White Paper
Vertica

Unlock Machine Learning for the New Speed and Scale of Business

Built into Vertica's core–with no need to download and install separate packages–in-database machine learning supports the entire predictive analytics process with massively parallel processing and a familiar SQL interface, allowing data scientists and analysts to embrace the power of Big Data and accelerate business outcomes with no limits and no compromises.

Table of Contents

Vertica's in-database machine learning allows you to embrace the power of big data and accelerate business outcomes with no limits and no compromises.

Machine Learning: A Competitive Advantage

In today's data-driven world, creating a competitive advantage depends on your ability to transform massive volumes of data into meaningful insights. Companies that use advanced analytics and machine learning are twice as likely to be top quartile financial performers, and three times more likely to execute effective decisions.[1] However, while 75% of business leaders identify "growth" as the key source of value from analytics, only 60% of these leaders have predictive analytics capabilities.[2]

Leveraging big data to the full can help provide an in-depth understanding of customer behavior, enabling you to personalize the consumer experience, prevent churn, detect fraud, and increase your bottom line. Unfortunately, the growing velocity, volume, and variety of data has increased the complexity of building predictive models, since few tools are capable of processing these massive datasets at the speed of business. Vertica's in-database machine learning, on the other hand, allows you to embrace the power of big data and accelerate business outcomes with no limits and no compromises.

What Is Machine Learning, and Why Is It Important?

Machine learning is gaining popularity as an essential way of not only identifying patterns and relationships, but also predicting outcomes. This is creating a fundamental shift in the way businesses are operating—from being reactive to being proactive. Machine learning enables processes and analyses that were previously too complex to handle manually. Comprising a variety of algorithms and statistical models to process and correlate data, machine learning helps data scientists and analysts unlock valuable insights from seemingly random or unrelated datasets. These insights—about customers, value chain processes, manufacturing inputs, and more—can be used to drive business decisions and improve your competitive standing. Simply put, machine learning algorithms facilitate more informed organizational decision making, and should be an essential input to your daily business functions, products, and operations.

To capture the full value of big data, however, we need to change the scale and speed through which these machine learning algorithms are modeled, trained, and deployed. Traditional analytics tools no longer scale in the world we live in. Today's data volumes are far too large for many traditional machine learning tools to handle, and the relationships between large, disparate data sources—from back-end customer databases to clickstream behavior—are far too complex for traditional tools to derive all hidden value available.

1 *"Creating Value through Advanced Analytics." Bain & Company, February 2015*

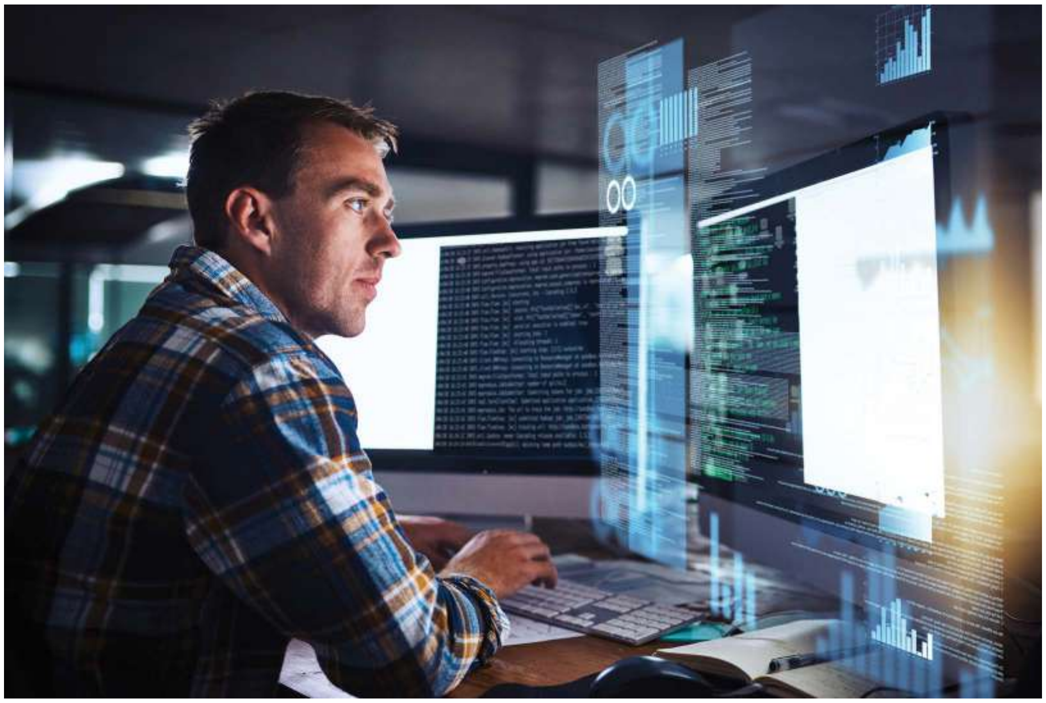
2 **www.marutitech.com/machine-learning-predictive-analytics/** *(see comment above).*

New in-database machine learning methods free data scientists from the volume constraints of traditional tools and enable them to discover and display patterns buried in ever-larger datasets. And as the volume of data ingested increases, the level and sophistication of learning also increases, resulting in more accurate predictions that can be turned into better customer service, superior products, and a competitive advantage.

"In the same way that a bank without databases can't compete with a bank that has them, a company without machine learning can't keep up with the ones that use it."[4]

In 2016, Google AlphaGo made history by becoming the first program to beat a professional player of Go—an ancient Chinese game with more possible board configurations than there are atoms in the universe.[3] How did it do it? By using machine learning algorithms to study a database of online Go matches—equivalent to the experience one would gain by playing Go for 80 years without a break.

The same is true of Google's self-driving car. According to Pedro Domingos, one of the world's leading experts on machine learning, "a self-driving car is not programmed to drive itself. Nobody actually knows how to program a car to drive. We know how to drive, but we can't even explain it to ourselves. The Google car learned by driving millions of miles, and observing people driving."[4]



3 *"Google just made artificial-intelligence history." Business Insider, March 2016.*
4 *"The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World." Pedro Domingos, September 2015.* **www.amazon.com/The-Master-Algorithm-Ultimate-Learning/dp/0465065708**

With the ever-increasing volume, velocity, and variety of data, it's critical that organizations discover the hidden insights contained in big data to create competitive differentiation and increase profitability.

Using the same machine learning algorithms, Netflix offers movie recommendations based on your viewing history. Amazon makes product recommendations based on your purchasing trends. Facebook identifies and tags your friends' faces in the photos you upload, and the list goes on. The benefit? A personalized consumer experience, increased loyalty, reduced churn, and a greater share of wallet. All by leveraging the power of machine learning.

Purpose of This White Paper

With the ever-increasing volume, velocity, and variety of data, it's critical that organizations discover the hidden insights contained in big data to create competitive differentiation and increase profitability. This white paper explains how in-database machine learning eliminates the constraints of small-scale analytics when applied to extremely large data sets for increased accuracy at lightning speeds. It also explains the unique benefits of Vertica's in-database machine learning algorithms and how these complement traditional tools (C++, Java, Python, and R).

Type of Machine Learning

Machine learning algorithms can be divided into two broad categories: supervised learning and unsupervised learning. Supervised learning and unsupervised learning are the two most widely adopted methods of machine learning, with algorithms for both built into Vertica.

- **Supervised learning** is used in cases where all data is labeled and the algorithms learn to predict the output from the input data. The learning algorithm ingests a set of training inputs together with a corresponding set of correct outputs. It then learns by comparing the actual output with the correct outputs, finding errors, and modifying the model accordingly. The process is repeated until the model provides the desired level of accuracy based on the training data.

  Supervised learning often uses historic data to predict future events—for example, the best customer demographic to target for a promotion based on past buying behavior, or predicting credit scores based on past financial behavior. Popular algorithms include decision trees, Naïve Bayes classification, random forest, linear regression, and logistic regression.

- **Unsupervised learning** is used in cases where all data is unlabeled and the algorithms learn the inherent structure from the input data. There are no correct answers and no "teacher," unlike supervised learning.

  The algorithm is responsible for analyzing the data and identifying a pattern—for example, customers with similar characteristics that can be collectively targeted in a marketing campaign. Popular classifications of techniques for unsupervised learning include association rule learning and clustering techniques such as hierarchal clustering and K-means.

Barriers to Applying Machine Learning at Scale

There are several challenges when it comes to applying machine learning to the massive volumes of data that organizations collect and store. Predictive analytics can be complex, especially when big data is added to the mix. Since larger data sets yield more accurate results, high-performance, distributed, and parallel processing is required to obtain insights at the speed of business. In addition, machine learning algorithms must be rewritten to take advantage of modern distributed and parallel engines.

Traditional machine learning applications and tools require data scientists to build and tune models using only small subsets of data (called down-sampling), often resulting in inaccuracies, delays, increased costs, and slower access to critical insights:

- **Slower development:** Delays in moving large volumes of data between systems increases the amount of time data scientists spend creating predictive analytics models, which delays time-to-value.
- **Inaccurate predictions:** Since large data sets cannot be processed due to memory and computational limitations with traditional methods, only a subset of the data is analyzed, reducing the accuracy of subsequent insights and putting at risk any business decisions based on these insights.
- **Delayed deployment:** Owing to complex processes, deploying predictive models into production is often slow and tedious, jeopardizing the success of big data initiatives.
- **Increased costs:** Additional hardware, software tools, and administrator and developer resources are required for moving data, building duplicate predictive models, and running them on multiple platforms to obtain the desired results.

Built from the ground up to handle massive volumes of data, Vertica is designed specifically to address the challenges of big data analytics using a balanced, distributed, compressed columnar paradigm.

"Vertica's new in-database machine learning capabilities are like gold! We are extremely excited to train our machine learning models on our data in Vertica and ship them with our platform to run on our customers' clusters. This is something that is much harder with any other tool."—Abhishek Sharma, Data Scientist, Fidelis Cybersecurity

Vertica Analytics Platform Delivers Predictive Analytics at Speed and Scale

Capable of storing large amounts of diverse data and key built-in machine learning algorithms, Vertica Analytics Platform eliminates or minimizes many of these barriers. Built from the ground up to handle massive volumes of data, Vertica is designed specifically to address the challenges of big data analytics using a balanced, distributed, compressed columnar paradigm.

Massively parallel processing enables data to be handled at petabyte scale for your most demanding use cases. Its columnar store capabilities provide data compression, reducing big data analytics query times from hours to minutes or minutes to seconds, compared to legacy technologies. In addition, as a full-featured analytics system, Vertica provides advanced SQL-based analytics including pattern matching, geospatial analytics, Monte Carlo simulations, and many more capabilities.

As an optimized platform enabling advanced predictive modeling to be run from within the database and across large data sets, Vertica eliminates the need for data duplication and processing on alternative platforms—typically requiring multi-vendor offerings—that add complexity and cost. Now that same speed, scale, and performance used for SQL-based analytics can be applied to machine learning algorithms, with both running on a single system for additional simplification and cost savings.

Implementing Machine Learning with the Vertica Analytics Platform

Machine learning is most effective when applied to very large data sets and thus is a natural fit for Vertica which is designed for fast processing of big data. There are two primary ways of deploying machine learning capabilities in Vertica: Vertica's in-database machine learning, and user-defined extensions (UDxs).

In-Database Machine Learning

With Vertica's in-database applied machine learning algorithms, some of the most commonly used machine learning models can be created and deployed natively for analyzing large data sets, accelerating decision making with pinpoint accuracy. Built into Vertica's core—with no need to download and install separate packages—the in-database machine learning algorithms deliver:

- **Scalability:** While most external tools like R and Python have limitations on the size of the data set they can handle—forcing users to down-sample for analysis and reducing the benefits of analyzing large volumes of data—Vertica's in-database machine learning takes advantage of the larger data sets supported to provide increased and more accurate insights.

- **Simplicity:** Vertica's native ingest, data preparation, and model management features cover the entire data mining lifecycle, eliminating the need to export and load data into another tool for analysis, and then export the results back into Vertica. In addition, users can train, test, and deploy machine learning models using a familiar, SQL-like interface, without having to learn new techniques or hire expensive resources with niche skills.
- **Speed:** Vertica's in-database machine learning accelerates time-to-insight by leveraging Vertica's massively parallel processing (MPP) architecture, including the use of multiple nodes in the cluster for faster computation when required.

Vertica provides several machine learning functions for performing in-database predictive analytics on large data sets to increase accuracy of predictions and accelerate access to hidden insights.

User-Defined Extensions (UDxs)

Vertica connects to hundreds of applications, data sources, ETL, and visualization modules, and what it doesn't connect to out of the box can be easily integrated using a UDx. A UDx allows you to develop your own analytic or data-loading tools for the Vertica Analytics Platform, including new types of data analysis and the ability to parse and load new types of data. Developed in C++, Java, Python, and R programming languages using the Vertica SDK, they're best suited for analytic operations that are typically difficult or slow to perform using standard SQL.

The broad array of user-defined capabilities (functions, transforms, aggregates, analytics, and loading) leverages the MPP capabilities of Vertica, increasing the power and flexibility of procedural code by bringing it closer to the data (structured, semi-structured, or unstructured). Vertica's user interface makes it easy to deploy and use procedural extensions, simplifying operational practices and promoting code reuse. However, although UDxs can expand data analytics functionality within Vertica, they don't match the speed or scale of Vertica's in-database machine learning capabilities.

In-Database Machine Learning Features

Vertica provides several machine learning functions for performing in-database predictive analytics on large data sets to increase accuracy of predictions and accelerate access to hidden insights. Some of the major use cases for machine learning applications revolve around classification, clustering, and prediction. Vertica's built-in machine learning algorithms cover all of these areas with K-means, linear regression, logistic regression, and Naïve Bayes.

The k-means algorithm is a type of unsupervised learning algorithm, meaning the input data is unlabeled. The purpose of k-means is to partition n observations into k clusters.

End-to-End Machine Learning Management

From data preparation to model scoring and deployment, Vertica supports the entire machine learning process. Users can prepare data with functions for normalization, outlier detection, sampling, imbalanced data processing, missing value imputation and more. Machine learning models can be created, trained and tested on massive data sets, and then evaluated with model-level statistics such as ROC tables and confusion matrices.

From data prep to deployment, Vertica supports the entire machine learning process.

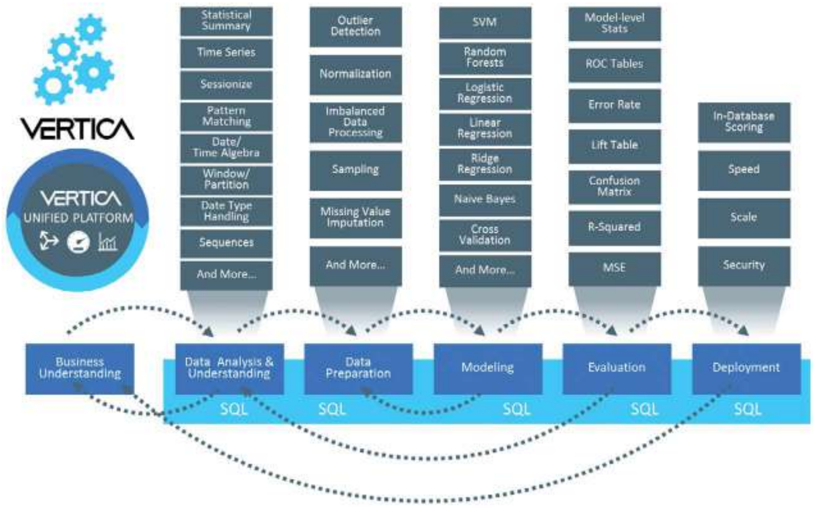


Fig. 1

End-to-End Machine Learning Management

K-Means Clustering

Clustering detects natural groupings within the data, items within a cluster having more in common with each other than they do with items outside of the cluster.

The k-means algorithm is a type of unsupervised learning algorithm, meaning the input data is unlabeled. The purpose of k-means is to partition n observations into k clusters.

The algorithm takes the unlabeled data and clusters the data points into k different clusters based on similarities between the data points. Through this partitioning, k-means assigns each observation to the cluster with the nearest mean. That nearest mean is also known as the cluster center. The resulting model can be used to add new data to existing clusters later by assigning it to the cluster with the nearest mean or cluster center.

K-means has a wide number of applications, including:

- **Customer segmentation:** Segment customers and buyers into distinct groups (clusters) based on similar attributes such as age, income, or product preferences, to target promotions, provide support, and explore cross-sell opportunities.
- **Fraud detection:** Identify individual observations that don't align to a distinct group (cluster) and identify types of clusters that are more likely to be at risk of fraudulent behavior.

You can apply the K-means algorithm using basic SQL syntax:

```
KMEANS ( 'model_name', 'input_table', 'input_columns', 'num_clusters')
```

The optional parameters for this function are columns to be excluded, epsilon value, maximum number of iterations, initial cluster center determination method, table with initial cluster centers, distance method, output view, and key columns.

Function Name	Functionality
kmeans	Find k cluster centers for an input table/view, optionally output each row with its assigned cluster.
summarize_model	Display the cluster centers along with some evaluation metrics.
apply_kmeans	Given an existing k-means model, assign rows from an input table to the correct cluster.

Logistic Regression

The logistic regression algorithm is used to classify data into groups based on the logical relationship between independent variables, or features, and some dependent variable, or outcome. The outcome of logistic regression is a binary value which represents an outcome such as true/false, pass/fail, yes/no, 1/0. You can build logistic regression models to:

- Use a loan applicant's credit history, income, and loan conditions (predictors) to determine the probability that the applicant will default on a loan (response). The result can be used for approving, denying, or changing loans terms.
- Predict the likelihood that a particular mechanical part of a system will malfunction or require maintenance (response) based on operating conditions and diagnostic measurements (predictors).
- Determine the likelihood of a patient's successful response to a particular medicine or treatment (response) based on factors like age, blood pressure, smoking, and drinking habits (predictors).

The logistic regression algorithm is used to classify data into groups based on the logical relationship between independent variables, or features, and some dependent variable, or outcome. Linear regression is primarily used to predict continuous numerical outcomes in linear relationships along a continuum.

Linear regression is primarily used to predict continuous numerical outcomes in linear relationships along a continuum.

You can apply the logistic regression algorithm using basic SQL syntax:

LOGISTIC_REG ('model_name', 'input_table', 'response_column', 'predictor_columns')

The optional parameters for this function are columns to be excluded, optimizer type, epsilon value, maximum number of iterations, and type of output desired (binary, probability).

Function Name	Functionality
logistic_reg	Training (two optimization techniques: BFGS and Newton)
summarize_model	Reading model summary
predict_logistic_reg	Scoring

Linear Regression

Linear regression is primarily used to predict continuous numerical outcomes in linear relationships along a continuum.

Using linear regression, you can model the linear relationship between independent variables, or features, and a dependent variable, or outcome. For example, linear regression models allow you to:

- Model residential home prices (response) as a function of the homes' features (predictors), such as living area, number of bedrooms, or number of bathrooms, etc.
- Model the demand for a service or good (response) based on its features (predictors)—for example, demand for different models of laptops based on monitor size, weight, price, operating system, etc.
- Determine linear relationship between the compressive strength of concrete (response) and varying amounts of its components (predictors) like cement, slag, fly ash, water, super plasticizer, coarse aggregate, etc.

You can apply the linear regression algorithm using basic SQL syntax:

LINEAR_REG ('model_name', 'input_table', 'response_column', 'predictor_columns')

The optional parameters for this function are columns to be excluded, optimizer type, epsilon value, and maximum number of iterations.

Function Name	Functionality
linear_reg	Training (two optimization techniques: BFGS and Newton)
summarize_model	Reading model summary
predict_linear_reg	Scoring
mse	Evaluation (mean squared error)
rSquared	Evaluation (R-squared value)

Naïve Bayes

The Naive Bayes algorithm is used to classify data. The algorithm uses different features as predictors to calculate the probability of a specific class, or more than one class. For instance, to predict the probability that an email is spam, you would use words normally associated with spam. The same would be true if you want to classify a document based on content—for example, news, finance, sport, etc.

This supervised machine learning algorithm has several applications, including:

- Spam filtering
- Classifying documents
- Classifying images
- Consumer habits

You can apply the Naïve Bayes algorithm using basic SQL syntax:

```
NAIVE_BAYES ( 'model_name', 'input_table', 'response_column', 'predictor_columns')
```

The optional parameters for this function are columns to be excluded and alpha value. In addition to the above machine learning algorithms, Vertica provides in-database machine learning functions for Support Vector Machines (SVM) and Random Forest. Learn more at: **www.vertica.com/machinelearning**

Vertica has a rich set of built-in analytic functions, including interpolation, pattern matching, event series joins, advanced aggregation, outlier detection, and sequences. These functions support the data preparation process, increase the efficiency of data analysis work, and ultimately speed up time-to-value.

Data Preparation and Normalization

Data preparation is an important pre-processing step for data analysis, often occupying most of the time it takes data scientists and analysts to complete a data analysis project. Vertica has a rich set of built-in analytic functions, including interpolation, pattern matching, event series joins, advanced aggregation, outlier detection, and sequences. These functions support the data preparation process, increase the efficiency of data analysis work, and ultimately speed up time-to-value.

Data scientists can also use additional functions—such as data normalization—to prepare the data before training a machine learning model. The purpose of normalization is, primarily, to organize different scales of numeric data to an equivalent scale. Without such data normalization functions, machine learning algorithms may exhibit poor performance if the values of different features include large variations. Machine learning for predictive analytics offers two data preparation methods, which utilize normalization, MinMax, and Z-score.

If you need to consider a categorical predictor for training a linear regression model in Vertica, convert it to a numerical value in advance. There are several techniques for converting a categorical variable to a numeric one. For example, you can use one-hot encoding.

Real-World Implementation of Vertica's In-Database Machine Learning Features

The following is an example of a linear regression model that uses the Prestige Data Set[5]. This data set is used to determine the relationship between income and other variables. In this example, we will demonstrate how to load the data set and use Vertica's linear_reg function to better understand how the predictors impact the response.

The data set contains the following information:

- Name of occupation
- Education (in years)
- Income—Average income of incumbents in dollars in 1971
- Women—Percentage of incumbents who are women
- Prestige—The Pineo-Porter prestige score for the occupation, from a social survey conducted in the mid-1960s

5 *"Prestige Data Set." Statistics Canada. Canada (1971) Census of Canada. Vol. 3, Part 6.*

- Census—Canadian Census occupational code
- Type—Occupational type, where bc indicates blue collar, wc indicates white collar, and prof indicates professional, managerial, or technical

The goal is to build a linear regression model using this data set that predicts income based on the other values in the data set and then evaluate the model's goodness of fit.

To begin, how do we choose which variables to use for this model? The type column can be eliminated because Vertica does not currently support categorical predictors. Since the occupation and census columns contain a lot of unique values, they are unlikely to predict income under this use case. That leaves the education, prestige, and women columns.

NOTE: In practice, if you need to consider a categorical predictor for training a linear regression model in Vertica, convert it to a numerical value in advance. There are several techniques for converting a categorical variable to a numeric one. For example, you can use one-hot encoding.

There are several techniques for converting a categorical variable to a numeric one. For example, you can use one-hot encoding.

Step 1. Load the Data

The following shows the table definition to store the Prestige Data Set:

```
=> DROP TABLE IF EXISTS public.prestige CASCADE;
=> CREATE TABLE public.prestige
      ( occupation VARCHAR(25), education NUMERIC(5,2), -- avg years of education
      income INTEGER, -- avg income
      women NUMERIC(5,2), -- % of woman
      prestige NUMERIC(5,2), -- avg prestige rating
      census INTEGER, -- occupation code
      type CHAR(4) -- Professional & managerial (prof)
      )             -- White collar (wc)
                    -- Blue collar (bc)
                    -- Not Available (na)
```

In Vertica, the PREDICT_
LINEAR_REG function
applies a linear regression
model on the input table.
You can read more
about this function in the
Vertica documentation.

To load the data from the Prestige Data Set into the Vertica table, use the following SQL statement:

```
=> COPY public.prestige
FROM stdin
        DELIMITER ','
        SKIP 1
        ABORT ON ERROR
        DIRECT
;
```

Step 2. Create a Linear Regression Model

Now let's use the Vertica machine learning function LINEAR_REG to create the linear regression model.

To create the model, run the LINEAR_REG function against the public.prestige table as shown. In this statement, income is the response, and the predictors are education, women, and prestige:

```
=> SELECT LINEAR_REG( 'prestige', 'public.prestige', 'income', 'education,women,prestige');
```

This statement is trying to identify the coefficients of the following equation:

$$income = \alpha + \beta_1 education + \beta_2 women + \beta_3 prestige \tag{4}$$

After you create the model, use the SUMMARIZE_MODEL function to observe the model's properties:

```
=> SELECT SUMMARIZE_MODEL('prestige');
```

SUMMARIZE_MODEL returns the following information:

```
SUMMARIZE_MODEL| coeff names : {Intercept, education, women, prestige}
coefficients: {–253.8390442, 177.1907572, –50.95063456, 141.463157}
p_value: {0.83275, 0.37062, 4.1569e–08, 8.84315e–06}
```

Using these coefficients, you can rewrite equation (4) to read:

$$\begin{aligned} income = &- 253.8390442 \\ &+ 177.1907572 * education \\ &- 50.950663456 * women \\ &+ 141.463157 * prestige \end{aligned} \tag{5}$$

Finally, let's explore how to measure how well the linear regression model fits the data, or goodness of fit. In Vertica, the PREDICT_LINEAR_REG function applies a linear regression model on the input table. You can read more about this function in the Vertica documentation.

Step 3. Evaluate Goodness of Fit

A common method used to test how well your linear regression model fits the observed data is the coefficient of determination. The coefficient is defined in the following equation:

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (y_i - \bar{y})^2} \qquad (6)$$

The coefficient of determination R2 ranges between 0 (no fit) and 1 (perfect fit). To calculate the coefficient of determination, use the Vertica RSQUARED function:

```
=> SELECT RSQUARED(income, predicted) OVER()
FROM ( SELECT
        income,
        PREDICT_LINEAR_REG (
                prestige, women
                USING PARAMETERS OWNER='dbadmin',
                MODEL_NAME='prestige')
        AS predicted
        FROM public.prestige
        ) x ;
```

rsq	Comment
0.63995924449805	Of 102 rows, 102 were used

NOTE: The OWNER parameter will be deprecated in Vertica 8.1.

The evaluation of the coefficient of determination often depends on what area you are investigating. In the social sciences, a coefficient of 0.6 is considered quite good.[6]

When evaluating a model, it is important to consider multiple metrics. A single metric could give you a good value, but the model itself may not be as useful as you need. It is important to understand the R-square value, as well as the other metrics, to evaluate the goodness of fit.

Use Cases for Machine Learning across Different Industries

The in-database machine learning capabilities of the Vertica Analytics Platform can be used to drive tangible business benefits across a broad range of industries.

The in-database machine learning capabilities of the Vertica Analytics Platform can be used to drive tangible business benefits across a broad range of industries.

Vertica's in-database machine learning capabilities allow users to take advantage of big data while simplifying and speeding up their predictive analytics processes to make better-informed decisions, compete more effectively, and accelerate time-to-insight.

- **Financial services** organizations can discover fraud, detect investment opportunities, identify clients with high-risk profiles, or determine the probability of an applicant defaulting on a loan.
- **Government** agencies, such as public safety and utilities, can use machine learning to help detect fraud, minimize identity theft, and analyze data from smart meters to identify ways to increase efficiency and save money.
- **Communication service providers** can leverage a variety of network probe and sensor data to analyze network performance, predict capacity constraints, and ensure quality service delivery to end customers.
- **Marketing and sales** organizations can use machine learning to analyze buying patterns, segment customers, personalize the shopping experience, and implement targeted marketing campaigns.
- **Oil and gas** organizations can leverage machine learning to analyze minerals to find new energy sources, streamline oil distribution for increased efficiency and cost-effectiveness, or predict mechanical or sensor failures for proactive maintenance.
- **Transportation** organizations can analyze trends and identify patterns that can be used to enhance customer service, optimize routes, and increase profitability.

Summary

Vertica's in-database machine learning capabilities allow users to take advantage of big data while simplifying and speeding up their predictive analytics processes to make better-informed decisions, compete more effectively, and accelerate time-to-insight. At the same time, it offers freedom of choice with support for user-defined extensions programmed in C++, Java, Python, or R.

To experience the benefits of Vertica in your own environment:

- Watch the webcast Building, training & operationalizing predictive analytics models with Vertica.
- Read the blog posts What's new in Vertica Machine Learning 8.0? and Machine Learning with Vertica.
- Download a free version of the Vertica Analytics Platform.

Resources

Machine Learning for Predictive Analytics, Vertica documentation.

Machine Learning Functions, Vertica documentation.

Machine Learning for Predictive Analytics, Hands On Vertica.

Learn More At

vertica.com/machinelearning

Vertica Headquarters
150 Cambridgepark Drive
Cambridge, MA 02140

Learn more at: **www.vertica.com**

www.vertica.com