customers not suited to a product offering). It also shows you how closely the model matches highs and lows in the data.



Above is a screenshot of the lift chart for a model that predicts the probability of a personal loan going bad. The orange line is the average proportion of loans that went bad, while the blue line is the comparable average predicted probability of a loan going bad. To the left of the plot are the loans that have the lowest predicted probability of going bad. To the right are the loans with the highest probability of going bad.

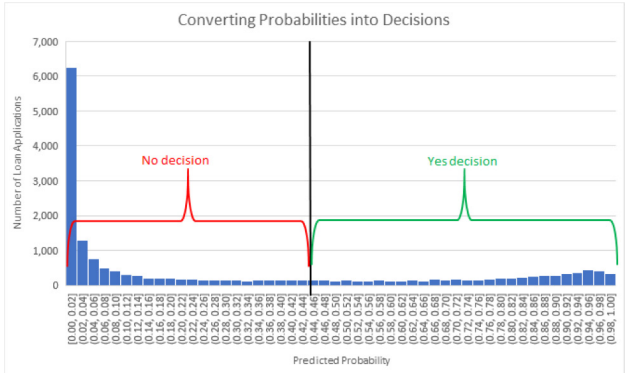
A model is more accurate when the orange and blue lines are close to each other. When considering accuracy across the entire range of outcomes, we also prefer where the orange and blue lines cross each other frequently, because that means that the model isn't consistently overestimating or underestimating. Accurate models show the greatest vertical range in the actual values, the orange line. For yes / no modeling cases (binary classification) only, better models have few values in the middle of the vertical range, because that means that there are relatively few examples where the model is unsure about whether the answer is a yes or a no.

But sometimes you only care about the accuracy across a particular range. For example, if you are predicting the probability that a customer will be interested in purchasing your product, then you may care more about the accuracy for high predicted values than for low predicted values, so you can reject those loan applications. Alternatively, you may care more about the accuracy in deciles 5 to 8 because those are the loans that require extra attention before you can fund them. The point is that accuracy means different things depending upon the business context, so you need to drill down into the accuracy to determine whether the model accuracy meets your needs.

Prediction Distribution Plot

For yes / no use cases a model will output probabilities. Sometimes the probability is exactly what you want (e.g., an insurer wants to set prices using the probability that you will claim and there is no yes or no). In other cases, you want the AI to make a yes or no decision, and in this case you need to turn that predicted probability into a decision. It is up to you to create a business rule that turns each output probability into a yes or a no, and this is typically done by choosing a threshold probability.

Every probability above that threshold becomes a yes decision, and the remainder becomes a no decision. For example, if the probability that a customer will purchase your product is 99% then you will say yes to including them in your marketing campaign, whereas if the probability was 0.1% you choose not to include them in the campaign. Somewhere between these two ranges lies a threshold where the probability is high enough that a no decision turns into a yes decision.



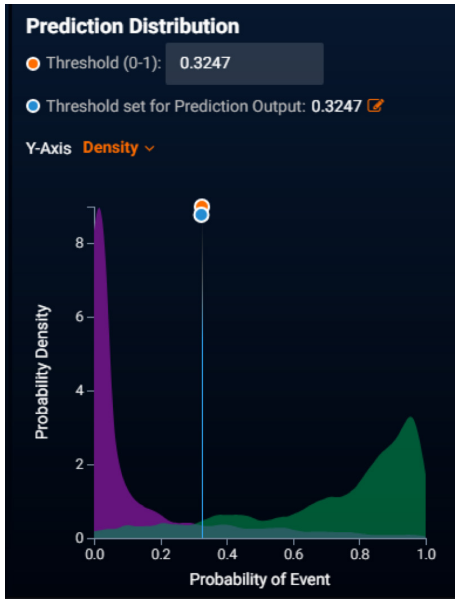
You determine the optimal probability threshold by considering the costs and benefits of each threshold. For example, you may predict the probability that a person has cancer and needs expensive and dangerous surgery:

How to Interpret
The Prediction Distribution Plot

1. Most of the purple and green regions do not overlap, and many of the predictions are close to 0 or 1. This means that the model can usually clearly separate yes decisions from no decisions.
2. For the probability range of 0.2 to 0.6, neither the purple nor green region dominates. Predicted probabilities in this range will not accurately choose yes or no outcomes - the model is reasonable at ranking the credit scoring for these cases but not at deciding a clear yes or no decision of whether the loan will go bad. If you need a clear yes or no result, rather than a probability, then you may wish to triage decisions to humans when the predicted probability lies within this range.

- A false positive occurs when you make a yes decision, but the patient does not have cancer. The cost of a false positive is the cost of a surgery, plus the risks of the surgery, plus the unnecessary stress you caused to the patient.
- A false negative occurs when you make a no decision, but the patient does have cancer. The cost of a false negative is that the patient's cancer continues to grow and spread, causing further health complications, possibly even death.
- A true positive occurs when you correctly make a yes decision. The patient receives the surgery they require. The benefit is the better health and extended life span for the patient, less the cost of surgery.
- A true negative occurs when you correctly make a no decision. The patient avoids unnecessary surgery and unnecessary worry.

You will choose the probability threshold that gives the optimal balance of benefits versus costs and risks. Moving the threshold higher will reduce the number of false positives (yes decisions that should have been no decisions), but at the cost of increasing the number of false negatives (no decisions that should have been yes decisions).



In most cases, some of the data rows will score a mid-range probability where there is a mix of yes and no outcomes. In such cases, we may wish to triage these difficult decisions to a human. Prediction distribution plots enable us to understand the effects of different probability thresholds and see the proportion of decisions that lie in the uncertain probability range.

The screenshot above shows the prediction distribution plot for mortgage defaults. The purple region is a histogram of the prediction probabilities for loans that did not default. The green region is a histogram of the prediction probabilities for loans that did default. The blue line is the currently selected probability threshold for choosing yes or no decisions. An accurate model will not have much overlap between the purple and green regions. The region of decision uncertainty is where the purple and green regions overlap and neither region dominates.

"When the data is complex, large and highly dimensional (even just ten or more columns), users either focus their time on exploring their own hypotheses in a subset of the data, or must manually explore all possible combinations and permutations to ensure a complete and accurate result. This can be very time-consuming. In many cases, therefore, users default to the former approach for expediency; or they may not even know all the possible permutations to explore. As such, they are likely to miss important insights and relationships."

— Augmented Analytics is the Future of Data and Analytics, Gartner

Quickly Find What's Important in Your Data

Have you ever been guilty of confirmation bias, the tendency to interpret new evidence as confirmation of one's existing beliefs or theories? Let's admit it. We have all been guilty of this cognitive bias, even those of us who like to think that we are data-driven.

At Gartner's 2016 "BI Bake-Off" at the Data and Analytics Summit in Dallas, Texas, Gartner gave representatives of several software vendors a set of university and college student demographic data and payroll data. They asked the vendors to derive insights about which university graduates would have the most earning power ten years after graduation. The college scorecard data was complex, with approximately 2,000 variables available to explore manually. With limited time to do the analysis, the data analysts did what expert analysts typically do and explored their own hypotheses first. In doing so, they all drew the same obvious conclusion (attending a leading university leads to higher incomes) but missed the strongest driver—the parental incomes of the students.

How often do business people draw suboptimal conclusions from their data? With dataset sizes getting larger and larger, with more and more input features, it is tempting to create a dashboard that shows the relationships for the key input features that we believe will matter the most. How many times might there be other more important factors that we did not think to explore?

Historically, finding the most important rows and columns was easy because you only had small datasets, with maybe only a dozen columns and a couple of dozen rows. A statistician could manually test the statistical significance of columns, then manually flag rows with results that are outliers, or rows having high leverage. But as datasets get wider and longer, manual identification of important rows and columns becomes too time-consuming and error-prone.

The days of manually finding what is important are ending. Automated machine learning is a game changer, removing the need for manually searching through wide datasets, and providing human-friendly insights without the statistical jargon of p-values and Z-scores.

How to Interpret
The Feature Impact Plot

1. The first five input features have significantly higher values, and therefore have significantly stronger predictive power than the remainder of the input features. These five columns are the most important columns, the ones you are most likely to include in a management dashboard. These are also the five columns which we should prioritize for sensibility checking the patterns that the model found in the data. We will discuss where to find and how to interpret these follow-up insights further on.

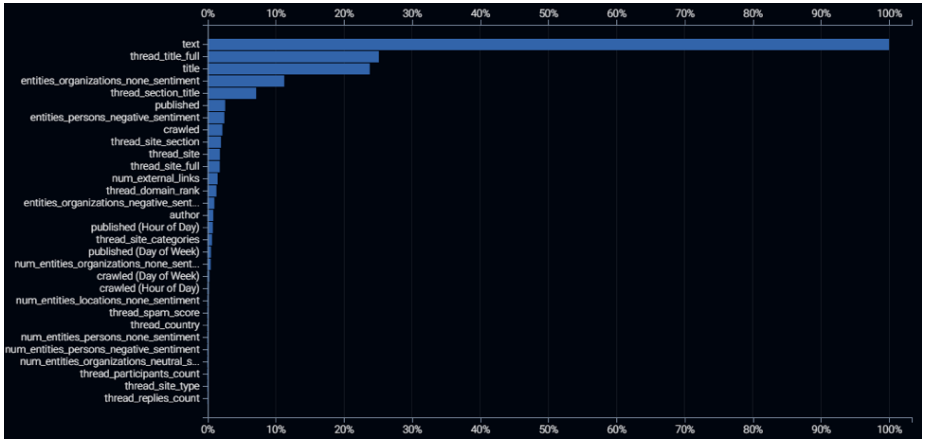
2. The relative importance of the input columns makes sense compared to our expectations. For example, if the hour of the day that the website was crawled was the most important for predicting virality, we would be suspicious of the veracity of the data and/or the model. Similarly, if the title of the news story were determined to be of negligible importance, we would have less trust in the model.

3. The text of the new story is the dominant input feature for predicting virality, easily eclipsing the title of the news story. This indicates the virality is determined more by the detail of what is written than by the first impressions gained by the news headline.

4. All of this data was obtained via an external data subscription, so if we were paying extra for the unimportant columns, such as the site type (ranked second-last in the plot), then we may want to consider lapsing that subscription.

Feature Impact

Modern datasets can have thousands of columns. Manufacturers have thousands of sensors along their production lines. Marketers use hundreds of Google AdWords search terms, plus a huge range of demographic and behavioral features for each customer. Healthcare providers have thousands of possible medical diagnostic results, diagnosis codes, and treatments in their databases. Data analysts are expected to know which columns to include in performance dashboards, whether any key inputs are causing issues with bias, to justify the cost of external data sources, and to apply common sense checks. It is a challenge for those data analysts to avoid being overwhelmed by the tsunami of data and develop valuable insights in a timely manner. To find important columns in a dataset, you will want to see their feature impact.



Above is the feature impact for a model that predicts which news stories will go viral, i.e. which online news stories will be shared the most across social media. The blue bars show the relative importance of each input feature, ranked from the most important (scaled to 100%) to the least important (with importance close to, or equal to 0%). All importance values are relative to the top-ranked input feature. Input feature importance is specific to a model and will vary from model to model because each algorithm uses data in its own unique way. The feature impact is a ranking of how important each input feature is to the predictive power of a trained model. So it is always within the context of which target column value you are trying to predict.

How to Interpret
The Anomaly Detection Table

1. The first few rows' values do not appear to contain errors. So, the anomalies are not incorrect data.
2. Focusing on the first row of data, we see unusually high values for total spending, overpayments, refunds and merchant credits. These values are also high versus the customer's income. This customer's behavior is unusual.
3. The rows with higher ranked anomaly scores are highly likely to require a suspicious activity report (SAR). So, anomalous behavior may be predictive of money laundering. We definitely want to keep such anomalous rows in the training data.
4. Since the anomalous behaviors have predictive power, it may be valuable to add new input features that make it easier to identify these anomalous behaviors, such as taking the ratio of spending versus income, and the ratio of overpayments, refunds and merchant credits versus spending.

Anomaly Detection

Modern datasets can have millions of rows. Each row may be a customer, a product sold, or a payment transaction. Some of these rows of data may be more worthy of your attention than others. Machine learning models, which power most AIs, learn by example from your historical data. Since incorrect data may teach the wrong ideas, you want your AI to be trained by trustworthy data. Weird-looking data is, therefore, worthy of your attention, as it may be incorrect or it may contain unexpected insights and ideas for further feature engineering. To find these rows we can look at the results of anomaly detection. Unlike feature impact (described in the previous section), anomaly detection is not specific to the target column values that you are trying to predict.

anomalyScore	SAR	kycRiskScore	income	tenureMonths	creditScore	state	nbrPurchases90d	avgTxnSize90d	totalSpend90d	csrNotes
1	1	2	160000	15	731	ME	609	145.78	88780.02	
0.9776	1	3	111000	24	694	RI	203	845.44	171624.32	
0.9161	0	3	191100	12	727	PA	209	1141.04	238477.36	
0.8922	1	3	193600	20	747	MA	490	523.25	256392.5	statemen
0.84	1	1	68500	76	724	NY	107	415.07	44412.49	
0.7685	1	1	84600	7	756	NY	446	451.26	201261.96	
0.7288	1	2	135200	7	665	NY	141	817.82	115312.62	passwor
0.7259	0	1	118300	12	690	MA	429	37	15873	billing ad
0.7104	1	2	158900	3	616	NY	256	48.83	12500.48	
0.6981	0	1	82800	22	724	NJ	206	201.46	41500.76	
0.6977	0	1	62500	56	727	MA	47	554.88	26079.36	
0.6975	0	2	78900	15	789	PA	55	382.28	21025.4	change n
0.6866	1	1	82400	33	702	RI	262	209.79	54964.98	

The screenshot above shows the anomaly detection results for a project that predicts whether a banking customer is a money launderer. The anomaly score lies within the range of 0 to 1, with a score of 1 meaning that a data row is very different from the other rows in the dataset, and a score of 0 indicating that there is nothing unusual in that data row. When we find an anomalous row, we look at the values for that row to find unusual values, particular values that appear to be errors. We may also be interested in seeing whether the anomaly scores are predictive of the target column outcomes.

See Patterns the Model Found in Your Data

Imagine you are hiring a new staff member who will decide which personal loan applications a bank will accept and which it will not. So far, the interview is going well. You've asked the applicant which data fields are most important and they've answered income, credit rating and amount of the loan. Next, you ask them which credit rating values aren't acceptable for loan acceptance and they confidently answer that applicants with the very highest credit scores, anything over 800, should be instantly rejected as bad risks. Would you give that person the job, or instead refer them to an introductory guide to credit rating scores?

It's no different for an artificial intelligence (AI). Just as you wouldn't hire a person who uses the wrong data values to make decisions, you should ask your AI which data values lead to high or low predictions, or yes or no decisions.

For some, finding patterns in data is the end-goal. They may want insights into customer behavior, the causes of equipment faults, or key drivers of personal loan risks. But the winners in the Fourth Industrial Revolution are those who have moved beyond insights, using those same models to automate decision making. Successful organizations build AIs that automatically offer the right product to the right customer, repair equipment before it fails, and approve loan applications.

Historically, finding the patterns used by a predictive model was easy because the models were simple. Some models were so simple that even an ordinary person could understand the mathematical formula (e.g., distance = speed x time). These deceptively simple-looking models often took weeks to design, as statisticians had to follow a trial-and-error process to find the best equation for each pattern in the data. But such simple models didn't deliver the accuracy required to remain competitive. So, data scientists developed more complex algorithms. Many of these modern complex algorithms use a formula that is thousands of lines long, so long that even the most mathematically adept person can't read and interpret the entire formula at once.

In the modern AI-driven organization, insights are used to decide whether an AI can be trusted enough to be deployed into production. The patterns the model found in the data are used for common sense checks (e.g., do higher credit scores make a loan application more likely to be accepted?) and providing human-friendly explanations of new models to management.

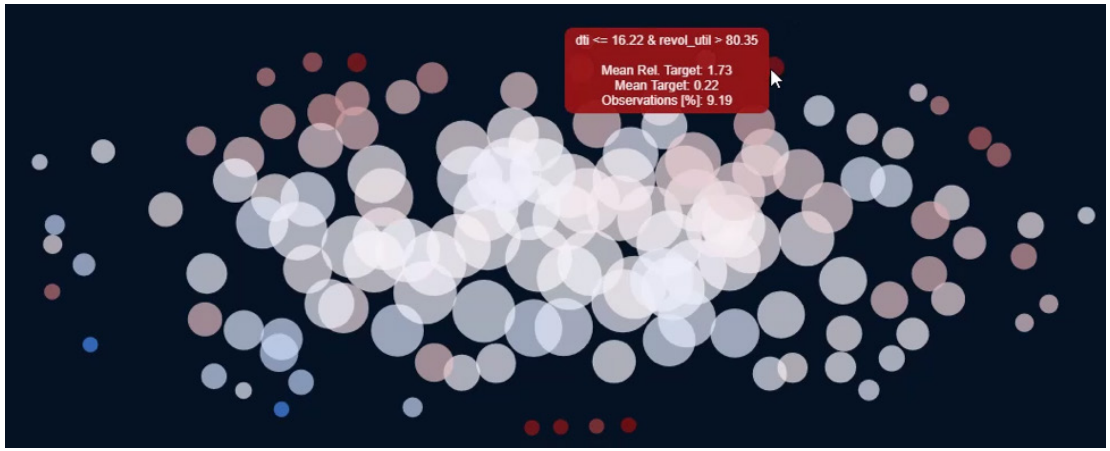
How to Interpret
The Hot Spots

1. There are several dark red circles and a couple of blue circles. This means that it is possible to predict differences in loan outcomes, although this may be easier for loans with poor outcomes than for especially good loan risks.

2. Looking at one of the high-risk hot spots, we see that the heuristic rules are that the debt-to-income ratio is less than or equal to 16.22% and also the applicant's line of credit usage exceeds 80.35% of the limit. This is an interesting interaction effect. Usually, we expect applicants with a low debt-to-income ratio to be good risks, and we expect applicants that max out their line of credit to be bad risks. When seen concurrently, the applicant is a bad risk.

Hot Spots

Data analysts want to quickly determine whether it is possible to predict different outcomes within the data. Before diving into complex models with complex patterns, it can be helpful to start with simple heuristics. These heuristics are simple data filtering rules that quickly find data segments having different outcomes to the average. If simple heuristics are successful, then other more complex models are likely to be successful. Heuristics are also useful for communicating key effects to non-technical colleagues. To see heuristics, you will want to look at hot spots.



Above are the hot spots for Lending Club's loan data, predicting which personal loans will go bad. Each circle is a subset of rows defined by heuristic rules. Dark red circles represent data segments with higher outcomes, (i.e., a higher proportion of bad loans). Dark blue circles represent data segments with lower outcomes, (i.e., a lower proportion of bad loans). White and pale circles have outcomes that are not much different from average outcomes. Note that a data row can appear in more than one circle. If the hotspots include both dark red and dark blue circles, then you know that it is possible to segment the data in such a way as to predict varying outcomes. Moving the mouse over a hotspot shows the heuristic rules, the average outcome and the proportion of data rows lying within the hot spot.

How to Interpret
The Feature Effects

1. The probability of a loan going bad generally reduces as the applicant's income increases. This makes sense. We would not trust the model if it said that higher incomes were poorer risks. The risk stops improving once incomes exceed $100,000 per annum (pa). This also makes sense, as once an applicant has more than enough money to repay a loan, any extra income won't make much of a difference.

2. While the pattern generally slopes downward from left to right, there are income ranges where it temporarily slopes upward. As we don't expect this from our general knowledge, we may conclude, after investigation, that it is due to random luck in the data. If these small bumps in the general pattern are too problematic for your organization's business processes, you would conclude that this model is not suitable, and look for a more suitable model in the DataRobot leaderboard. Another option, for advanced users, is to apply DataRobot's monotonicity constraint settings to force the model to never slope upward.

3. There is a sudden drop in risk when income is $42,000 pa that may be an artifact of Lending Club's loan acceptance criteria, with different acceptance criteria for applicants below and above this income threshold. This insight warrants further investigation because if it proves to be a replicable effect, it could be used to optimize your lending strategy.

Feature Effects

Thanks to feature impact, you already know which input feature columns are the most important. The next step is to discover what your model is doing with each of these important input features, the patterns the model found that are driving its predictions and decisions. The patterns the model found in the data may help you prevent unwanted bias and discrimination, plus they may suggest further investigations and/or data preparation steps. To understand how a model is using non-textual features, you will want to look at the feature effects.

The plot above shows the feature effects for a model that predicts the probability of a personal loan going bad, drilling down by the applicants' annual income. The yellow line shows how the prediction changes on a typical data row as you change the value of income. There are no absolute rules for what is right or wrong. You need to apply your human understanding of the business rules, general knowledge and common sense to decide whether the pattern is suitable. In particular, you will typically pay attention to the slope (which input values result in higher or lower predictions) and the smoothness of the pattern.

How to Interpret
The Word Cloud

1. Injury descriptions containing phrases such as “lower back,” “fractured” or “knee” are likely to cost more to settle. This makes sense, because back and joint injuries can be debilitating and difficult to treat.

2. Injury descriptions containing the phrases “eye,” “left eye” or “right eye” are likely to not cost much. This is not what one would initially expect, but detailed investigations of the data show that the eye injuries are caused by workers getting dust in their eyes when using a grinder, and this is easily treated by flushing the eye.

3. The words “laceration” and “lacerated” appear frequently with similar effects. You may wish to apply stemming in your data pre-processing to change these words to always be the same tense.

Word Clouds

While feature effects tell you about patterns a model used for non-text input features, word clouds tell you about patterns a model used for free-text input features. Word clouds show the words and phrases that drive predictions.

Above is a word cloud of the injury description for an insurer’s workers’ compensation claims, predicting the total payout for each claim. Dark red words and phrases are associated with higher predictions, while dark blue words and phrases are associated with lower predictions. The size of the font indicates how frequently a word or phrase appears in the data. Since a text field contains multiple words and phrases, the prediction is determined by the balance of red and blue words and phrases in the data field. When reviewing a word cloud, you will apply common sense checks, and look for words that should be merged together in data pre-processing.

When You Absolutely Must Have a Formula

An AP-AOL news poll found that mathematics was the subject that 37% of Americans hated most during their school days, easily ranking as the most hated school subject. People feel anxiety when confronted with mathematics. A search on Amazon will return dozens of self-help books dedicated to overcoming "math anxiety."

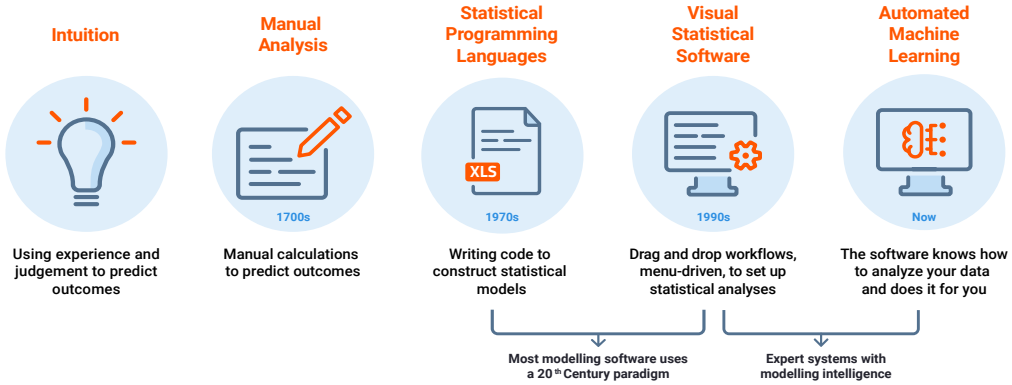
So, why is it that some people think that the answer to understanding a machine learning model is to see a mathematical formula? Would you ask a human to explain their decision-making process by showing an MRI scan of their brain?

Here at DataRobot, we recommend that you seek to interpret a model by asking it the same questions you would of a human:

1. Which data inputs are the most important for making decisions?
2. What data values tend to result in high or low predictions? In yes or no decisions? What patterns did you derive from the data?
3. What characteristics of an individual datapoint caused an algorithmic decision about it to have a different outcome to a typical result?

Sometimes you just can't avoid creating a formula. In certain regulatory domains, the regulator requires you to send them the complete formula that you used. For example, in South Korea, the banking regulator requires banks to submit the complete formula that they use for each algorithmic decision. Other times your legacy IT infrastructure doesn't support modern architectures such as REST APIs or your AI runs in a disconnected environment, such as on an iPad disconnected from the internet, so you need the formula to embed within your computer code. Finally, if you are conducting research or publishing an academic paper, you may need to disclose the formula for the trained model in your research results.

A Quick History of Predictions and Models



Historically, predictions and data-driven decisions weren't so complex. Up until the 1700s, there was no theoretical basis for making data-based decisions, and consequently most decisions were based upon intuition and tradition. This changed with the development of statistical theory. Statistical methods were manually applied to scientific experiments, small in scale, with strict constraints upon the experimental design. It wasn't until the 1970s that the first statistical programming languages were released, carrying out what had previously been manual calculations. Even when visual (drag and drop) statistical software was developed in the 1990s, the underlying analytic methodology remained unchanged. Statistical methods test a hypothesis expressed as a formula. Humans interpreted the results by looking at the formula. Models were deployed by applying the formula. This worked because the formulae were simple and rarely changed.

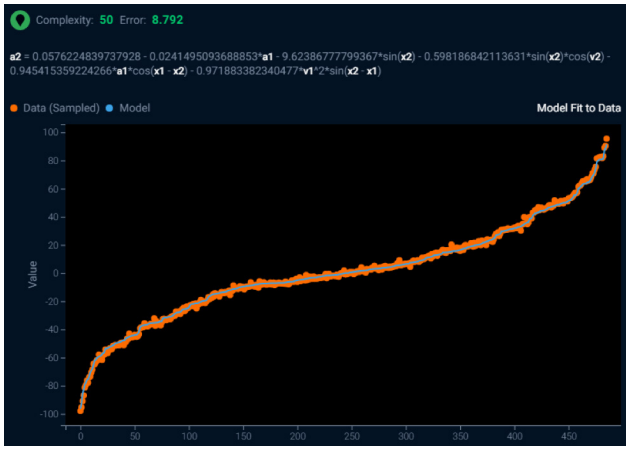
In the modern AI-driven organization, things have changed. Data-driven decisions are a competitive advantage, and data isn't always collected in a manner that is consistent with the strict theoretical demands of statistical methods. Data volumes are higher, enabling organizations to build more complex models, and competition is fiercer, requiring more accurate models. The world is changing faster than ever, requiring models to be trained faster and refreshed more often. Simple statistical formulae are not sufficient for modern AI-driven organizations.

How to Interpret
The Eureqa Formula

1. You will need a mathematical background in geometry and trigonometry to understand the sine and cosine functions used in the generated formula.
2. The complex motion of the double pendulum can be expressed as an equation containing just six coefficients and six trigonometrical transformations.

Eureqa Formula

If your ultimate goal is to find a formula, then one of the best ways to achieve that goal is by using an algorithm that is specially designed to create a formula. The Eureqa blueprint leverages a proprietary evolutionary algorithm to automatically generate complex models as analytical expressions. Rather than taking an existing model structure and fitting the data to the model, Eureqa uses combinations of non-linear transformations to build highly accurate models from your data that can still be easily interpreted as simple mathematical equations. Eureqa models optimize parsimony, distilling the underlying patterns in your data with the least amount of complexity possible, even able to build the equation of motion from the observations of a double pendulum.



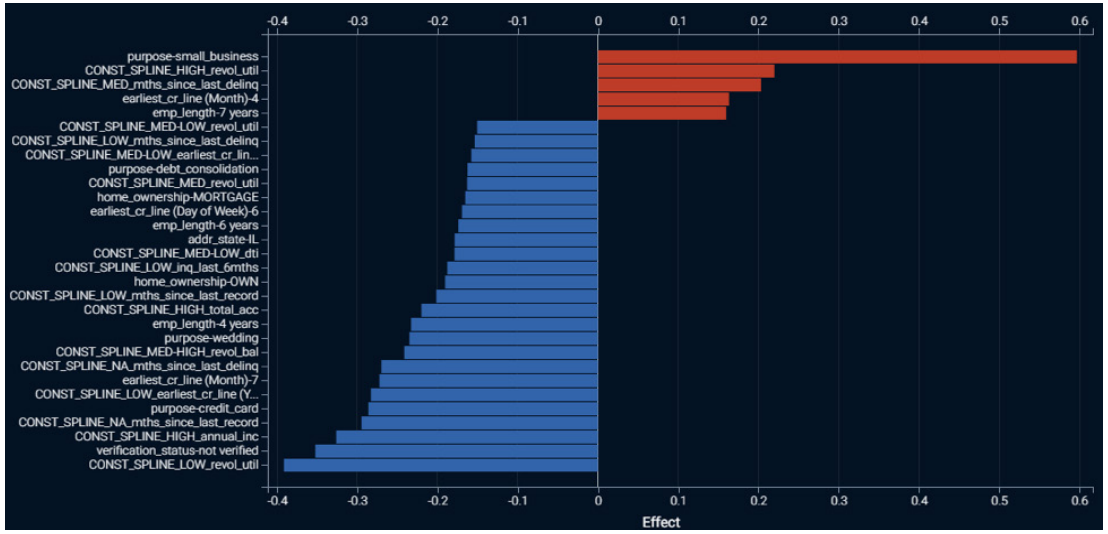
Above is a screenshot of the Eureqa output for a model fitted to observations of double pendulum movement. A double pendulum consists of two pendula attached end to end. In physics and mathematics, in the area of dynamical systems, a double pendulum is a pendulum with another pendulum attached to its end and is a simple physical system that exhibits rich dynamic behavior with a strong sensitivity to initial conditions.

In the Eureqa algorithm output, the higher the complexity score, the more complex the generated formula. You can find a simpler generated formula by selecting a lower complexity score from the training history. The lower the error score, the closer the formula matches the observations, and the more accurate the formula. Immediately beneath the complexity and error score is the generated formula.

a2 = 0.0576224839737928 - 0.0241495093688853*a1 - 9.62386777799367*sin(x2) - 0.598186842113631*sin(x2)*cos(v2) - 0.945415359224266*a1*cos(x1 - x2) - 0.971883382340477*v1^2*sin(x2 - x1)

Model Coefficients

Sometimes a simple formula is enough for your needs. Many statistical models are expressed as a simple linear formula, multiplying each input variable by a weight, then summing to get the prediction. Linear models can also be found in machine learning, however machine learning takes a slightly different approach, and its linear models are regularized (i.e., there is a penalty for the complexity of the formula). Linear models are often easy for people to understand, but that simplicity comes at the cost of poorer accuracy. This is particularly true when the input variables interact with each other (e.g., the effect of snow upon traffic speeds is different depending upon the geographic location and climate). Linear models are rarely as accurate as modern machine learning algorithms.



The screenshot above shows a bar chart of the strongest effects found within a regularized GLM trained on Lending Club's loan data, predicting which personal loans will go bad. The red bars indicate positive effects, and the longer the red bar the higher the prediction. Similarly, the blue bars indicate negative effects, and the longer the blue bar the lower the prediction. The labels on the vertical axis are the feature names and the value that each feature takes. The horizontal axis is the linear effect, the coefficient or weight given for the presence of

How to Interpret
The Coefficient Table

1. Since the link function is "logit," this is a classification model that predicts probabilities.

2. You will need a mathematical background in generalized linear models to understand the link function and construct a formula. The inverse logit function is

$$\text{logit}^{-1}(\alpha) = \text{logistic}(\alpha) = \frac{1}{1+\exp(-\alpha)} = \frac{\exp(\alpha)}{\exp(\alpha)+1}$$

3. The alpha value used in the link function is calculated by summing the intercept, plus each coefficient multiplied by its corresponding feature value, giving a long formula:

$$\begin{aligned}\alpha \quad &= -0.366732002683 + \\ &0.110129 \times \text{if}(\text{addr\_state}=\text{'AZ'},1,0) + \\ &0.040184 \times \text{if}(\text{addr\_state}=\text{'CA'},1,0) + \\ &\ldots + \\ &0.085662 \times \text{if}(-\infty < \text{annual\_inc} < 37383.0002894,1,0) + \\ &-0.025699 \times \text{if}(37383.002894 \leq \text{annual\_inc} < 50000.0002656,1,0) + \\ &\ldots + \\ &-0.040994 \times \text{if}(\text{zipcode}=\text{'112xx'},1,0)\end{aligned}$$

4. You will need a mathematical background to understand and explain this formula.

the feature value. Where you see the label "CONST_SPLINE", that indicates that a numeric feature was split into bins or ranges. You can find the definitions for each range by downloading a table containing all of the model coefficients.

To create a formula, look for a blueprint with a βi badge, then go to the coefficients tab and download the table containing all of the coefficients. Here is a snippet from the table corresponding to the bar chart shown on the previous page.

Intercept: -0.366732002683
Link function: logit

Feature Name	Type	Derived Feature	Transform1	Value1	Coefficient
addr_state	CAT	addr_state-AZ	One-hot	'AZ'	0.110129
addr_state	CAT	addr_state-CA	One-hot	'CA'	0.040184
addr_state	CAT	addr_state-CO	One-hot	'CO'	-0.056726
addr_state	CAT	addr_state-CT	One-hot	'CT'	0.029624
addr_state	CAT	addr_state-FL	One-hot	'FL'	-0.003161
addr_state	CAT	addr_state-GA	One-hot	'GA'	-0.008156
addr_state	CAT	addr_state-IL	One-hot	'IL'	-0.178555
addr_state	CAT	addr_state-MA	One-hot	'MA'	0.011998
addr_state	CAT	addr_state-MD	One-hot	'MD'	-0.045806
addr_state	CAT	addr_state-MI	One-hot	'MI'	-0.027885
addr_state	CAT	addr_state-MO	One-hot	'MO'	0.022557
addr_state	CAT	addr_state-NC	One-hot	'NC'	0.141152
addr_state	CAT	addr_state-NJ	One-hot	'NJ'	-0.008222
addr_state	CAT	addr_state-NY	One-hot	'NY'	-0.127358
addr_state	CAT	addr_state-OH	One-hot	'OH'	-0.009102
addr_state	CAT	addr_state-OR	One-hot	'OR'	0.006639
addr_state	CAT	addr_state-PA	One-hot	'PA'	-0.149844
addr_state	CAT	addr_state-TX	One-hot	'TX'	-0.012326
addr_state	CAT	addr_state-VA	One-hot	'VA'	-0.016546
addr_state	CAT	addr_state-WA	One-hot	'WA'	-0.064122
addr_state	CAT	addr_state-small_count	One-hot	Other categories	-0.021208
annual_inc	NUM	CONST_SPLINE_LOW_annual_inc	Constant splines	(-inf, 37383.0002894)	0.085662
annual_inc	NUM	CONST_SPLINE_MED-LOW_annual_inc	Constant splines	[37383.0002894, 50000.0002656)	-0.025699
annual_inc	NUM	CONST_SPLINE_MED_annual_inc	Constant splines	[50000.0002656, 65000.0001608) (default for NA)	-0.008771
annual_inc	NUM	CONST_SPLINE_MED-HIGH_annual_inc	Constant splines	[65000.0001608, 90000.0003132)	-0.091770
annual_inc	NUM	CONST_SPLINE_HIGH_annual_inc	Constant splines	[90000.0003132, inf)	-0.326154
collections_12_mths_ex_med	NUM	CONST_SPLINE_HIGH_collections_12_mths_ex_med	Constant splines	[1.25136e-08, inf)	0.000000
collections_12_mths_ex_med	NUM	CONST_SPLINE_NA_collections_12_mths_ex_med	Constant splines	Missing value	-0.069059
delinq_2yrs	NUM	CONST_SPLINE_HIGH_delinq_2yrs	Constant splines	[1.25136e-08, inf)	0.115421

The table tells us the link function, each coefficient, and if the variable is numeric, then the range of values the coefficient applies to. One can construct a formula by adding all of the coefficients times the feature values, then applying the inverse of the link function. If the link function is "log", which is the case for regression models, then each coefficient can be translated into a percentage adjustment to the prediction. If the link function is a "logit", which is the case for classification models, the coefficients cannot be translated into percentage adjustments to the prediction.

As you can see, just because a formula has a simple structure, it doesn't mean that it is always easy for people to understand completely. It is recommended that if you have to explain a linear formula to a non-technical business person, that you focus on discussing which effects are strongest and which increase or decrease the prediction.

How to Interpret
The Rating Table

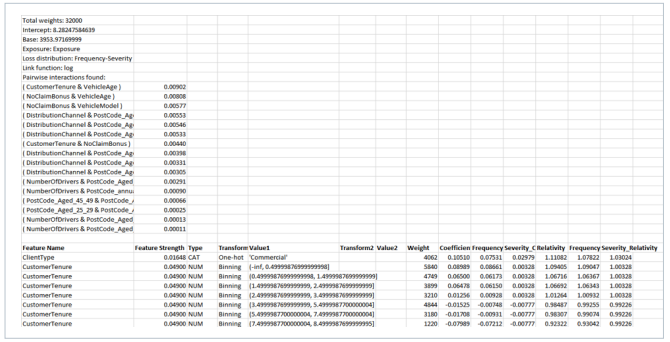
1. Since the link function is “log”, this is a regression model that predicts probabilities.
2. You will need a mathematical background in generalized linear models to understand the link function and construct a formula. The inverse log is the natural exponential.
3. The value used in the link function is calculated by summing the base, plus each coefficient multiplied by its corresponding feature value, giving a long formula:

   x = 3953.97 +
   0.10510×if(ClientType='Commercial',1,0) +
   0.08989×if(-∞<CustomerTenure<0.49999,1,0) +
   ...+
   ...+
   0.00568×if(DistributionChannel='112xx' AND PostCode_Aged_40_44=NA,1,0)

4. You will need a mathematical background to understand and explain this formula.

Generalized Additive2 Model - Rating Tables

As noted above, linear models sacrifice accuracy for a simpler structure, primarily because they ignore interaction effects. But sometimes you want both accuracy and a linear structure. This can be the case in insurance and banking, where the regulator is expecting model submissions to be in a linear format. When you want accuracy plus transparency, Generalized Additive2 Models (GA2Ms) are an attractive option. All of the coefficients, including for text and interaction effects, are available for download as a “rating table”.



Total weights: 32000
Intercept: 8.2824758463
Base: 3953.97169999
Exposure: Exposure
Loss distribution: Frequency-Severity
Link function: log
Pairwise interactions found:

(CustomerTenure & VehicleAge)	0.00902
(NoClaimBonus & VehicleAge)	0.00808
(NoClaimBonus & VehicleModel)	0.00577
(DistributionChannel & PostCode_Ag	0.00553
(DistributionChannel & PostCode_Ag	0.00546
(DistributionChannel & PostCode_Ag	0.00533
(CustomerTenure & NoClaimBonus)	0.00440
(DistributionChannel & PostCode_Ag	0.00398
(DistributionChannel & PostCode_Ag	0.00331
(DistributionChannel & PostCode_Ag	0.00305
(NumberOfDrivers & PostCode_Aged	0.00291
(NumberOfDrivers & PostCode_annu	0.00090
(PostCode_Aged_45_49 & PostCode_	0.00066
(PostCode_Aged_25_29 & PostCode_	0.00025
(NumberOfDrivers & PostCode_Aged	0.00013
(NumberOfDrivers & PostCode_Aged	0.00011

Feature Name	Feature Strength	Type	Transform	Value1	Transform2	Value2	Weight	Coefficien	Frequency	Severity_C	Relativity	Frequency	Severity_Relativity
ClientType	0.01648	CAT	One-hot	'Commercial'			4062	0.10510	0.07531	0.02979	1.11082	1.07822	1.03024
CustomerTenure	0.04900	NUM	Binning	(-inf, 0.4999987699999998]			5840	0.08989	0.08661	0.00328	1.09405	1.09047	1.00328
CustomerTenure	0.04900	NUM	Binning	(0.4999987699999998, 1.4999987699999999]			4749	0.06500	0.06173	0.00328	1.06716	1.06367	1.00328
CustomerTenure	0.04900	NUM	Binning	(1.4999987699999999, 2.4999987699999999]			3899	0.04478	0.04150	0.00328	1.04592	1.04343	1.00328
CustomerTenure	0.04900	NUM	Binning	(2.4999987699999999, 3.4999987699999999]			3210	0.01256	0.00928	0.00328	1.01264	1.00932	1.00328
CustomerTenure	0.04900	NUM	Binning	(3.4999987699999999, 5.4999987700000004]			4844	-0.01525	-0.00748	-0.00777	0.98487	0.99255	0.99226
CustomerTenure	0.04900	NUM	Binning	(5.4999987700000004, 7.4999987700000004]			3180	-0.01708	-0.00931	-0.00777	0.98307	0.99074	0.99226
CustomerTenure	0.04900	NUM	Binning	(7.4999987700000004, 8.4999987699999995]			1220	-0.07989	-0.07212	-0.00777	0.92322	0.93042	0.99226

The chart on the right is an excerpt from the rating table of a GA2M trained on insurance data to predict claims costs per insurance policy. Just as with model coefficients tables for regularized GLMs, the rating table tells us the link function, each coefficient, and if the variable is numeric then the range of values the coefficient applies to. Unlike regularized GLMs, it also shows interaction effects, the coefficients for combinations of two different features, listing them from the strongest effect to the weakest. One can construct a formula by adding all of the coefficients times the feature values, then applying the inverse of the link function. If the link function is “log”, which is the case for regression models, then each coefficient can be translated into a percentage adjustment to the prediction. If the link function is a “logit”, which is the case for classification models, the coefficients cannot be translated into percentage adjustments to the prediction. For regression models, the rating table will include multiplicative relativities that correspond to the coefficients.

Just because a formula has a simple structure, it doesn’t mean that it is always easy for people to understand completely. It is recommended that if you have to explain a rating table formula to a non-technical business person, that you focus on discussing which effects are strongest, and which increase or decrease the prediction. You may also choose to plot the interaction effects as a heat map or a 3D plot. Examples of such plots can be found here. For non-interaction effects, DataRobot automatically provides feature effects plots.

Understanding How a DataRobot Model Was Made

Just as we care about the ingredients in our food, some people want to know the ingredients in the machine learning algorithm that powers their artificial intelligence (AI). They want to know how the data was prepared to suit the algorithm, any feature engineering applied to the data, and whether any post-processing was applied to the algorithm's results. Model blueprints are the core of DataRobot's technology, encapsulating data processing, feature engineering, and model tuning.

Data processing and feature engineering are often overlooked when building machine learning models, even though they are essential to building a great model and are much more complicated to master. Research shows that "selecting the best model and tuning it leads to approximately a 20% increase in accuracy, up to more than a 60% improvement for certain datasets". DataRobot's model blueprints are data science recipes, combining best-practice data science processes as the ingredients, used and tested by the world best data scientists, packaged ready to produce high-quality machine learning algorithms. And this production-line quality and accuracy directly impacts the bottom line—one organization that switched to DataRobot's model blueprints reported saving hundreds of millions of dollars per annum via improved model accuracy.

Historically, data scientists manually created scripts that trained and ran the machine learning algorithms that powered AI. Each script was craftsman made, a work of art, and each one unique. Scripts can be written in many different languages (e.g., Python, Java, Julia or R), but one thing that all scripts have in common is that they are not suitable for a normal business person to understand. Over the past decade, many standard machine learning libraries have been released by the open source community, removing the need to script every detail, but scripts that use these libraries remain too complex for a normal business person to comprehend. Scripted solutions are little better than black box solutions to anyone who is not an experienced data scientist.

In the modern AI-driven organization, there are dozens, if not hundreds, of machine learning algorithms deployed throughout the organization, too many for each and every one to be built manually using complex scripting. Much like modern organizations manage their software, the modern AI-driven organization wants standardization of AI workflows, repeatability, reduced human error, reduced key-man risk, and human-friendly and regulator-friendly documentation of each and every AI.

Blueprint Diagrams

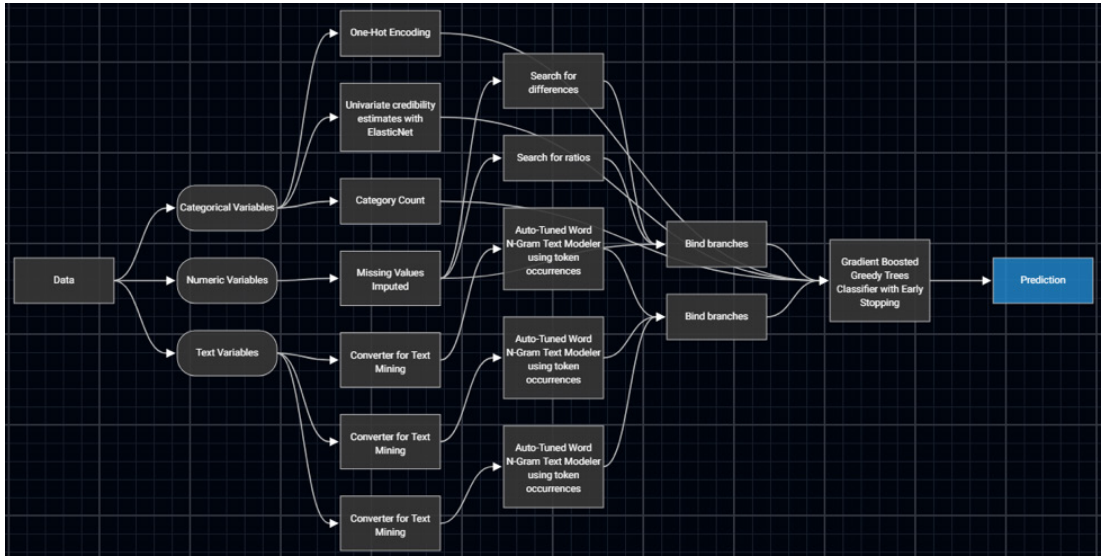
But sometimes there is a need to see the inside of the model to see how it prepares the data, newly generated features, and any post-processing it does. For example, one blueprint may improve accuracy by applying credibility weighting, while another may use automated feature engineering to improve accuracy by adding cluster analysis. Sometimes there is a need to find the source of an algorithm, the academic papers behind its methodology, or the open source library from which it was sourced. Maybe the regulator wants to know the details. Maybe your boss, the Analytics Director, wants to know how this model is different from another model you fitted to this data. Maybe one of your fellow data scientists wants to review your choice of model or wants ideas about what may work for their project. Or maybe one of your business colleagues wants to know whether the text features were used after the other features had first been applied.

Above is a screenshot of a more complex model blueprint for a Gradient Boosted Greedy Trees algorithm fitted

Sometimes there is a need to find the source of an algorithm, the academic papers behind its methodology, or the open source library from which it was sourced.

How to Interpret
The Blueprint Diagram

1. The features are split by data type into categorical, numeric and text.
2. For categorical features:
   - the data is prepared by applying one-hot encoding, and
   - new features are generated by counting the occurrence.
3. For numeric features:
   - the data is prepared by doing missing value imputation, and
   - new features are generated by subtracting one numeric field from another, or by dividing one numeric field by another.
4. For text features the raw text data is turned into numeric data, suitable for this machine learning algorithm, by first running text mining algorithms on each of the three text features in this data.
5. The main algorithm being used is a Gradient Boosted Greedy Trees Classifier with Early Stopping. Clicking on the box to get the documentation shows us a description of what this algorithm does and tells us that this algorithm was sourced from the scikit-learn library in Python.
6. There is no post-processing after the main algorithm. Text mining is used to create primary features to fit against the target column. No scaling of predictions is required.



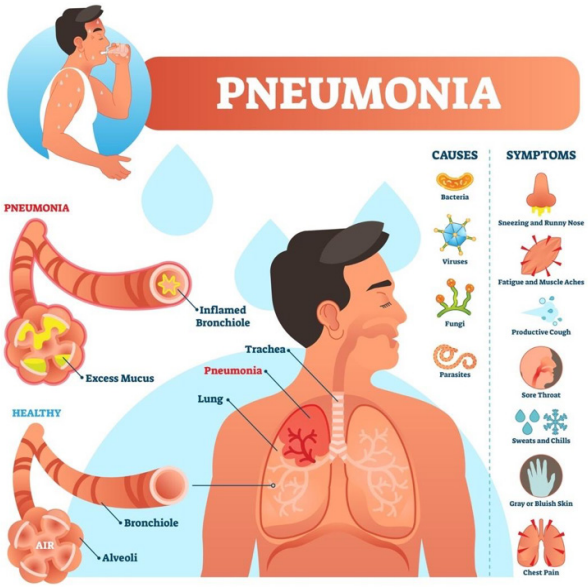
to Lending Club's loan data, used to predict which loans will go bad. Blueprint diagrams always start with a Data box and end with a Prediction box. After the Data box, each feature is split by its data type, so that the most appropriate pre-processing can be applied to prepare it for the algorithm. The next rounds of boxes are for data processing and feature engineering. Then there is a machine learning algorithm taking this data to learn from or to calculate new predictions. Sometimes there is an extra step after the machine learning algorithm, where text mining is trained on the residual errors from the main algorithm, or sometimes there is a prediction scaling step after the main algorithm. You can find a quick explanation of this particular blueprint in our blog about automated feature engineering.

To get the documentation for any step, simply click on the box for that step. This will open documentation that explains what that box does within the pipeline, and often provides links to published research and/or the open sources libraries it uses.

Understanding Why a Prediction Has Its Value

A recently published research paper from Columbia University described a common dilemma in machine learning. Back in the mid-1990s, one cost-effective healthcare initiative investigated the application of machine learning to predict the probability of death for patients with pneumonia so that high-risk patients could be admitted to the hospital while low-risk patients were treated as outpatients. This had the potential to simultaneously improve health outcomes while reducing hospitalization costs. But the most accurate machine learning algorithms were not intelligible. With people's lives on the line, it was decided that the risk of black-box models was not acceptable, and they selected a less accurate rule-based algorithm that was more intelligible.

This choice of algorithm was vindicated when they discovered something strange. The rule-based algorithm learned the rule that patients with pneumonia who have a history of asthma have a lower risk of dying from pneumonia than the general population. This was counterintuitive, with studies showing asthmatics having a high mortality risk from pneumonia. Further investigation showed that upon admission to hospital, asthmatics suffering from pneumonia were often sent directly to the Intensive Care Unit and given the extra effective care that they needed, with consequently improved outcomes. If they had used the more accurate but unintelligible algorithms, they may have inadvertently put asthmatics lives at risk by assessing them as low risk and sending them home.



While your organization may not make life-and-death decisions, the consequences of using unintelligible black-box models could still be serious. Think of the reputation damage from applying unfair recruitment algorithms, thousands left without electricity when a power grid fails, the losses from lending money to bad risks, reduced sales from misselling products to customers, or unclean water discharged into a river after a miscalculation in a water treatment plant.

Historically, statisticians built models designed to test a single hypothesis. The conclusion was either that the hypothesis was correct or that the hypothesis was incorrect. This led to simple and clear explanations (e.g., the patient's health improved because they were treated by a new drug). However, statistical models were more difficult to interpret in multivariate analysis when models simultaneously considered the effects of multiple input feature. It became increasingly difficult to apportion a result to multiple inputs. With the introduction of machine learning, interpretability became more difficult, as the formulae became more complex, capturing complex effects and interactions between inputs. This led many to despair that machine learning was doomed to be black-box technology and that the only choice was between accuracy versus interpretability.

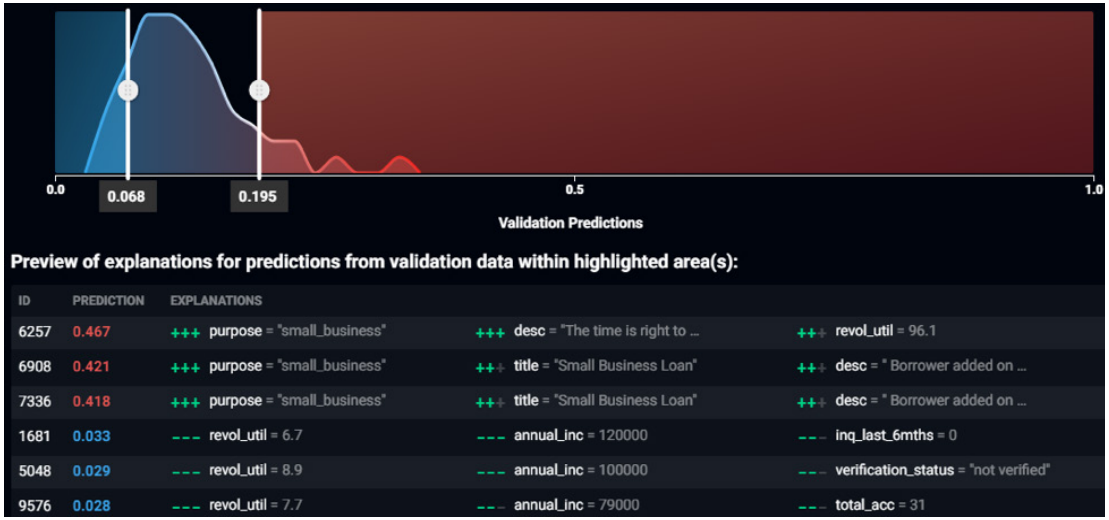
The modern AI-driven organization does not choose between accuracy and interpretability of predictions. It wants and needs both accuracy and interpretability. Its business staff wants actionable insights (e.g., why a specific customer is likely to churn). With the trust and buy-in of those business staff, AIs would not have been accepted within the broader business. Increasing consumer activism and regulatory restrictions mean that organizations must explain algorithmic decisions as they affect their customers. The modern AI-driven organization achieves this by using the latest generation of AI, which explains its decisions in a human-friendly manner, accessible to data scientists and business people alike.

Prediction Explanations

Data analysts want insights into individual customers. Marketers want to align campaign messaging with the reasons why customers were selected for the campaign by an algorithm. Business staff want to see detailed worked examples to check against business rules and common sense. For these purposes and more, DataRobot has prediction explanations. A prediction explanation tells us which input features caused a prediction to have its value. Prediction explanations are best at explaining why a datapoint has a prediction that is different to a typical prediction by considering two aspects of that datapoint:

1. **What is different about this datapoint versus a typical datapoint?**

2. **How does that difference in data values change the prediction versus a typical prediction?**



Above is a screenshot of the prediction explanations for a complex algorithm trained on Lending Club's loan data, used to predict which loans will go bad.

The plot above the table shows the distribution of predictions for the dataset. Blue indicates low predictions and red indicates high predictions. The two sliding ranges define the prediction ranges for which prediction explanations will be calculated. Since prediction explanations are computationally intensive and designed to explain why a prediction is higher or lower than average, the sliding ranges default to only show explanations for very high and very low predictions. You can change which explanations are calculated by moving the sliders - to obtain explanations for all datapoints, just move the sliders so that they touch.

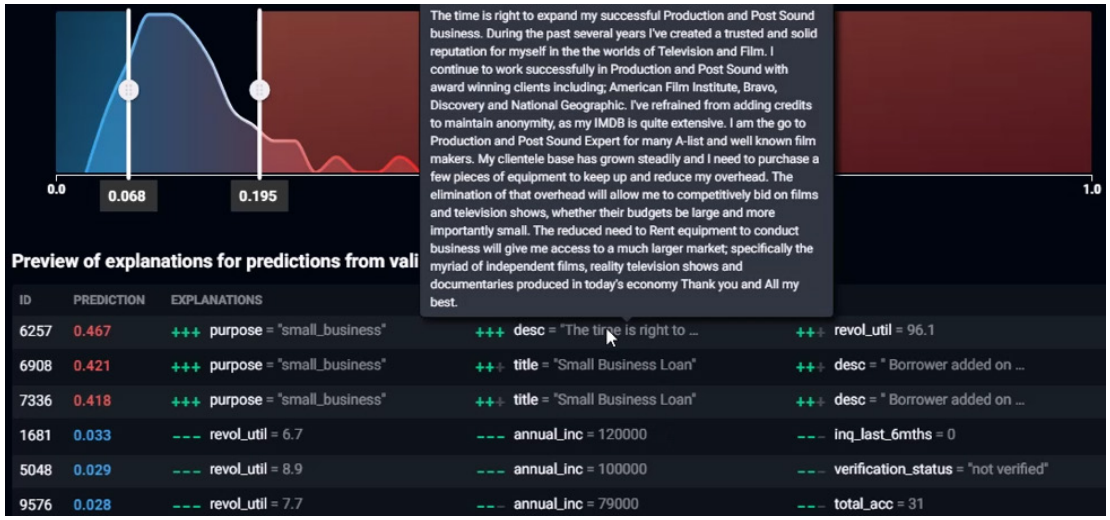
The table below the plot gives six examples of prediction explanations, showing three of the highest predictions and three of the lowest predictions. If you want explanations for more rows, don't despair, as prediction explanations are available for each and every row in the training data. Most important, prediction explanations are available for new data. ID is the row number in the training dataset. Prediction is the predicted value, a probability for classification projects, an amount for regression projects. Following the prediction are the values of the three input features that have the greatest effect upon the prediction and their relative strengths. The input

How to Interpret
The Prediction Explanations

1. The top row, with ID 6257, has a high probability of the loan going bad because the loan is for a small business, the borrower wants to use the money to film documentaries, and because the borrower's line of credit facility has already reached 96.1% of its limit. All three of these input features increase the probability of default and have two to three plus marks against them, indicating that the effects are material. We can apply business knowledge and common sense to confirm that this prediction makes sense and craft a narrative, as the borrower has already borrowed close to their limit and wants more money to fund a business that is unlikely to make a profit.

2. The row third from the bottom, with ID 1681, has a low probability of the loan going bad because the borrower has 6.7% utilization of their line of credit facility, an annual income of $120,000, and zero inquiries in the past six months. All three of these input features reduce the probability of default, and have two to three minus marks against them, indicating that the effects are material. We can apply business knowledge and common sense to confirm that this prediction makes sense and craft a narrative, as the borrower is a high-income earner who has hardly used their line of credit facility and hasn't been actively borrowing money.

features with the greatest effect upon the prediction are on the left, and the ones with the weakest effects are to the right of the table. Green plus signs indicate that the input feature caused the prediction to be higher, and the number of plus signs indicates the relative strength of that effect. Similarly, red minus signs indicate that the input feature caused the prediction to be lower.



Placing the mouse over a feature value will show a window displaying the full contents of that input feature. In the screenshot above, you can see a loan description text feature that describes how the loan applicant wants to borrow money to fund a new business that films documentaries.

The path to trusting an AI includes knowing whether the way it is using the data is suitable and reasonable. The path to building an AI involves training multiple machine learning algorithms to find the one that best suits your needs, and the only practical way for you to quickly find the model that is suitable is to use automated machine learning, which generates the prediction explanations for each and every datapoint, for each and every model.

If your AI is a black box that can't explain the predictions and decisions that it makes, then it's time to update to DataRobot for AI that you can trust.

Conclusion

The path to trusting your AI solution involves certain key steps that will make all the difference in finding an optimal outcome to your business problem.

First, do not settle for black-box models. Look for interpretable models that will explain in human-friendly terms how they reached the conclusion they reached. You should be able to ask of your model the same kinds of questions you would ask a human in order to understand how it came to its conclusion, and you should understand why you can trust that conclusion.

In addition, finding the best model means finding the one that is the most accurate. In order to do this, it is a good idea to hold competitions between algorithms and let the results sort the most accurate from the least accurate. The only practical way for you to quickly find the model that is suitable is to use automated machine learning, which generates visualizations of the pipelines of each and every blueprint.

Finally, you should be able to quickly find the key points of your data and understand the patterns it produces. These insights are what will help you discern the issues and deploy solutions into production.

If your AI can't answer these questions and give you real and interpretable models that you can understand, then it's time to upgrade to DataRobot for models that you can trust.

For a DataRobot demo, visit datarobot.com/contact-us.

DataRobot

DataRobot helps enterprises embrace artificial intelligence (AI). Invented by DataRobot, automated machine learning enables organizations to build predictive models that unlock value in data, making machine learning accessible to business analysts and allowing data scientists to accomplish more faster. With DataRobot, organizations become AI-driven and are enabled to automate processes, optimize outcomes, and extract deeper insights.

Sign up for a free trial today to find out how DataRobot can help your organization at **datarobot.com**