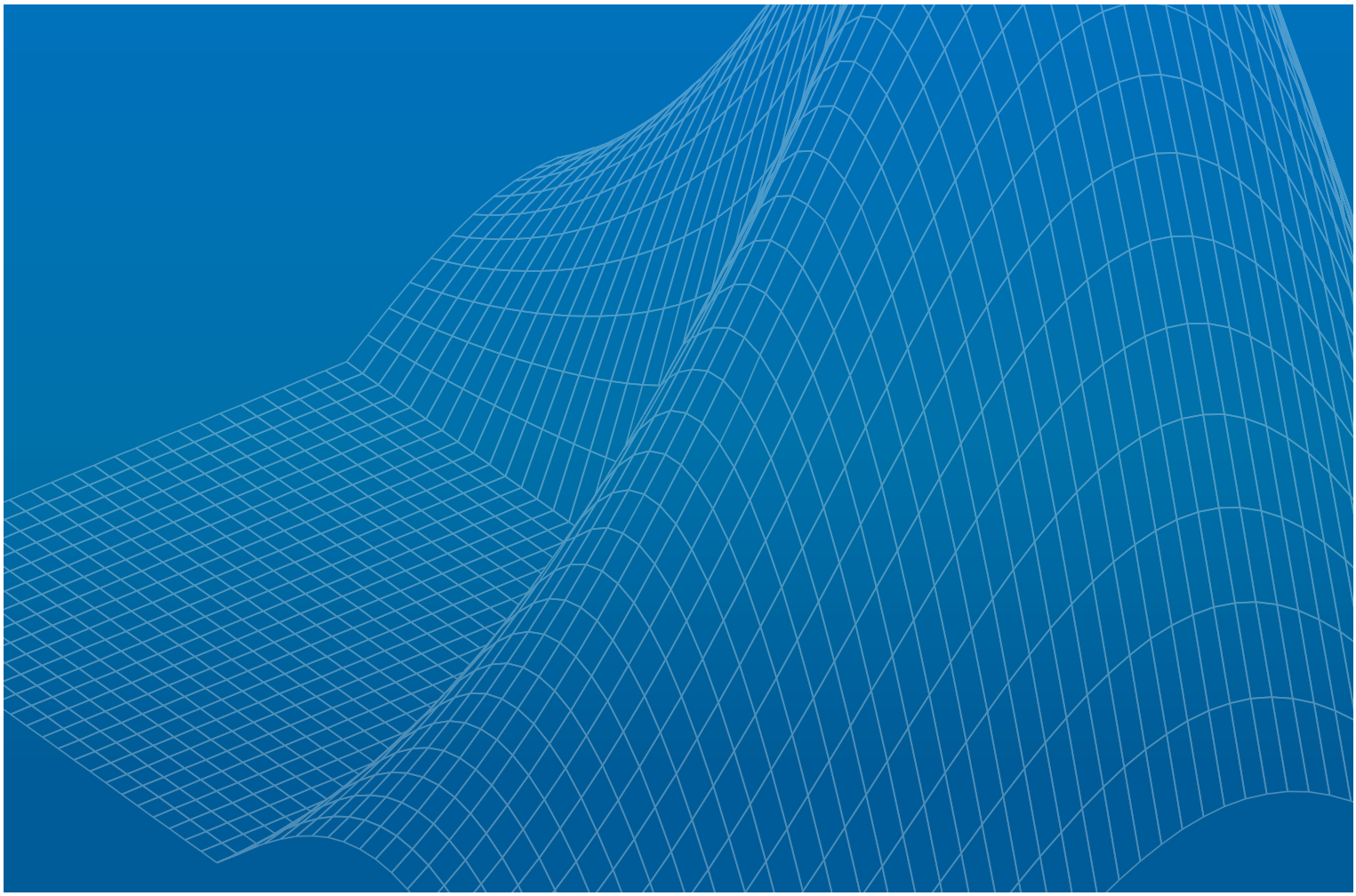


Using Analytics and Machine Learning to Build Intelligent Products and Services

Engineers and designers are building smarter products and services driven by analytics based on business and engineering data. This paper describes how these smart, real-time systems work and illustrates the power of using predictive algorithms and controls in real-time applications. Using examples from a variety of industries, we'll show how you can incorporate analytics into your production and embedded systems.

WHITE PAPER

MathWorks®

Introducing Analytics-Driven Systems

Analytics-driven embedded systems are here. The ability to create analytics that process massive amounts of business and engineering data enables designers in many industries to develop intelligent products and services. Designers can use analytics to describe and predict a system's behavior, and further combine analytics with embedded control systems to automate actions and decisions.

Cloud-Based, Real-Time System for Optimizing Energy Management

In some implementations, the analytics are performed in the cloud to improve embedded-system performance. Borislav Savkovic, a controls systems engineer by training, led a team at ***BuildingIQ*** to design a climate-control system for commercial buildings that uses analytics to reduce energy consumption (Figure 1).

The system starts with gigabytes of engineering and business data. The engineering data comes from power meters, thermometers, pressure sensors, and other HVAC sensors. The business data comes from weather forecasts, real-time energy prices, and demand response data. In the analytics-driven system, the team uses signal processing to remove noise, machine learning to detect spikes, control theory to account for heating and cooling dynamics, and multi-objective optimization with hundreds of parameters. The analytics running in BuildingIQ's cloud service tune the building's HVAC embedded systems. The result: an analytics-driven system that reduces energy consumption up to 25% in commercial buildings.

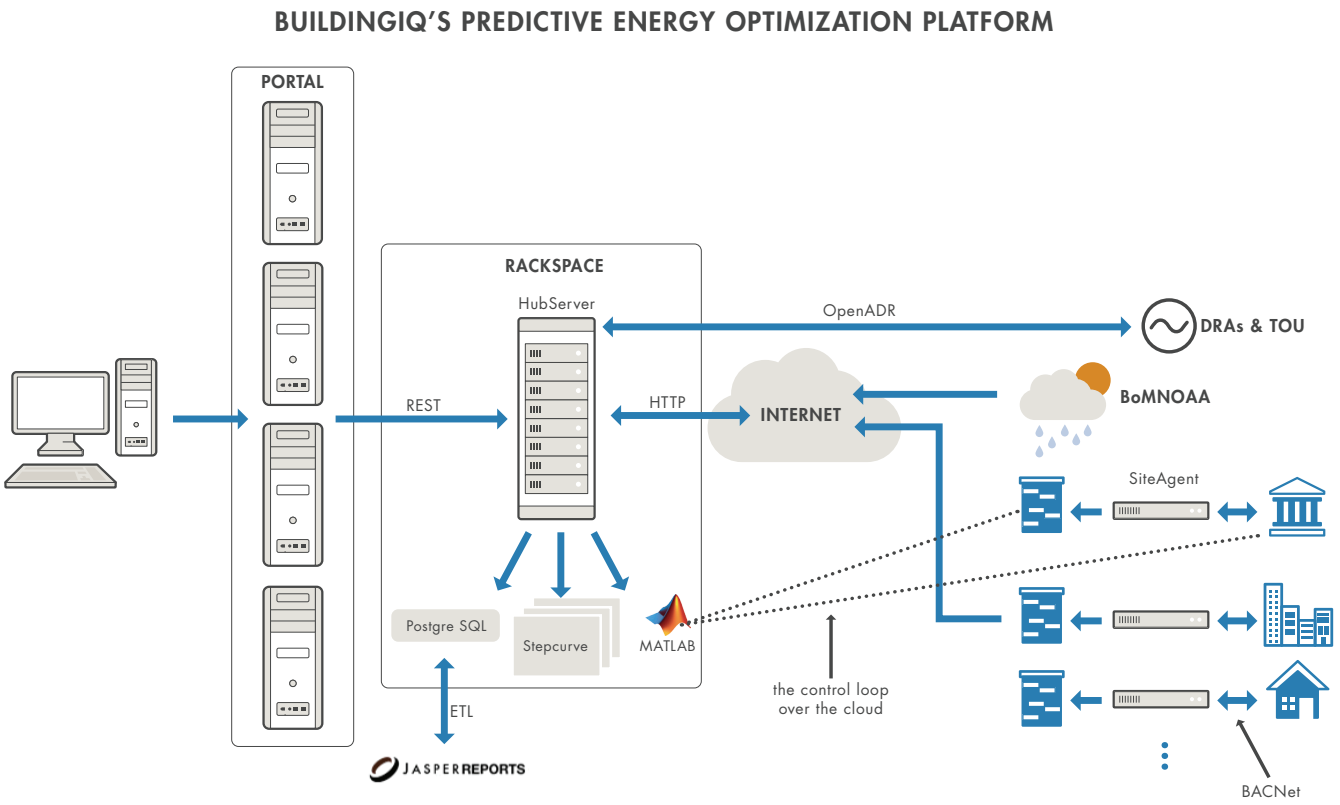


Figure 1: BuildingIQ's predictive energy optimization platform, using analytics to reduce energy consumption up to 25%.

Running Analytics in Embedded Systems

In other cases, the analytics run directly in the embedded systems. The design team at ***Scania***, the Swedish truck manufacturer, embeds analytics into their emergency braking systems to provide real-time crash avoidance to reduce accidents and meet stringent EU regulations. Engineering data from cameras and radar are processed in real time for object detection and road marking detection, and subsequently fused to signal collision warning alerts and automatic brake request (Figure 2). System safety and reliability are ensured with exhaustive test and verification, including test scenario creation, system modeling with simulated and recorded data, and HIL testing.

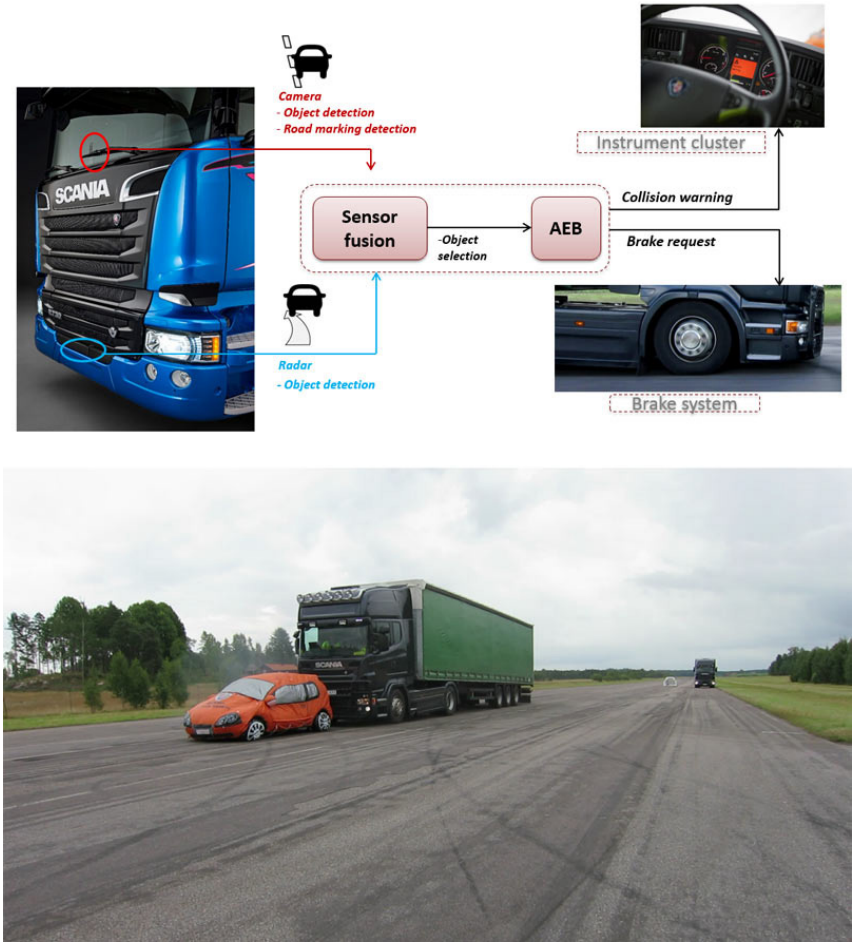


Figure 2: Top: Overview of Scania's real-time emergency braking system. Bottom: A typical scenario, in which a truck with the system installed approaches a slow-moving vehicle.

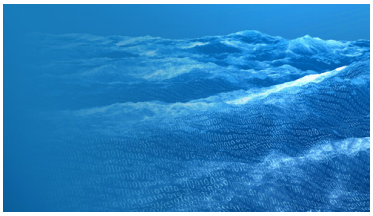
These two examples highlight the steps engineers use in designing analytics-driven systems:

1. Accessing data from varied sources
2. Preprocessing massive amounts of data
3. Developing analytic algorithms
4. Running analytics and controls in real time
5. Integrating analytics with sensors and embedded devices, and possibly other non-embedded resources such as IT systems and the cloud

Accessing Data

The first step in developing analytics is to access the wealth of available data to explore patterns and develop deeper insights. The datasets are not only large in size, but also can come from many different sources and represent many different attributes. Therefore, the software tools you use for exploratory analysis and analytics development should be capable of accessing all the data sources and formats you plan to use. File types might include text, spreadsheet, image, audio, video, geospatial, web, and XML. You may also need application-specific data formats such as CDF/HDF for scientific data and CAN for automotive data. You also should be able to access data from the storage and generation points, such as:

- **Stored data:** Databases, data warehouses, distributed file systems, and Hadoop big data systems
- **Equipment data:** For example, live and historical industrial plant data stored in distributed control systems (DCS), supervisory control and data acquisition systems (SCADA), and programmable logic controllers (PLC).
- **Internet of Things devices:** Such as sensors, local hub or cloud data aggregators



Learn more:

Big Data with MATLAB® (Overview)

Preprocessing Data

A key step is data cleaning and preparation before developing predictive models. For example, data might have missing values or erroneous values, or it might use different timestamp formats. Predictions from erroneous data can be difficult to debug or worse can lead to inaccurate or misleading predictions that impact system performance and reliability. Common preprocessing tasks include:

- Cleaning data that has errors, outliers, or duplicates
- Handling missing data with discarding, filtering, or imputation
- Removing noise from sensor data with advanced signal processing techniques
- Merging and time-aligning data with different sample rates

Another important part of preprocessing is data transformation and reduction. The goal here is to find the most predictive features of the data, and filter data that will not enhance the predictive power of the analytics model. Some common techniques include:

- **Feature selection** to reduce high-dimension data
- **Feature extraction** and **transformation** for dimensionality reduction
- **Domain analysis** such as signal, image, and video processing

There's an increasing need to put more of the data preprocessing and data reduction on the sensor or embedded device itself. There are many reasons for this, but the two most prominent requirements are low power and speed. Many smart devices need to run for an extended period of time without charging. The use of wireless communication to send data to a server or cloud analytics engine can consume a lot of power. Since streaming all raw sensor data can be prohibitively costly, good system designs should do local preprocessing and only upload the useful information or a predictive signal itself. Speed is often paramount as the value of any real-time system is the timely response an interconnected system offers. You may need to embed intelligence in the sensor or embedded device to determine when or how often to communicate with other devices in the network to achieve this desired responsiveness.

The BuildingIQ and Scania examples illustrate the challenges and successful approaches to capturing and preprocessing engineering data and combining it with traditional business data to develop analytics-driven systems.

In the next section, we'll cover advanced analytics algorithms such as machine learning and deep learning, and then we'll show how software tools enable domain experts to develop and run analytics and prescriptive controls in real time.

Predictive Analytic Algorithms and Controls

It's important to consider whether an analytic algorithm is your best approach. In cases where system behavior can be well characterized by known scientific equations, proven ***mathematical modeling*** can be a simple and efficient way to meet design objectives. This approach uses techniques such as data fitting, statistical modeling, ode and pde solving, and parameter estimation. Models built this way have the advantage of being predetermined with historical data or based on first principles. They can be memory and computationally efficient to implement on embedded systems, and can be simpler to develop and maintain. So, before considering data-centric machine learning techniques, it's prudent to first consider if "workhorse" modeling methods can meet your design objectives. However, for an increasing number of design challenges, such as dynamically setting climate control points in the BuildingIQ example or object identification in the Scania braking application, **machine learning** is the best approach.

What is Machine Learning?

Machine learning algorithms use computational methods to "learn" information directly from data without relying on a predetermined equation as a model. It turns out this ability to train models using the data itself opens up a broad spectrum of use cases for predictive modeling – such as financial

credit scoring and online recommendations for movies, songs, and retail purchases. In embedded systems, machine learning is used for a rapidly growing range of applications, including face recognition, tumor detection, electricity load forecasting, and the BuildingIQ and Scania applications mentioned earlier. The increased availability of "big data," compute power, and software tools make it easier than ever to use machine learning in engineering applications.

Machine learning is broadly divided into two types of learning methods, supervised learning and unsupervised learning, each containing several algorithms tailored for different problems.

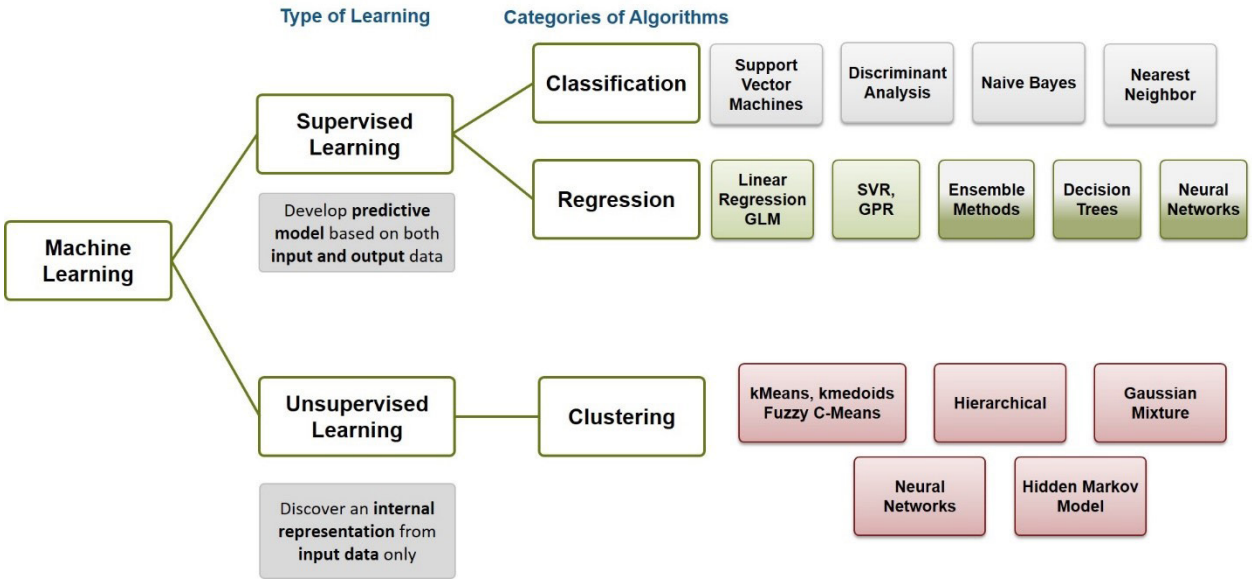


Figure 3: Different types of machine learning methods and categories of algorithms.

Supervised learning is a type of machine learning that uses a known dataset (called the training dataset) to make predictions. The training dataset includes input data and labeled response values. From it, the supervised learning algorithm seeks to build a model that can make predictions of the response values for a new dataset. A test dataset is often used to validate the model. Using larger training datasets often yield models with higher predictive power that can generalize well for new datasets.

Supervised learning includes two categories of algorithms:

- **Classification:** for categorical-response values, where the data can be separated into specific "classes". Common classification algorithms include: support vector machines (SVM), neural networks, Naïve Bayes classifiers, decision trees, discriminant analysis, and nearest neighbor (kNN).
- **Regression:** for prediction when continuous-response values are desired. Common regression algorithms include: linear regression, nonlinear regression, generalized linear models, decision trees, and neural networks.

Choosing algorithms depends on a number of design factors, such as memory usage, prediction speed, and interpretability of the model. Other considerations include whether a single or multiclass

response is needed and if predictors are continuous or categorical. Since the models are only as good as the labeled training data used, it's important to take care in using representative training datasets. The machine learning workflow starts with selecting features, then specifying training and validation sets, training with multiple algorithms, and finally assessing results. An interactive app like the one shown in Figure 4 makes the machine learning workflow easy to learn and use.

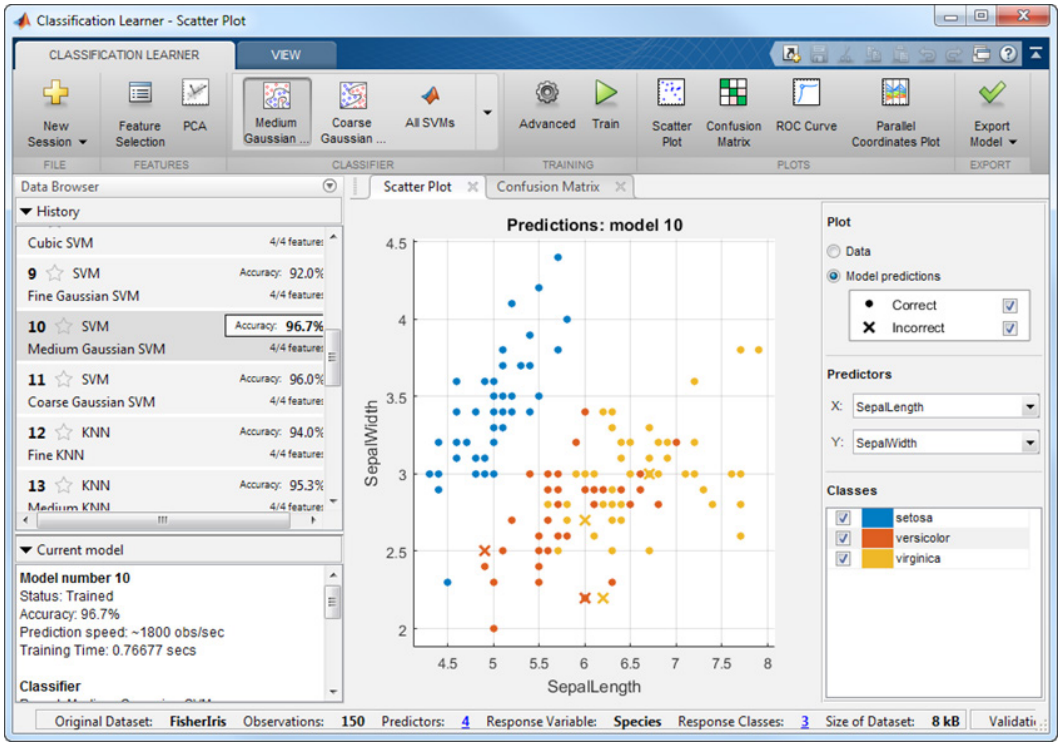
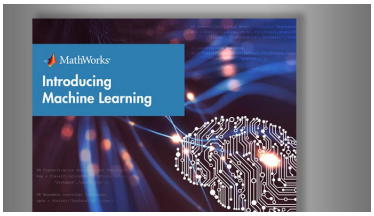


Figure 4: An example of the MATLAB app (Classification Learner app) used to train models for classification. You can explore data, select features, specify cross-validation schemes, train models, and assess results. The resulting models can be exported for use in applications such as computer vision and signal processing.

Unsupervised learning is a type of machine learning used to draw inferences from datasets consisting of input data without labeled responses.

- **Cluster analysis** is the most common unsupervised learning method and is used for exploratory data analysis to find hidden patterns or grouping in data. **k-means is a popular cluster modeling algorithm** that partitions data into k distinct clusters based on a distance measure to the centroid of a cluster. **Hierarchical clustering** uses a different approach of building a multilevel hierarchy cluster tree, which offers visual interpretation but has higher computational requirements, making it less suitable for large amounts of data. Other algorithms include Gaussian mixture models, Hidden Markov models, and Self-organizing neural network maps.

The BuildingIQ team used cluster analysis as part of their model creation process. They used k-means clustering and Gaussian mixture models to segment the data and determine the relative contributions of gas, electric, steam, and solar power to the heating and cooling processes.



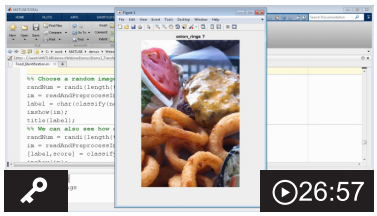
Learn more:

Machine Learning (eBook)

Using Deep Learning for Advanced Analytics

For classification problems involving images, text, and signals, ***deep learning*** has emerged as a new category of advanced analytics. When trained on large labeled training datasets (often requiring hardware acceleration with GPUs and intensive training and assessment), deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance in object classification. For image classification, convolutional neural networks (CNNs) have become popular because they eliminate the need for manual feature extraction by extracting features directly from raw images. This automated feature extraction makes CNN models highly accurate for computer vision tasks such as object classification.

The approaches and algorithms listed above are becoming more accessible to systems engineers to combine effective analytics in their embedded systems. Next, we'll address performing analytics and predictive controls in real time and integrating these into an overall solution, including sensors and embedded systems, as well as enterprise IT systems and cloud infrastructure.



Learn more:

Object Recognition: Deep Learning and Machine Learning for Computer Vision (Video)

Developing Analytics and Prescriptive Controls

Let's look at how to implement the analytics and integrate them in production and embedded systems.

Running Analytics and Controls in Real Time

Figure 5 shows the workflows in which data are accessed and preprocessed, and algorithms are selected. In the training workflow, you use stored data to develop preprocessing code and machine learning to develop a trained model. In the prediction workflow, the same preprocessing code and the trained model are applied to live data to perform real-time analytics.

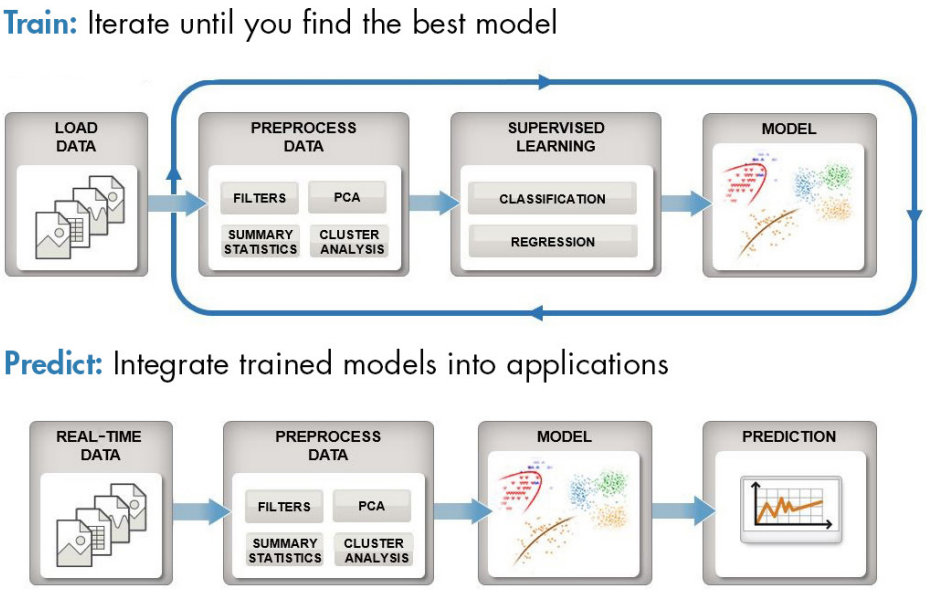


Figure 5: Machine learning workflow for both the training path and the prediction path for real-time analytics.

You can use this workflow to combine engineering, scientific, and field data with business and transactional data. Figure 6 shows examples of data sources for each type. This allows the creation of sophisticated analytics to develop smarter systems.

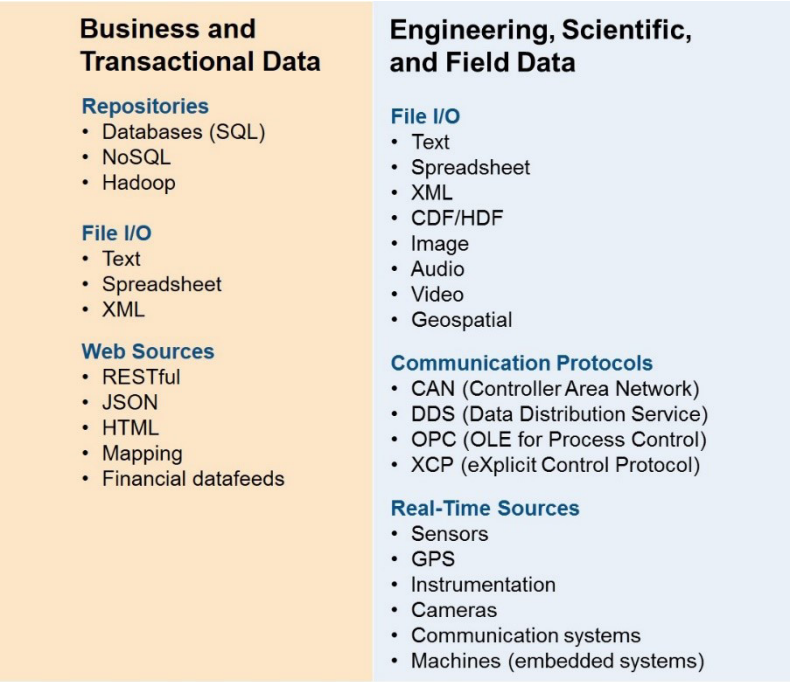


Figure 6: Real-time analytics combining sensor data with data sources from engineering, scientific, and field data, as well as business and transactional data.

Combining sensor-generated data with other real-time sources and historical data is the power behind the Internet of Things (IoT), the machine-to-machine coordination of Industry 4.0, and the automotive trend towards a connected and autonomous vehicle.

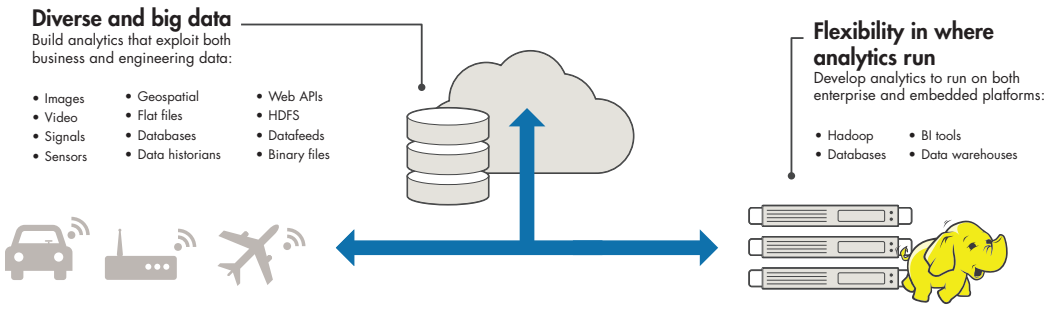


Figure 7: Diverse and big data feeding a data analytics system.

Integrating Analytics in IT Systems and the Cloud

In some system implementations, such as the energy management system developed by ***BuildingIQ***, analytics are performed in enterprise IT systems to improve embedded-system performance. The analytics can be automatically generated as deployable components compatible with IT development environments such as Java, Microsoft .NET, Excel, and C/C++, enabling them to be integrated – without recoding – into web, database, desktop, and scalable enterprise applications running on-premise or in a private or public cloud.

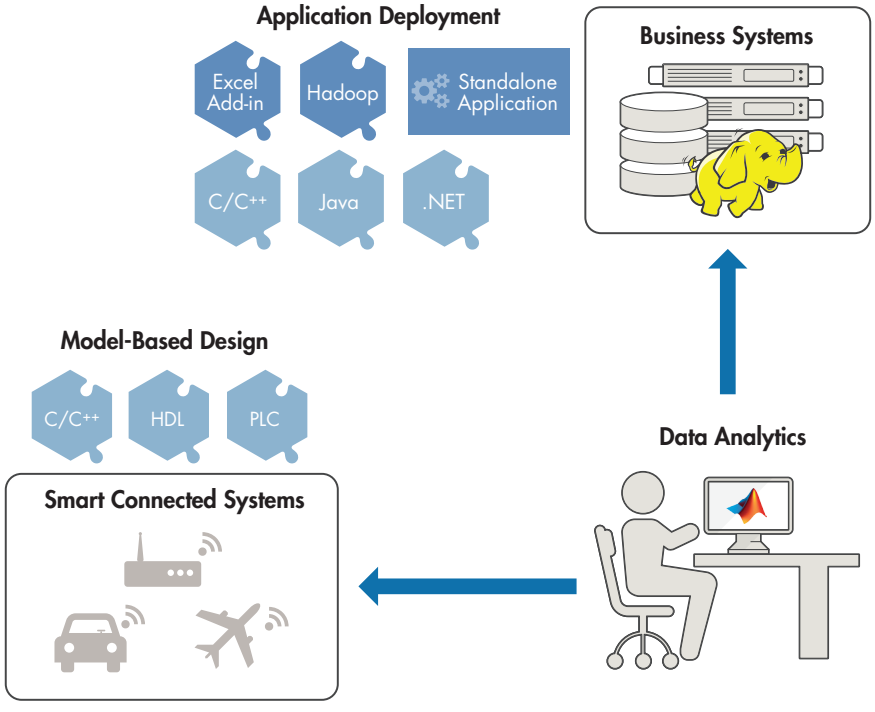


Figure 8: Analytics integrated into business systems (top), smart connected systems (left), or a combination of both.

Integrating Analytics with Sensors and Embedded Devices

Figure 8 also shows a path where data reduction, sensor fusion, or ***predictive analytics*** can be integrated to run directly on embedded systems in smart connected systems. The Scania emergency braking system described above is a good example of this.

The accelerating IoT trend towards smarter and more connected sensors is adding pressure to move more processing and analytics as close to the sensors as possible. This has the benefit of shrinking the amount of data that is transferred over the network, which reduces the cost of transmission and can reduce the power consumption of wireless devices. For embedded system designers, it's important to consider not only algorithm performance, but also the overall system robustness, reliability, and cost in the architecture and design. You can use ***Model-Based Design*** to simulate the system, automatically generate embedded code, and continuously test and verify the analytics being integrated into the embedded system.

An example of this is the innovation in using big data and analytics to make our cars smarter. Automotive OEMs are collecting enormous amounts of data from real-world driving situations (think millions of miles of driving), recording data such as engine performance, video, radar, and other signals. This data helps generate important metrics such as fuel economy and performance at the fleet level. Engineering teams also use this real-world data to design, develop, and test new types of automotive systems, such as ***advanced driver assistance systems*** (ADAS). By combining data analytics and Model-Based Design workflows, ADAS engineers can refine algorithms using large test sets with ground truth labeling, perform rigorous simulation and validation, and then automatically generate code of the validated algorithms for automotive embedded systems.

Learn more:

Internet of Things (IoT) (Overview)

Conclusion

Analytics-driven embedded systems are here. Engineers and data scientists work with large amounts of data in a variety of formats such as sensor, image, video, telemetry, databases, and more. With machine learning techniques, they can find patterns in data and build models that predict future outcomes based on historical data.

This ability to create analytics that process massive amounts of business and engineering data is enabling design engineers in many industries to develop smarter products and services. They can use analytics to describe and predict a system's behavior, and further combine analytics with embedded control systems to automate actions and decisions. With diverse data sources and big data, companies are building increasingly complex and vital applications.

Resources

Machine Learning (eBook)

Data Analytics (Overview)

Essential Tools for Machine Learning (Webinar)

Object Recognition: Deep Learning and Machine Learning for Computer Vision (Video)

Baker Hughes Develops Predictive Maintenance Software for Gas and Oil Extraction Using Data Analytics and Machine Learning (User Story)