

Usage of Fuzzy Logic and Machine Learning in Web Development

Zlatko Čović

Subotica Tech – College of Applied Sciences, Subotica, Serbia

chole@vts.su.ac.rs

Abstract — This paper presents the possibility of using fuzzy logic and machine learning in web development with PHP programming language. Fuzzy logic and machine learning found their applications in various fields of web development. The Python programming language is one of the most popular for developing applications that utilize fuzzy logic and machine learning. When it comes to integrated web systems using some form of AI, it's common to see the AI module implemented using the Python programming language, while the rest of the system is written using other programming languages and technologies. API endpoints are created for data exchange among different languages within such systems. PHP, as a server-side programming language, is used on over 75% of websites, according to the W3Techs report. The work briefly describes the most popular libraries in the PHP programming language that are used for applying fuzzy logic and machine learning. Implementation requirements and basic features are listed.

Keywords: fuzzy logic, machine learning, web development, php

1 INTRODUCTION

Fuzzy logic is a mathematical framework designed to address uncertainty and imprecision in decision-making and reasoning. It enables the representation of levels of truth, as opposed to rigid true or false values. Fuzzy logic employs fuzzy sets, fuzzy rules, and fuzzy inference to manage imprecise or ambiguous information and arrive at approximate decisions.

It is applied in diverse fields such as control systems, decision support systems, expert systems, natural language processing, and machine learning. Fuzzy logic empowers systems to process uncertain and subjective information, offering greater flexibility and nuanced reasoning abilities.

Fuzzy logic can be employed to enable adaptability in web interface design. In [1], authors presented development of web portal to support online shopping, with such characteristic. The web portal stored past shopping records and maintained the actual shopping list. Portal used collected data and a fuzzy approach to create adapted functionality in web interface. According to the results, it can be said that usage of fuzzy methodology for adaptation into Web interfaces can feasibly be implemented.

In various loyalty programs, employers create mobile applications and/or websites that utilize some form of machine learning and fuzzy logic. Based on information obtained from consumer habits (what they buy, when they buy, where they buy, how much they buy, etc.), algorithms are triggered to predict a personalized offer for the consumer. Based on reactions to the generated offer, the algorithm learns and continues to improve.

Accurate effort estimation is crucial in software development, as it directly influences project timelines and success. The study [2] introduces a model for estimating effort in web-based mobile app development using an object-oriented approach. The proposed model combines functional size measurement with object-oriented web app metrics and mobile characteristics. Estimations are performed using the Mamdani fuzzy logic method.

Artificial intelligence (AI) pertains to the development and deployment of intelligent systems capable of executing tasks that usually necessitate human intelligence. It encompasses the emulation of human-like intelligence in machines to equip them with the capacity to perceive, reason, learn, and make decisions.

Machine learning is a component of artificial intelligence that concentrates on creating algorithms and models enabling computers to learn from data and make predictions or decisions without explicit programming. This entails scrutinizing extensive datasets to discern patterns, extract meaningful insights, and construct predictive models.

The popularity of the Python programming language arises from its utilization in developing applications incorporating fuzzy logic and machine learning. This popularity is attributed to its rapid execution speed, relatively uncomplicated programming syntax, and a multitude of pre-existing libraries equipped to execute various tasks linked to artificial intelligence.

In scenarios that involve integrated web systems using some form of artificial intelligence, it is common to utilize an AI module that is created using the Python programming language. The rest of the system is typically implemented using other programming languages and technologies.

Within such systems, API endpoints are generated to facilitate data exchange across diverse languages. Data exchange is conducted in one of the accepted standard formats, but the JSON format is the most popular due to its processing speed and the fact that all programming languages have good functions and libraries for working with this data format.

In these systems, sometimes ML modules are created as standalone components with the option to import and export data into a text file without direct communication with the rest of the system.

As per the W3Techs report [3], PHP, functioning as a server-side programming language, is used on more than 75% of websites. The paper briefly outlines the most prominent libraries within the PHP programming language, utilized for the application of fuzzy logic and machine learning.

1.1 Application of Fuzzy logic in web development

In the realm of web development, various applications of fuzzy logic can be observed:

Enhanced Search Functionality: Fuzzy logic contributes to refining search capabilities on websites by yielding more precise and pertinent outcomes. This involves accommodating approximate matches and managing partial correspondences, empowering users to locate desired information even in the absence of exact search terms.

Efficient User Input Handling: Fuzzy logic proves valuable in adeptly managing user input within web forms. Its capacity to decipher and comprehend ambiguous or vaguely articulated inputs mitigates errors and elevates the overall user experience.

Facilitating Decision-making: Fuzzy logic plays a pivotal role in aiding decision-making processes within websites. Its ability to embody uncertainty and imprecision within decision models is particularly beneficial for recommendation systems, personalized content delivery, and adaptive user interfaces.

1.2 Application of machine learning in web development

Personalization and Recommendation Systems: Machine learning analyzes user data for personalized website recommendations, often seen in e-commerce, streaming, and social media platforms. Decision Trees and Random Forests algorithms are versatile and can be used for both classification and regression tasks.

Natural Language Processing (NLP): ML, especially NLP, creates chatbots, virtual assistants, and voice interfaces for websites, enhancing user interaction.

Fraud Detection and Security: ML detects website fraud patterns (e.g., payment fraud, fake accounts) to enhance web app security.

Predictive Analytics: ML algorithms analyze website data for predictions and insights, aiding in user behavior forecasts, campaign optimization, and data-driven decisions. Linear Regression is an algorithm that can be suitable for these types of tasks.

Text classification tasks: Naive Bayes algorithm could be used for tasks like sentiment analysis in social media or customer support interactions.

Image classification (recognizing objects in images): Support Vector Machines (SVMs) excel in classification tasks, particularly in situations where there exists a distinct and well-defined separation between classes.

1.3 Application of AI in web development

Artificial intelligence (AI) is seamlessly integrated into the web development industry, handling increasingly intricate tasks. Most popular 7 ways AI is used in a web development are:

- Coding Assistants
- Artificial Design Intelligence (ADI)
- Smart Chatbots for a Personalized End-User Experience
- Voice-Based Interactions for a Convenient Search Experience

- Search Engine Optimization (SEO) Made Easier
- Dynamic Content Personalization
- Website and Vulnerability Testing [4]

1.3.1 Coding Assistants

Tools such as Codex and Github's Copilot aid programmers by proposing the subsequent lines of code they're actively engaged with, or even compiling small code segments for integration. Nevertheless, these tools are limited to managing concise and uncomplicated programs, and their precision rates are modest [4].

1.3.2 Artificial Design Intelligence (ADI)

Artificial Design Intelligence (ADI) can produce complete e-commerce websites tailored to users, covering a range of elements including visual design, layout, and content. Through a series of straightforward inquiries and drawing upon its extensive training data, ADI can create websites that match any objective and accommodate a variety of tastes. One of the examples is Wix ADI, technology that has the capability to create individual websites [4].

1.3.3 Smart Chatbots for a Personalized End-User Experience

An AI chatbot enables users to quickly access information through natural language queries, leveraging website content and customizable responses. Intercom offers AI-driven chatbots that seamlessly integrate into websites, apps, and other software solutions for companies [4].

1.3.4 Voice-Based Interactions for a Convenient Search Experience

Voice search offers a more efficient way to access information compared to manually browsing through written content. Users can verbally pose their question, and an AI system optimized for voice recognition retrieves and delivers the relevant information audibly. Apple's virtual assistant, Siri, can retrieve information and perform various tasks through voice commands [4].

1.3.5 Search Engine Optimization (SEO) Made Easier

An AI-driven SEO tool can manage these tasks on your behalf. It can generate effective headlines and topic suggestions for your writing, along with supplying the essential keywords to enhance your content's ranking.

SEO.ai is software for swift and effective search engine optimization content creation [4].

1.3.6 Dynamic Content Personalization

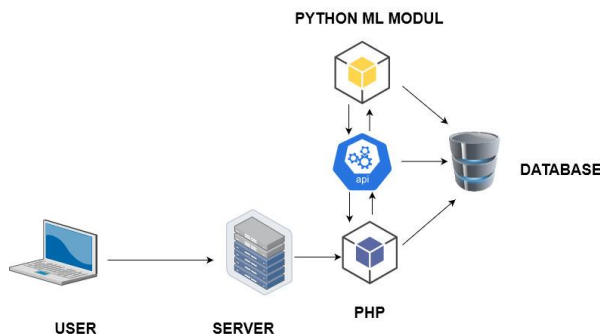
AI-powered dynamic content personalization enables custom experiences for large customer bases, leveraging diverse data sources to understand behaviors and craft tailored marketing strategies. Braze is a customer engagement platform that helps brands maintain audience engagement through customer-centric interactions [4].

1.3.7 Website and Vulnerability Testing

Conventional website code testing requires hands-on effort, with programmers closely overseeing outcomes, refining tests, and monitoring changes. AI can automate simpler tests, allowing programmers to focus on complex issues, streamlining testing and improving efficiency. Diffblue Cover uses AI to automate Java unit test generation, aiding programmers in Java application creation, updating, and maintenance [4].

2 WEB SYSTEM WITH ML MODULE

Integrated web systems often incorporate a Python-based ML/AI module, while other languages and technologies handle the rest of the system. API endpoints facilitate data exchange across languages, commonly using JSON due to its speed and universal support. In some cases, ML modules function independently, allowing data import and export via text files without direct system communication. The figure (1. figure) shows the most common architecture of integrated web system with ML module. In many systems, the AI/ML module is physically located on a separate server.



1. figure: Example of architecture of integrated web system with ML module

PHP is the most popular server-side programming language, and it is used on over 75% of websites. For using PHP instead of Python in ML modules, it is necessary to be familiar with existing classes and libraries.

3 PHP CLASSES AND LIBRARIES

PHP has several classes and libraries that are used for performing fuzzy logic and machine learning operations. One of the more popular built-in PHP functions is *levenshein()*, which calculates Levenshtein distance between two strings.

3.1 Levenshtein

The *Levenshtein* Distance (LD) is a fuzzy matching technique used to measure the difference between two strings. The resulting number indicates the degree of dissimilarity between the two strings. A higher *Levenshtein* edit distance signifies a greater disparity between the two terms.

```
levenshtein(
    string $string1,
    string $string2,
    int $insertion_cost = 1,
```

```
    int $replacement_cost = 1,
    int $deletion_cost = 1
): int
```

The Levenshtein distance is defined as the minimal number of characters you must replace, insert or delete to transform string1 into string2 [8].

If insertion_cost, replacement_cost and/or deletion_cost are unequal to 1, the algorithm adapts to choose the cheapest transforms. E.g. if \$insertion_cost + \$deletion_cost < \$replacement_cost, no replacements will be done, but rather inserts and deletions instead [8].

As an illustration, when measuring the distance between "Cristian" and "Christian," the distance is 1, as they differ by only one letter, "h." This term is commonly used interchangeably with "edit distance".

3.2 FuzzyDM library

Fuzzy logic decision making library is one of the PHP's libraries created for the Fuzzy logic. Library can be installed with composer via *Packagist* web site (<https://packagist.org/packages/ketili/fuzzydm>). It requires 5.3.0 or above version of PHP.

FuzzyDM is a small PHP library to make multi criteria decisions. It relies on fuzzy-logic fundamentals and employs simple membership and aggregate functions. With this library, decisions can be made based on some predefined numeric features.

3.3 PHP AI libraries

Integrating AI into PHP programming is achievable through various libraries and frameworks. Notable ones within the PHP ecosystem, like PHP-ML and Rubix ML, enable this incorporation. They provide a wide range of AI algorithms, including neural networks, decision trees, and clustering, allowing for the creation of intelligent applications.

Additionally, PHP frameworks like Laravel and Symfony come with specialized tools and extensions that facilitate the use of AI libraries and seamless integration of AI capabilities into web applications.

3.4 PHP-ML

PHP-ML is a PHP library dedicated to machine learning. It offers a diverse array of algorithms and tools for tasks like classification, regression, clustering, dimensionality reduction, and beyond. The design prioritizes user-friendliness and seamless integration with existing PHP applications. PHP-ML is released under the MIT Licence. It requires PHP version 7.1 or above. Complete documentation of usage of this library and sample examples for every feature can be found at <https://php-ml.readthedocs.io/en/latest/>.

Here are some advantages of PHP-ML [5]:

- Seamless Integration: PHP-ML effortlessly integrates with various PHP frameworks and libraries.
- User-Friendly Interface: The library boasts a user-friendly interface, facilitating an easy entry point for developers venturing into machine learning.

- **Thriving Community:** PHP-ML benefits from a robust developer community, actively contributing to its advancement and offering support.
- **Diverse Algorithmic Support:** PHP-ML accommodates an extensive array of machine learning algorithms, encompassing classification, regression, clustering, and more.
- **Swift Execution:** PHP-ML is engineered for efficiency, ensuring rapid execution of machine learning algorithms.
- **Open-Source:** PHP-ML is open source, affording developers the freedom to utilize and customize it according to their specific requirements.

PHP-ML offers a lot of features. Here are some of them [5]:

- **Modular Approach:** PHP-ML presents a collection of distinct components that stand alone effectively or meld seamlessly to construct a comprehensive machine-learning system.
- **Easy Usage:** PHP-ML boasts a user-friendly interface, ensuring accessibility for both newcomers and seasoned machine learning experts.
- **Extensive Algorithmic Diversity:** PHP-ML encompasses a vast spectrum of machine learning algorithms, including classification, regression, dimensionality reduction and clustering.
- **Open-Source Advantage:** PHP-ML operates as an open-source solution, granting the freedom to utilize and adapt as per individual requirements.
- **Streamlined Data Preprocessing:** PHP-ML streamlines data preprocessing capabilities, encompassing functions like normalization, standardization, and feature scaling.
- **Robust Cross-Validation:** PHP-ML facilitates robust cross-validation capabilities, a pivotal tool for appraising the efficacy of machine learning models [5].
- **Integrated Visualization:** PHP-ML seamlessly integrates built-in visualization features catering to data and machine learning models.
- **Versatility Across Formats:** PHP-ML offers support for a multitude of data formats, encompassing CSV, TXT, and JSON, simplifying compatibility with diverse data sources.

Several general procedures need to be involved in utilizing PHP-ML. First it is necessary to install the library using Composer, which will add the library to your project dependencies.

As it works with data, it needs to be loaded into an array or a dataset object. The data can be read from a database, a file, or any other source.

After loading, preprocess data as needed for machine learning, including normalization, outlier removal, and converting categorical variables to numerical. Data division is necessary in training and testing sets. The training set instructs the model, and the testing set assesses the effectiveness of the model.

The next step is to choose a machine learning algorithm that is proper for a given problem. Model can be trained by

providing the training data and the selected algorithm to the relevant class in PHP-ML.

The testing data can be passed to the trained model for the assessment of its performance. Once your model has been trained and tested, it can be employed for predictions on new data. The trained model can be saved to a file for future reuse. Deployment to a web application or any other software system where it can be utilized for predictions is also possible. Below a simple example of classification is shown [5].

```
require_once __DIR__ . '/vendor/autoload.php';

use Phpml\Classification\KNearestNeighbors;

$samples = [[1, 3], [1, 4], [2, 4], [3, 1],
[4, 1], [4, 2]];
$labels = ['a', 'a', 'a', 'b', 'b', 'b'];

$classifier = new KNearestNeighbors();
$classifier->train($samples, $labels);

$classifier->predict([3, 2]);
// return 'b'
```

Example scripts are available in a separate GitHub repository <https://github.com/php-ai/php-ml-examples>. One of the scripts is *Code Review Estimator*, which is a simple showcase of machine learning for code review cost estimation.

Many of the examples have been used in articles written by the author of this library, for example: Prediction of air pollution with k-Nearest Neighbors. The k-Nearest Neighbors algorithm is a straightforward method for classification (supervised learning). Its operational principle involves identifying the k closest neighbors to the target value, usually determined using a distance metric [5].

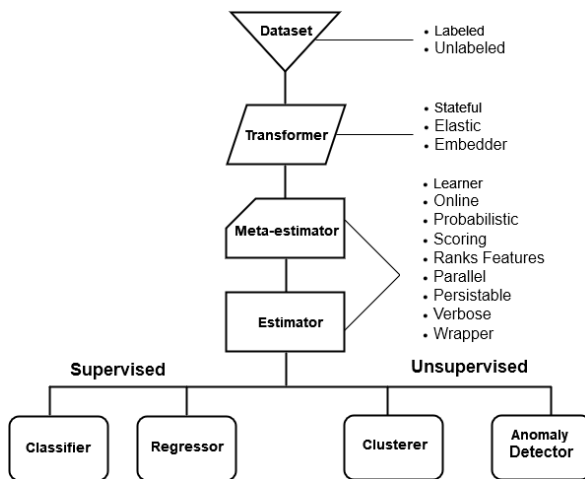
3.5 Rubix ML

Rubix ML is a free open-source machine learning library that allows you to build programs that learn from your data using the PHP language. It is the most popular ML library for PHP. It provides tools for the entire machine learning life cycle from ETL to training, cross-validation, and production with over 40 supervised and unsupervised learning algorithms. Rubix ML strives to enhance the accessibility of machine learning for PHP developers, offering a user-friendly platform to construct, train, and assess machine learning models [6].

It can be installed using *Composer*. Library requires PHP version 7.4 or above. *Tensor extension* for fast Matrix/Vector computing is recommended. It is a library and extension that provides objects for scientific computing in PHP. All Rubix ML projects are constructed to run from the PHP command line interface (CLI). The reason almost always boils down to performance and memory consumption [6].

Optional extensions for working with the Rubix ML library can also be installed: GD extension for image support, Mbstring extension for fast multibyte string manipulation, SVM extension for Support Vector Machine engine (libsvm), PDO extension for relational database support and GraphViz for graph visualization [6].

A flexible modular architecture characterized by several fundamental abstractions and their corresponding types and interfaces is employed by Rubix ML [6]. The figure (2. figure) shows the Rubix ML system architecture.



2. figure: Rubix ML system architecture

Some benefits of using Rubix ML include [5]:

- **User-Friendly Design:** Rubix ML is crafted for ease of use, ensuring a straightforward learning curve. Its intuitive and uniform interface empowers developers to effortlessly build machine-learning models.
- **Versatility:** Rubix ML accommodates an extensive array of machine learning algorithms, granting developers the freedom to select the one that aligns most closely with their requirements. Additionally, it permits users to tailor their models and algorithms to suit specific preferences.
- **Efficiency:** Rubix ML is engineered for high performance and scalability, enabling it to process sizable datasets and intricate models with optimal effectiveness.
- **Data and Model Visualization:** Rubix ML offers resources for visually representing data and models. This aids developers in gaining a deeper comprehension of their data and enhancing their models.
- **Open source:** Rubix ML is freely available for use, and developers can actively contribute to its ongoing development.

In addition to the mentioned advantages, Rubix ML has features that make it a good choice for use in various projects. Some of them are:

- Accommodates a diverse array of machine learning algorithms.
- Furnishes a user-friendly and uniform interface.
- Enables training on various devices (e.g., CPU, GPU).
- Includes tools for feature selection and data preprocessing.
- Facilitates model persistence and serialization.

- Allows for hyperparameter tuning and model selection.
- Provides support for ensemble learning and pipelines.
- Comes with extensive documentation and a substantial collection of examples.
- Permits the development of custom models.
- Applicable in both research and production settings [5][6].

Example projects can be found in Tutorial page [7]. Many of them come with usage instructions and a pre-cleaned dataset. The most numerous examples are related to prediction, recognition, and analysis.

For example projects to be more easily understood, it is recommended to master the basic steps in creating project structure and applying this library. Additionally, you should have a fundamental understanding of various types of machine learning such as classification and regression [7].

Below, the steps in the basic example will be presented.

3.5.1 Obtaining data

Projects of machine learning usually begins with a question. Define the questions and collect the answers for them. For examples, data should be structured and represented in a 2-d array (or matrix), and if labels are used, they should be presented as a 1-d array.

```
$samples = [
    [3, 4, 40.2],
    [2, 2, 17.5],
    [2, 3, 14.7],
    [3, 4, 52.0],
];

$labels = ['graduated', 'ungraduated',
'graduated', 'ungraduated'];
```

3.5.2 Construction of Labeled or Unlabeled dataset

In Rubix ML, datasets are conveyed using specialized containers known as Dataset objects. Dataset objects manage the selection, subsampling, splitting, randomization, and sorting of both samples and labels they encompass. There are two categories of datasets: Labeled and Unlabeled.

```
use Rubix\ML\Datasets\Labeled;
$dataset = new Labeled($samples, $labels);
```

3.5.3 Estimator selection

The core of the Rubix ML library is composed of estimators. They furnish the *predict()* API and are tasked with generating predictions for unfamiliar samples. Estimators that can be trained with data are referred to as Learners and necessitate training prior to making predictions. Example shows the usage of K Nearest Neighbors classifier.

```
use Rubix\ML\Classifiers\KNearestNeighbors;
$estimator = new KNearestNeighbors(3);
```

3.5.4 Training

The process of feeding the learning algorithm data so that it can build an internal representation (or model) of the task it is trying to learn is known as training. This representation is comprised of all the parameters (except hyper-parameters) that are needed to make a prediction.

To initiate training, provide the training dataset as an argument to the *train()* method on the learner instance.

```
$estimator->train($dataset);
```

3.5.5 Making Predictions

The *predict()* method can be used with unlabeled dataset on newly trained estimator. The process of generating predictions is referred to as inference, as the estimator utilizes the model created during training to deduce the label of unknown samples.

```
use Rubix\ML\Datasets\Unlabeled;
```

```
$samples = [
    [3, 4, 40.2],
    [2, 2, 17.5],
    [2, 3, 14.7],
    [3, 4, 52.0],
];
```

```
$dataset = new Unlabeled($samples);
$predictions = $estimator->predict($dataset);
```

The estimator produces the predicted class labels for the unknown samples.

3.5.6 Model Evaluation

By setting some data aside, the model can be tested on data it has never encountered before. This technique, known as cross validation, aims to evaluate an estimator's capacity to generalize from its training.

```
use Rubix\ML\CrossValidation\HoldOut;
$validator = new HoldOut(0.2);
```

The *HoldOut* validator allocates a portion of the dataset for testing and reserves the remainder for training. The user needs to specify the ratio of testing to training samples as a constructor parameter for the *Hold Out* validator. For example, factor of 0.2 (20%) of the dataset for testing is chosen.

```
use Rubix\ML\CrossValidation\HoldOut;
$validator = new HoldOut(0.2);
```

For the *test()* method on the validator, it is necessary to select a compatible validation Metric to serve as the scoring function. In this example, Accuracy metric is used.

```
use Rubix\ML\Datasets\Labeled;
use Rubix\ML\CrossValidation\Metrics\Accuracy;
```

```
$dataset = new Labeled($samples, $labels);
```

```
$score = $validator->test($estimator, $dataset,
    new Accuracy());
```

4 CONCLUSIONS

Fuzzy logic and machine learning are potent technologies applicable across diverse fields, including web development.

The web development industry is already fully integrated with artificial intelligence, which is now handling increasingly complex tasks.

Python is highly favored for developing applications with fuzzy logic and machine learning. In integrated web systems employing AI, the AI module is typically written in Python, while the rest of the system uses different languages and technologies. API endpoints facilitate data exchange between languages in these systems. In the web environment, PHP can be a valuable alternative to the Python programming language, boasting strong support for AI through popular libraries.

The paper presented PHP classes and libraries for executing fuzzy logic and machine learning operations. Description of features and short examples of usage are explained.

REFERENCES

- [1] H. O. Nyongesa, Tian Shicheng, S. Maleki-Dizaji, S. . -T. Huang and J. Siddiqi, "Adaptive Web interface design using fuzzy logic," Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003), Halifax, NS, Canada, 2003, pp. 671-674, doi: 10.1109/WI.2003.1241293.
- [2] Agusta, S., Suharjo, Girsang, A.S. (2018). Effort Estimation Development Model for Web-based Mobile Application Using Fuzzy Logic, *Telkonnika*, vol. 16, no.5, pp.2082-2090, doi: 10.12928/TELKOMNIKA.v16i5.6561.
- [3] Web Technology Surveys, <https://w3techs.com>
- [4] Springboard, Monica J. W. (2023). 7 ways AI Is Being Used in Web Development, <https://www.springboard.com/blog/data-science/ai-web-development/>
- [5] Haider, H. (2023). Artificial Intelligence in PHP programming, <https://www.linkedin.com/pulse/artificial-intelligence-php-programming-haris-haider>
- [6] Rubix official page, <https://rubixml.com/>
- [7] Rubix ML example projects, <https://docs.rubixml.com/2.0/index.html#tutorials-example-projects>
- [8] PHP Levenshtein function, <https://www.php.net/manual/en/function levenshtein.php>