krasamo

WHAT IS MACHINE LEARNING?

By Jose Luis Amoros

What is Machine Learning?

by Jose Luis Amoros May 24, 2020

Machine Learning is an application in which machines can learn from their experiences or train data to make predictions. The Machine Learning approach is different from traditional programming in that the computer learns automatically, detecting patterns and creating its own rules, thereby making it more accurate and easier to maintain.
rules, thereby making it more accurate and easier to maintain.

According to Stanford University professor Andrew Ng, “Machine Learning (ML) is the science of getting computers to act without being explicitly programmed.” Instead of writing code, the user feeds the dataset to the generic algorithm, and the algorithm or the machine will operate with logic based on the given data. Just as our brains use our experience to help us improve at a task, so does the computer.

For example, let’s say you want a computer that can tell the difference between a picture of a car and a picture of a bus. So, you begin updating images that show cars and buses. The computer has to figure out that cars are smaller in size and have several variants, whereas buses are different and larger.

Then, in the future, when the computer sees a picture, it will check the picture’s pattern and decipher if it is a car or a bus. There can (and will) be mistakes, of course, but the algorithm will become more accurate in its predictions over time as it receives more data.
The Machine Learning model is becoming more and more prevalent in daily life in the 21st century. Considered one of the most significant innovations since the microchip, ML has the potential to transform our world in truly mind-blowing ways. Consider how Machine Learning has already impacted our daily lives:

Applications that Use Machine Learning:

- Speech recognition—Alexa, Google Assistant, Google Home, Siri
- Face or image recognition (automatic friend-tagging suggestions)
- Self-driving cars by Tesla
- Recommendations on Netflix, Spotify, YouTube, etc.
- Google Search
- Google Maps (showing traffic conditions)
- Spam filters on emails
- Medical diagnoses and healthcare
- Online fraud detection
- Online shopping with Amazon, eBay, etc.
- Intelligent Gaming—AlphaGo, Deep Blue, etc.
- Social networking—Facebook, Instagram, Twitter, etc.

TYPES OF MACHINE LEARNING SYSTEMS

Machine Learning systems are classified according to how they are trained to learn incrementally, how they generalize, and how data points are compared or built to detect patterns. The output model can be combined with other Machine Learning systems and data components to create the right solution.

1 Supervised Machine Learning:.

Supervised learning, as the name indicates, involves a supervisor's presence as a teacher and a training set that includes the solutions (labels). This is the simplest form of Machine Learning, a method where you have input variables (X) and output variables (Y), which implies Y = f(X). Our end goal is to approximate the mapping function (f) so that we can predict the output variables (Y) when we have new input data (X). This training dataset includes inputs and correct outputs, which allows the model to learn over time. For example, a machine learns to classify whether an image is of a bird or an animal.

2 Unsupervised Machine Learning:

As the name indicates, in the case of unsupervised learning, there is no help from the user for the computer to learn, i.e., no labeled training sets. This allows the model to work on its own to discover patterns and information previously undetected. There are no actual data points in unsupervised learning, and references are drawn from observations in the input data. For example, an ML model could help a user understand different client groups around which to build a business strategy.

3 Semi-Supervised Machine Learning:

Semi-supervised machine learning combines supervised and unsupervised learning by using a few labeled data and plenty of unlabeled data, which helps avoid the challenge of finding large amounts of labeled data. This model is trained to label data.

4 Reinforcement Learning:

In reinforcement learning, the model learns with a rewarding process (positive or negative) to perform tasks in a better way, based on the actions or experiences

Categories of Semi-Supervised Learning Methods:

- Inductive Learning (Inference): In this method, the model learns from a specific dataset and generalizes to make predictions on unseen data.
- Transductive Learning: This method refers to reasoning from specific observed (training) instances to specific observed (unlabeled) instances. An example would be a text document classifier. A semi-supervised learning algorithm can label data and retrain the model with the newly labeled dataset.

Primary Components of Reinforcement Learning:

- Agent (the learning system)
- Environment (agent comes in contact with the environment to select actions)
- Reward or penalty
- Policy (learn the strategy—define the appropriate action in a given situation)
- Iterate (the update policy)

An agent learns from the environment by interacting with it—taking the necessary action to achieve the best result—and learns to create a “policy” that defines future actions.

5 Batch Learning/ Offline Learning:

In batch learning, the ML system is trained using the total data available. The model only works with a limited set of data and does not learn incrementally. When new data becomes available, the system is updated to a new version. Batch learning algorithms are used for small quantities

6 Online Learning/ Incremental Learning:

In online learning, the ML system learns incrementally with sequential instances. This method is best for systems with continuous flow and fast changes in the data.

7 Instance-Based Learning:

In batch learning, the ML system is trained using the total data available. The model only works with a limited set of data and does not learn incrementally. When new data becomes available, the system is updated to a new version. Batch learning algorithms are used for small quantities of data with no incoming data.

8 Model-Based Learning:

In model-based algorithms, a model is created from samples and generalizations in order to make predictions.

TECHNIQUES TO IMPLEMENT MACHINE LEARNING

Linear Regression:

The regression method belongs to the category of supervised ML. Linear regression is the simplest and most common method of learning predictive modeling. It is used to estimate real values (cost of houses, number of calls, total sales, the stock price will increase or decrease, etc.) based on continuous variables. In linear regression, the relationship between the input variables (x) and output variable (y) is expressed as an equation: Y = a + bX. Thus, linear regression aims to determine the values of coefficients and tries to fit data with the best hyperplane that goes through the points.

Classification:

Classification is an essential component for AI applications and is also required for e-commerce applications. This method allows us to make more informed decisions—sort out spam, predict whether a borrower will return a loan, predict whether or not an online customer will buy a product, conduct fraud detection, tag friends in a Facebook image, and so on. These algorithms predict discrete variable labels. Several classification models are logistic regression, decision tree, random forest, gradient-boosted tree, multilayer perceptron, one-vs-rest, and Naive Bayes.

Clustering:

Clustering algorithms are unsupervised learning methods which group the unlabeled dataset. This method develops collections of objects based on similarity and dissimilarity. Clustering is widely used in sales and marketing for customer segmentation and personalized communication. For example, Amazon and Netflix use clustering to provide new recommendations based on a past search. A few common clustering algorithms are k-means clustering, mean-shift, and expectation-maximization.

Decision Tree:

This is a supervised learning algorithm mainly used for classification problems and regression problems. A decision tree asks a question and, based on the answer (Yes/No), it further splits the tree into subtrees. A typical example of a decision tree would be identifying the insurance premium that someone should be charged based on that individual's situation.

HOW TO DEVELOP AND TRAIN ML MODELS

Training a Machine Learning model means finding the parameters that will fit the training data when running the algorithm to make predictions.

Problem Framing:

The first step to developing an ML model is to identify the business case and success criteria. Once these are determined, then a plan for achieving the objectives of the project can be created.

Model Naming:

Select a name for your model. Add a description of the model. Attach appropriate tags to your model. (Tags are designed to make your model searchable.)

Collect and Prepare the Data:

In this step, data from several sources is searched and divided (data splits) for training, test sets, and validation. Data has to be modified, assembled, cleaned, and labeled. Also, all duplicates must be removed and all errors corrected. (It's worth writing functions specifically for this purpose.)

Identify and Extract Data:

In this step, you need to explore and manage the quality and quantity of data. Therefore, understand how the model will work on real-world data, and select and integrate data from several sources. Having good data is vital, as the model will learn from this data.

Data Analysis:

Data analysis is a process that provides a clear understanding of the data and prepares the data to fit the characteristics of the model. Analyzing data helps identify the feature engineering for the ML model. Data preparation is automated for trying out combinations.

Select Your Machine Learning Model:

There are many models from which to choose, depending on the problem. This step includes algorithms of prediction, classification, clustering, deep learning, linear regression, and so on. Experiment with several models from various algorithm categories to find the best performing model. Perform transformations and feature engineering.

Train Your Machine Model:

The goal of training is to answer a question or make a prediction correctly as often as possible. This step involves training the datasets to operate smoothly. Algorithms and techniques are involved in training the machine model, such as training with hyperparameters to find the optimal, for example. Also, the model training code is developed during this step.

Model Performance and Adjustment:

Machine learning models are tested in real-life situations. For testing purposes, data is divided into the training set and the test set. The model is tested using the "test set" on new instances and checked for generalization errors to determine how the model is overfitting or underfitting the data. At this point, the model should be predicting and ready for deployment.

Model Quality Evaluation:

The evaluation step includes selecting the metrics and conducting the actual evaluation. In this step, you evaluate the machine models on the test set, running the pipeline to transform the data, cross-validation, anomaly detection, novelty detection, quality evaluation, etc.

Launch, Monitor, and Maintain:

Now the model must be deployed to the production environment. ML models need to be monitored with humans or machines to check their performance in order to ensure an optimum data pipeline and data quality. This step involves confirming that the system works with current data and is set to trigger alerts if the model needs retraining.

Krasamo's Machine Learning engineering practices strive to systematize and formalize the production of ML models across the client organization.

Contact Us for More Information.

MACHINE LEARNING AND SCALABILITY

In ML applications, scalability is often a primary concern. Businesses need applications that can maintain the same efficiency when the workload grows, updating to new data and producing predictions. For example, predictions in the stock market happen every millisecond. So, scalability requires building an effective data pipeline. These pipelines should be flexible enough to accommodate many data as well as the high processing velocity required by new ML applications.
Therefore, it is essential to make the Machine Learning infrastructure interoperable to incorporate it into the existing and future resources. For this, we need to set up scalable ML applications to increase the systems' overall performance. Scalable ML algorithms are a class of algorithms that can deal with any amount of data without consuming a tremendous number of resources, such as memory. The primary purpose of scalable algorithms is to allow fast computations for massive data sets.

DEVELOPING A SCALABLE MACHINE LEARNING PIPELINE: —AN ML TRAINING PIPELINE

The task: Develop an ML training pipeline that describes ML workflows for each ML model and keeps a repository for each model candidate.
Completing this task will involve a number of steps:

Choosing the proper framework and language:

ML-based applications can use programming languages such as Python, C++, JavaScript, Java, C#, Julia, Shell, R, TypeScript, and Scala. Python is the most recommended programming language for ML applications. The language can be chosen depending on frameworks, such as TensorFlow, PyTorch, SciKit-Learn, MXNet, Gluon, Sonnet, and Keras. All these frameworks have numerous features. A deep learning framework allows building learning models that are production-ready without getting into the underlying algorithms of the details. Choosing the proper framework that will support your preferred programming language is essential.

Selecting a suitable processor:

Selecting the proper hardware plays a critical role in scalability. In many cases, for ML, the best CPU is a GPU (Graphical Processing Unit), as they are comparatively faster—and quicker in distributing computations across GPU servers. A traditional CPU (Central Processing Unit) is not ideal for large-scale machine learning. Beyond CPU and GPU, there are TPUs (Tensor Processing Units), Google's custom-developed application-specific integrated circuits, which are used to accelerate machine learning workloads.

Data collection:

Data collection is the process of gathering and measuring data that needs to be formatted, cleansed, reduced, and rescaled to make it better. Data storage is also essential in order to develop solutions for the business problem at hand.

Model Training:

A significant step for scalability, model training includes exploring and cleaning the data as well as engineering new features. Training the model means learning good values for all the weights and the bias from labeled examples.

Steps include:

- Inputting training data source
- Naming the data attribute that contains the target to be predicted
- Preparing data transformation instructions
- Training parameters to control the learning algorithm
- Writing a script that runs automatically to train the model
- Testing and validation

Parameters in Machine Learning:

- Model parameters—set parameters for the model to fit the training set
- Hyperparameters training (write scripts)
- Model Scoring

The Input Pipeline:

Data is entered into the learning algorithm as a set of inputs/pipelines. At this stage, the data can be divided (data segregation) into subsets and components, transformed, and then fed it into the system. The data set is then added to the pipeline. Data processing components are self-contained and usually run asynchronously.

Optimization:

The final step is to optimize. To achieve that, optimal parameters must be identified. Optimization algorithms check the input parameter to a function that results in the minimum or maximum output. Evaluation of data performance provides estimates on how the model is overfitting or underfitting the training data.

- **Machine Learning overfitting** happens when the model is too complex and doesn't perform correctly with the training data, giving generalization errors during the validation set. The ML team should find a balance between bias and variances.
- **Machine Learning underfitting** is when the model is too simple for the intended dataset and predictions are inaccurate. The model needs additional parameters or, perhaps, more parameters should be added to the features of the algorithm.

Testing Model:

Before the final deployment happens, it is necessary to test whether the input data is in line with the output data and is yielding maximum predictions. These tests—cross-validation, error analysis, and data validation—should be done and monitored multiple times.

Deploying:

Deployment is when the prediction service goes live. Effectively deploying Machine Learning models is an art rather than a science. Think of deploying frequent model versions of the entire ML system. Cloud hosting platforms are ideal for this purpose. Models can be deployed as a web service and used by web applications that use REST API, deployed as an API for prediction, or containerized. Deploying a model on the cloud using Google AI Cloud Platform provides scalability and load balancing with a perfect environment for running TensorFlow models.

Model monitoring

The model monitoring phase ensures active performance monitoring to catch errors in production, detect degradation, and ensure consistency of inference data and metrics with business objectives. Monitoring code checks live performance of the models.

OPERATIONALIZATION OF MACHINE LEARNING MODELS

Operationalization involves building an integrated ML system that is continuously in production.

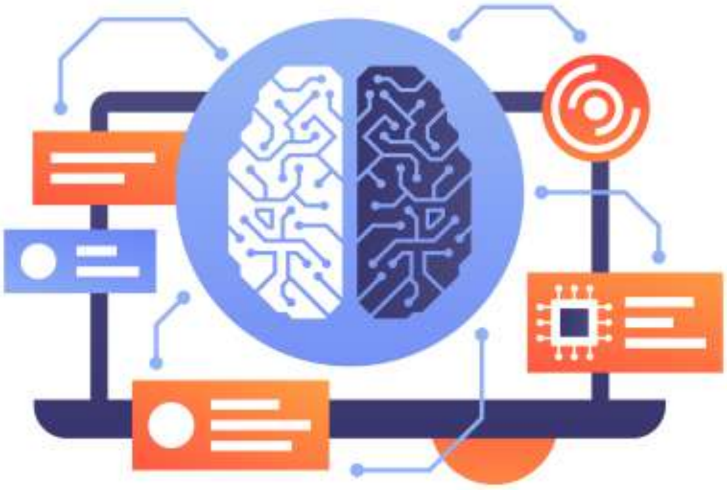
This process refers to the deployment of ML models to be consumed by business applications to predict the target class or target value of a classification/regression problem. In simple terms, operationalization refers to how the designed and developed models will keep running so that those functions or applications that integrate continue to perform with a high level of accuracy and stability.

Operationalization and managing ML models are complex tasks that require maintaining and continuously testing and validating the code, data, and models, managing performance and experiments, and maintaining the accuracy of the algorithms and data to avoid degradation of the models.

A model's success depends on data collection, data engineering, and data science, which means collecting the correct data, understanding the effort needed to extract data from the system, and applying the engineering principles necessary to format, transform, and get the data ready from a data science standpoint. All three lead to the deployment of data.

Challenges with Machine Learning Systems

- Requires continuous testing and validation of data and models
- Needs to integrate external solutions to the ML pipeline
- Model analysis and retraining of candidate ML model
- MLOps culture implementation throughout the machine learning development lifecycle
- Process and metadata management
- Automate the steps of building the ML model
- Data collection and verification
- Retraining models due to degradation or decay (tracking statistics and looking for emerging patterns)
- Tracking experiments to reproduce and reuse during the lifecycle



MACHINE LEARNING DATASETS

A dataset is a synchronized collection of data. It can be in the form of a table, a schema, or an object. This process is, once again, an integral part of the Machine Learning process. In simple terms, the word "datasets" means a collection of data. You can usually find a dataset in a tabular form. Each column denotes a specific variable, and each row signifies a specific member of the dataset.

1 Numerical dataset:

A numerical dataset consists of only the numbers—for example, the height and weight of a person, the total number of pages in a notebook, the number of apples in a grocery store, etc.

2 Correlation dataset:

This type denotes the relationship of variables or attributes between datasets. For example, people that exercise regularly have lower cholesterol levels.

3 Multivariate dataset:

This dataset consists of multiple variables, such as the length, breadth, and height of a rectangular box, for example.

4 Categorical dataset:

This type of dataset consists of the characteristics of a defined object or person, such as an individual's gender and relationship status, for example.

5 Bivariate dataset:

This type consists of two variables, such as students' academic scores and their ages, for example.

Datasets are updated regularly. The most significant benefit of using a dataset is that it helps the user obtain desired data in an organized manner, retrieving the required information quickly from a massive collection of data, thereby saving time and executing tasks more quickly.

MACHINE LEARNING WITH ELASTICSEARCH

Using Elasticsearch in Machine Learning is a helpful method to find data insights, patterns, anomalies, and outliers. This tool enables users to stock, hunt, and inspect significant volumes of data very quickly.

Algorithms are applied to data in order to find metadata, uncovering information and identifying causes that make sense in context when updating ML models. Data is ingested and applied to the ML models.

The benefit of having Elasticsearch is being able to perform and combine many different kinds of complex queries for all types of data and retrieve complex summaries. Data is stored in JSON documents searchable in near real-time with powerful REST APIs.

Advantages of Elasticsearch

- Scalability
- High-speed performance
- Multilingual
- Schema-free
- All data is stored in a JSON format document, a widely accepted web format, and supports auto-completion and instant search.

How to Set Up Elasticsearch on a Windows Platform

- Java must be installed (Version 7 or higher).
- Install the Elasticsearch zip file from the website and unzip the file. Go to the Bin folder and run the Elasticsearch for a batch file.
- An output will occur on the screen, and Elasticsearch is installed.

ML PIPELINE AUTOMATION FOR A PREDICTION SERVICE

Machine Learning systems usually create a multistep pipeline that trains and validates models manually. Still, once your team learns and develops a workable model, the ML pipeline matures and must be automated. Automation speeds up new model training and implementations.

An ML development and operations culture (MLOps) is needed in order to solve the challenges involved in keeping the ML systems in production continuously. DevOps teams evolve to integrate all the elements with the CI/CD environment in order to automate the building, testing, and deployment of ML pipelines.

Continuous training (CT) automation and experimentation in model architecture, feature engineering, and tuning hyperparameters are then added to the operations.

STEPS IN AUTOMATING ML MODEL TRAINING

1. Evaluation—Analyze how to automate the steps of the ML training pipeline to achieve continuous training models from new data for faster iteration and readiness.
2. Exploratory Data Analysis (EDA)—Data analysis is a manual step to understand the data for building the model before making assumptions. (The model analysis is also a manual task.)
3. Build and Test—Try algorithms and models and develop the source code for the ML pipeline steps to automate. Build, test, and package components. Source code is sent to the repository. Many types of testing and verifications are performed on the training models.
4. Modular Components—Components, code, packages, artifacts, and executables are shared for reusability.
5. Automated Data Validation—Ensures that the expected data behaviors, patterns, and expected features comply with data schema. (Watch trigger alerts.)
6. Automated Model Validation—With the trained model, test a dataset to verify the quality of the prediction results, checking the values of the variances. Validation is performed offline, and then online validation is handled with canary deployment.

7. ML Metadata—Metadata is stored when the ML pipeline is executed; the metadata is then used to compare versions during model validation. This process also helps in evaluating metrics, debugging errors, and finding anomalies.
8. Triggers—Monitoring code is written to check the ML system's performance and trigger response alerts when detecting changes in data that feed the training model.
9. Feature Store—A feature store is a repository of features for training and serving that allows the reuse of feature sets with metadata that can be fetched automatically in a batch to the prediction service.
10. Verify the Integration and Deploy—Verify that configurations are correct for integrating with the target environment (APIs, REST API, etc.) and deploy artifacts. Before deploying, check resources and IT infrastructure.
11. Schedule Automatic Execution—The trained model is pushed to the registry.
12. Prediction Service Model—The model is deployed and working with live data.
13. Continuous Monitoring – Apply continuous testing and verifications to validate and ensure performance.

With **Krasamo**, the process of automating Machine Learning pipelines involves the gradual transition from manual to semi-automated to fully automated.

Navigating the timeline for testing and deploying implementations includes a number of different skills and processes. Our expert teams are ready to provide a consultation for your business today.

Contact Krasamo Sales

INFORMATIONAL RESOURCES:

- **TensorFlow Extended (TFX):** An end-to-end platform for creating and managing an ML pipeline

https://www.tensorflow.org/tfx

- **TFX: Production ML with TensorFlow in 2020 (TF Dev Summit '20)**

https://youtu.be/I3MjuFGmJrg

- **Deploy ML Pipelines with Kubernetes Using Kubeflow**

https://www.kubeflow.org/docs/about/kubeflow/

- **Check a Pipeline Code Sample**

https://github.com/kubeflow/pipelines/tree/master/samples/core/xgboost_training_cm

KRASAMO'S MANAGED SERVICES FOR MACHINE LEARNING

You can build and deploy ML models and manage your Machine Learning operations (MLOps) with a collaborative partnership with Krasamo. Our expert teams will help implement your ML operations through third-party services such as AWS or Google.

This modality will simplify and help manage your workflows, thereby avoiding infrastructure management tasks. Building and managing ML pipelines requires significant effort, so partnering with Krasamo for managed services will help your business boost productivity.

MACHINE LEARNING MANAGED SERVICES SOLUTIONS WITH KRASAMO

- Build ML pipeline using TensorFlow
- Evaluate and monitor model performance
- Metadata management (metadata tracking)
- Track artifacts and lineage
- Collaboration capabilities
- Feature engineering
- Track data and performance (notifications)

AMAZON SAGEMAKER

AZURE MACHINE LEARNING

GOOGLE CLOUD AUTOML

MACHINE LEARNING IN PYTHON

Python is particularly suitable for building Machine Learning models and applications. Intuitive and easy to read, it is supported by an extensive collection of ML libraries and frameworks.

- **Get Started Learning Python**

https://www.learnpython.org/

- **The Python Tutorial**

https://docs.python.org/3/tutorial/index.html

- **Python Libraries (NumPy, Pandas, Matplotlib, etc.)**

https://www.scipy.org/

PYTHON FRAMEWORKS

Production-ready Python frameworks include:

- **TensorFlow**
- **Scikit-Learn**
- **Keras**

- **Get Started on Heroku with Python**

https://devcenter.heroku.com/articles/getting-started-with-python

- **Google Machine Learning with TensorFlow APIs Crash Course**

https://developers.google.com/machine-learning/crash-course/

CONCLUSION

Machine learning has become an integral part of business operations in the digital age, together with large amounts of data and computing power.

Have you been wondering how to power your products and features through Machine Learning? Or how to transform your business in disruptive environments?

Machine Learning is particularly suitable for products that require solving complex problems and typically demand high human involvement for fine-tuning and analyzing large amounts of data.

Machine Learning can help product managers improve products and product offerings by mining data from ML algorithms to find new patterns from predictions that were previously unknown or difficult to see.

A successful Machine Learning strategy emphasizes business issues to solve, builds a business case, and puts users at the center while applying the relevant technical aspects to the project.

A Krasamo business analyst can run an ROI analysis and cost analysis (fixed vs. variable) for you with projections of Machine Learning models and simulated algorithm running.

Want to add **Machine Learning** to your data?
Or discover which Machine Learning algorithm to use?
Or perform an **ML model training** simulation?

Krasamo has a team with expertise in Machine Learning models ready to meet your requirements.
Contact Krasamo Sales Today

LEARN MORE ABOUT:

- **Machine Learning**

https://www.krasamo.com/machine-learning/

- **ETL Data Strategy**

https://www.krasamo.com/etl-data-strategy/

- **Digital Strategy**

https://www.krasamo.com/digital-strategy-101/