8 steps to build a machine learning model

The approach used to build an ML model can be as diverse and unique as the business undertaking the project. However, a successful path typically involves the following eight steps:

1. Define tasks and goals

ML models aren't ubiquitous and are normally designed, built, deployed and maintained as task-specific software elements. The first step in any ML model development is to define the model's tasks or goals. This might involve performing analytics or interpreting spoken words.

It's also important to understand the business value of the ML model and how the model will bring about that value. Both determinations directly affect post-deployment monitoring and the metrics and KPIs used to measure the model's success.

2. Identify required data

Next, businesses must specify the model's data needs. Precisely what does the model need to know? For example, a model built for image classification will require image and video sources to train it and provide a data source for production.

Many sources can supply required data, including databases, file libraries, application programming interfaces and third-party sources. Data science teams must analyze initial training and testing data sources to understand the initial data composition, distribution and patterns. The data science team should also review initial data for bias and relevance.

3. Preprocess required data

Raw data rarely provides the quality and features that an ML algorithm can use directly. Data science teams must preprocess initial data for cleaning and feature engineering. Data cleaning identifies and resolves common data quality issues, including incomplete, inaccurate and irrelevant data elements.

Feature engineering transforms the cleaned data set into the values and scales that the ML algorithm needs. This preprocessed initial data set should be representative of the data provided in production, such as from IoT sensors or live video sources.

4. Create foundational data sets

The creation of an ML model involves three functional phases: training, validation and testing. Training data teaches the model. Validation data is used to fine-tune the model's hyperparameters and prevents overfitting, where the model has trouble providing results outside of its trained data. Testing data checks the model's performance with new, unseen data. The same data shouldn't be used for all three functional phases. Data sets can be version controlled to ensure reproducible outcomes and further compliance evaluation.

5. Build the model

Software development groups, such as a DevOps team, handle ML model development, training, validation, testing and deployment. They choose an algorithm, such as regression or decision tree, that's most appropriate for the model's goals.

As with any software project, the model should be version-controlled to ensure transparency and reproducibility. Some ML model projects benefit from experimentation, using several potential algorithms to determine the one with the best performance and accuracy.

6. Train and evaluate the model

The training data set enables the model to learn patterns and relationships that are key to its intended business task. The validation data set is used to test the model's hyperparameters. Models can be evaluated with various metrics and other measures, such as the Machine Intelligence Quotient. A key part of testing is to compare the model's performance against the current, nonmodel baseline.

7. Optimize the model

Models will typically benefit from some amount of optimization. For example, if the model doesn't perform well on nontraining data, it might be overfit and in need of hyperparameter adjustments until accuracy and performance are optimal.Similarly, it might be necessary to update or enhance training data to address gaps or mitigate ML bias. Finally, the model algorithm might need updates to improve performance or stability.

8. Deploy and monitor the model

Once the ML model is well-developed and validated, teams can deploy it into production where it integrates with existing data sources and performs its tasks in real time. Metrics and KPIs should be applied to the model to provide ongoing measures of accuracy, performance and business value.

Data trends and business needs change over time, and this can cause the model to drift where its accuracy degrades. Metrics and KPIs serve as vital triggers to retrain or update the model. Changing business needs can provide new or updated goals, spawning a new set of steps to build an ML model.