ISSN: 2754-6659

Journal of Artificial Intelligence & Cloud Computing

Review Article

Open Access

Machine Learning for Software Developers: An Introductory Guide

Vamsi Thatikonda

8921 Satterlee Ave Se, Snoqualmie, WA, USA

ABSTRACT
The paper acts as a guide into the fundamental principles of Machine Learning, and its significance for software developers. It offers insight into types of machine learning, important concepts, and relevant practical applications of the concepts to provide a deeper understanding of the context. Tools and frameworks have also been presented, which are necessary to implement a machine learning algorithm within the organization. The paper however, emphasizes over the importance of data preparation, model building and training for successful implementation. Additional resources and practical examples that might be useful towards learning and application have also been provided to make it a seamless process of information exchange. It also addresses common challenges, ethical concerns and latest trends emerging within the field, to ensure that a complete overview has been presented to software developers to prepare for the future effectively.

***Corresponding author**
Vamsi Thatikonda, 8921 Satterlee Ave Se, Snoqualmie, WA, USA.

Received: September 08, 2022; **Accepted:** October 16, 2022; **Published:** November 15, 2022

Keywords: Machine Learning, Fundamentals, Data Preprocessing, Model Selection, Hyperparameter Tuning, Bias and Fairness

Introduction

The concept of machine learning has only recently been recognized as a transformative force within technology since it has significantly changed the economy [1]. Software developers are presented with an innovative tool through which they can create intelligent and data-driven applications, that have the ability to learn and evolve with time. These applications also become responsive and predictive based on its intelligent technology. Developers have been mainly found to improve the user experience through using data to make significant decisions shaping their applications, while also automating their tasks and identifying new opportunities to capitalized from the market. The fundamentals of machine learning therefore, become an important asset for software developers to learn [2].

This paper is also aimed at providing software developers with an overview of the concept of machine learning, introducing them to this rather complex field in a seamless manner. The paper provides useful knowledge, tools and resources that can initiate the step of a software developer towards integrating machine learning in their tasks. Despite providing a broad view of the field, it is only an introductory guide and topics that are relevant yet advanced, have not been discussed in detail. Moreover, as machine learning itself is rapidly evolving, there might be new tools and frameworks in the market over time that surpass the effectiveness of the tools and frameworks provided. However, rest assured, software developers can use this paper as a foundation towards embarking their venture within the field of machine learning.

Machine Learning Fundamentals

Machine Learning is a part of a broader concept; Artificial Intelligence (AI). It gives computer systems the ability to learn from data and subsequently use the insights to improve their performance automatically without any external interference within its programming. It hence, teaches machines to identify and generalize patterns or trends occurring within data gathered/provided, and make decisions and predictions based on it. The process itself includes developing algorithms that helps the computer adapt and evolve as new information is obtained, making it a powerful tool in context of; data analytics, pattern recognition and predictive modeling [3].

There are three categories of machine learning; unsupervised learning, supervised learning and reinforcement learning. Unsupervised learning is an algorithm that operates on unlabeled data, targeted at identifying patterns or structures within the data provided. Whereas, the supervised learning is based on a labeled dataset, which is trained to learn to map input data according to the corresponding output. The last category, reinforcement learning, trains algorithms to make sequences of decisions through the trial and error method, within the context of an environment [4].

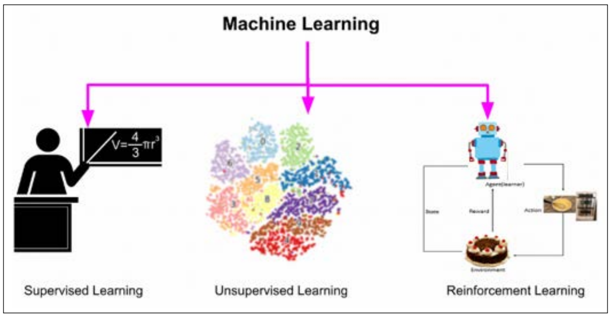

Figure 1: Types of Machine Learning

There are several key concepts within the field of Machine Learning. Data is the input provided for machine learning; features are the characteristics associated with the data whereas labels is the desired outcome of the algorithm. The data is split into a training set and a testing set, where the training set teaches the model and the testing model evaluates its performance [5]. Machine learning algorithms such as decisions trees, neural networks and support vector machines help make models that can make predictions [6].

Machine learning has its use cases in various industries of the economy, ranging from recommendation systems in e-commerce, fraud detection in finance, to image and speech recognition in healthcare and autonomous vehicles. It is also used to conduct a sentiment analysis on social media, forecast demand for retail and predict maintenance requirements in manufacturing, displaying that it is versatile across all industries [7].

Tools and Frameworks

Machine Learning Libraries- Machine learning libraries provide a collection of pre-built algorithms and functions that can assist individuals in developing and implementing a machine learning model with ease. They also offer different tools including; data manipulation, model training and evaluation which reduces the need to develop an algorithm from scratch [8].

Programming Languages- Among the two most popular programming languages are Python and R. Python is a simplistic language, having an extensive library and a thriving community, because of which it is highly favored. In comparison, R is favored for its statistical capabilities because of which it aids data analysis and machine learning [9].

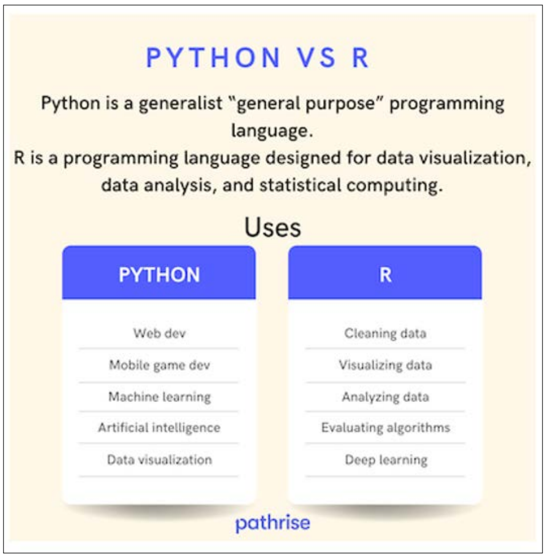

Figure 2: Python vs R

TensorFlow and PyTorch

These names belong to the leading machine learning frameworks prevalent within the industry. TensorFlow has been developed by google, is scalable and can be applied to a wide range of use cases. PyTorch meanwhile has been developed by Facebook, offering a dynamic computational graph and is highly favored for research and prototyping [10].

Additional Tools

There are additional tools and frameworks that are useful for machine learning. Scikit-learn is a classical machine learning algorithm, Keras aids building neural networks and XGBoost is used for gradient boosting. These tools help cater different needs and preferences within the field [11].

Data Preparation for Machine Learning

Data preparation is an important step when implementing machine learning since it affects the performance and effectiveness of the model.

Data can initially be collected from a variety of sources including; databases, APIs, and sensors etc [12]. The data can be structured such as in the form of tables or unstructured data such as in the form of text or images. It is also important to consider the quality and relevance of the data because of which data sources need to be carefully selected and extracted [13].

Data preprocessing includes a series of tasks which can clean and prepare the data, to analyze it effectively. This can help manage handling missing values, removing duplicates and addressing any outliers. Data can also be standardized or normalized to ensure that all features are on the same scale. Data is then encoded to convert the categorical variables into numbers for the machine learning algorithm. To avoid issues in performance, it is important to conduct proper preprocessing [14].

Feature engineering is a process that can also be used to create new or modify existing features within the model to improve its capability of identifying patterns within data. It can include feature selection which selects relevant attributes, can create interactions terms while also transform variables. Feature engineering can help build an efficient model [15].

When dealing with imbalanced data, there are many techniques that can be used including; oversampling the minority, under sampling the majority or simply using synthetic data generation methods such as SMOTE (Synthetic Minority Over-sampling Technique). Such instances may occur due to one of the groups or classes outnumbering the other in terms of classification. It can include biasness within the model, performing poorly in the minority class. In order to avoid such an imbalance, data needs to be handled carefully [16].

Data for Machine Learning

Data is the most crucial ingredient for building machine learning models. The quality and relevance of data determines the success of any ML application.

Data Sources

Some common data sources include [17]:

- Databases - Relational, NoSQL, graph databases etc. containing enterprise data.
- Sensors - IoT devices generating real-time telemetry data.
- Web - Scraped data, APIs providing access to web data.
- Transactions - User behavioral data from mobile, web apps.
- Enterprise Content - Documents, emails, reports within organization.
- Public Datasets - Repositories like Kaggle, government open data.

Data Modalities

ML can be applied to different data modalities [18]:

- Structured Data - Tabular data in database rows and columns.
  Text - Documents, emails, social media posts etc. requiring NLP.

- Images - Image classifiers using CNN models.
- Audio - Speech recognition, sentiment analysis from audio data.
- Time-Series - Forecasting models for temporal data.

Big Data Technologies

For large datasets that exceed memory/compute limits, distributed big data tools like Apache Spark can be leveraged on clusters [19]. Spark provides distributed data processing on RDDs resilient to failures. Spark MLlib contains distributed ML algorithms for big data.

Model Building and Training

For any machine learning model, it is important to identify a suitable algorithm. The algorithm can be selected based on the problem and data characteristics. For instance, in context of classification tasks, decision trees, random forests or support vector machines can be used whereas in context of regression tasks, linear regression or ensemble methods can be used. Deep learning can also be used with neural networks when addressing complex tasks such as image and speech recognition. The entire process however may include experimenting with different algorithms initially considering factors such as model complexity, interpretability and scalability [20].

After the algorithm, data needs to be split into training and testing sets. The training set as pointed out, teaches the model whereas the testing set evaluates its performance on unseen data [5]. Such splits may include; 70/30 or 80/20. A proper separation ensures that the model can be generalized instead of only memorizing the given training data [21]. Cross-validation techniques such as k-fold cross-validation improves reliability since it divides the data into multiple subsets for training and testing, evaluating the data more effectively [22].

Hyperparameter Tuning is essential as well since it influences the behavior of the model. Hyperparameters can be tuned to optimize the model performance. Relevant combinations can be explored through techniques including grid and random search. Algorithms may have hyperparameters such as learning rates, regularization strengths and the number of layers in neural networks. Tuning is an important part of this process, which finds a balance in maximizing the predictive power of the model, avoiding any overfitting [23].

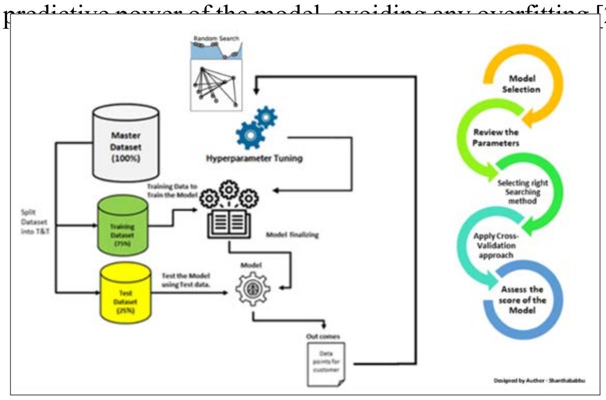

Figure 3: Hyperparameter Tuning

The model is then training using training data, and adjusting the internal parameters of the model based on the data using optimization techniques. Evaluation is an important step within this process to ensure the quality of the model. Important metrics such as accuracy, precision, recall and F1 score can help in classification tasks however mean squared error and R-squared can be used in regression. The evaluation metrics used mainly depend over the problems and the importance of different types of errors.

Deploying Models to Production

After developing a machine learning model, the next step is deployment to integrate it into business applications. This requires converting research prototypes into production-ready systems.

REST API'S

Models are typically exposed as REST APIs that accept input data and return predictions. Python frameworks like Flask can be used to wrap models in API code for inference [24].

Containers

Docker containers package the model code, dependencies and configs into a portable image that can be deployed standalone or orchestrated using Kubernetes [25].

ML Ops

MLOps practices help manage and automate the ML lifecycle including model retraining, continuous integration/deployment, monitoring, governance etc [26].

Monitoring

Monitoring model metrics like latency, errors, data drift during production detects problems early. Logs and telemetry data are ingested into tools like Prometheus, Grafana for alerting.

Resources for Learning and Practice

Resources to learn machine learning are quite abundant, ranging between preferences and skill levels. Online courses and tutorials can be used to guide individuals to get started with machine learning or understand it. Coursera, edX and Udacity have all provided courses from renowned institutions and experts [27]. Even Kaggle and DataCamp provide practical tutorials and challenges that can help the learner apply the concepts within a practical setting [28]. Books and publications can also be used as an essential resource for learning. Publications such as "Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido or "Pattern Recognition and Machine Learning" by Christopher M. Bishop provide a detailed introduction of machine learning concepts. Online communities and forums also provide a vibrant space for knowledge sharing such as Stack Overflow having sections on machine learning where experts and learners engage. Similarly, Reddit's platform provides opportunities to interact with fellow enthusiasts [29]. Meanwhile, learners can also participate in competitions conducted on Kaggle and DrivenData which provides real-time experiences. These competitions provide actual datasets and tasks to solve, asking them to use their skills to compete with others. Datasets can also be used for personal projects and experimentation [28].

Best Practices for Beginners

Algorithm Selection

Choose algorithms suited for the problem type (regression, classification etc), data size and complexity. Simple linear models are fast and interpretable. Decision trees work for classification and regression. SVMs perform well on complex, high-dimensional data [30].

Hyperparameter Tuning

Tune hyperparameters like learning rate, epochs etc. through grid/random search to strike a balance between model performance and overfitting [31].

Prevent Overfitting

Use regularization methods like dropout or early stopping to reduce overfitting on training data. Get more data samples. Use cross-validation for evaluation [32].

Additional Best Practices

Standardize data, use pipelines for consistency [33]
Ensemble models tend to outperform single models [34]
Feature selection removes redundant features [35]

ML for Software Engineering

Code Completion

ML models can suggest context-relevant method/API names and code snippets to engineers while programming based on statistical language models [36].

Bug Detection

ML classifiers identify buggy code by learning patterns from source code metrics like churn, complexity, dependencies etc [37].

Performance Profiling

Models can detect performance regressions by relating code features to performance indicators as training data.

Challenges and Pitfalls

Some of the common mistakes in machine learning includes inadequate data preprocessing leading towards skewed data or overfitting because of which models perform well on training data yet perform poorly on unseen data [14]. Even feature engineering error or improper model selection are also significant challenges [15]. Insufficient validation and testing can also reduce the generalizability of the model [21].

There are a few ethical concerns within machine learning such as bias from training data, putting certain groups at a disadvantage. Models can also perpetuate stereotypes or make unfair decisions especially in hiring and lending, which makes it important to address bias and fairness in machine learning applications [16].

Machine learning systems also face problems with scaling since they can’t manage large datasets or high traffic. Inadequate infrastructure can also create performance bottlenecks. Algorithms need to be optimized, leveraging distributed computing and efficient data storage and retrieval techniques [10].

Conclusion

There are interesting new opportunities emerging in the field of Machine learning due to advancements in deep learning, reinforcement learning and transfer learning. Explainable AI and quantum machine learning are among the recent findings. AutoML will further help simplify model development whereas federated learning improvs privacy. Interpretability and ethics within AI will also gain advance [38].

In context of software development, as models become more complex understanding their internal systems while ensuring ethical compliance is important. Data scientists and experts can be collaborated with to cater data handling, model deployment and addressing these ethical concerns. To prepare for the future, it is necessary to consistently learn and adapt to new tools and frameworks. Moreover, staying updated with industry trends can help create a strong understanding of ethics and challenges within the field. Whereas, collaboration will also be a key strategy in ensuring success within this evolving field [39].

References

1. Dimiduk Dennis M, Elizabeth A Holm, Stephen R Niezgoda (2018) Perspectives on the impact of machine learning, deep learning, and artificial intelligence on materials, processes, and structures engineering. Integrating Materials and Manufacturing Innovation 7: 157-172.
2. Tyagi Amit Kumar, Shashvi Mishra, Shabnam Kumari, Terrance Frederick Fernandez (2021) Intelligent automation systems at the core of industry 4.0. International conference on intelligent systems design and applications. Cham: Springer International Publishing 1-8.
3. Rebala, Gopinath, Ajay Ravi, Sanjay Churiwala (2019) Machine learning definition and basics. An introduction to machine learning 1-17.
4. Morales Eduardo F, Hugo Jair Escalante (2022) A brief introduction to supervised, unsupervised, and reinforcement learning. Biosignal processing and classification using computational learning and intelligence. Academic Press 111-129.
5. Nasteski Vladimir (2017) An overview of the supervised machine learning methods. Horizons 51-62.
6. Teles Germanno, Joel Rodrigues, Ricardo AL Rabelo, Sergei Arkadevich Kozlov (2021) Comparative study of support vector machines and random forests machine learning algorithms on credit operation. Software: Practice and Experience 51: 2492-2500.
7. Howard John (2019) Artificial intelligence: Implications for the future of work. American journal of industrial medicine 62: 917-926.
8. Gevorkyan Migran N, Anastasia Demidova, Tatiana S Demidova, Anton A Sobolev (2019) Review and comparative analysis of machine learning libraries for machine learning. Discrete and Continuous Models and Applied Computational Science 27: 305-315.
9. Ozgur, Ceyhun, Taylor Colliau, Grace Rogers, Zachariah Hughes (2017) MatLab vs. Python vs. R. Journal of data Science 15: 355-371.
10. Khan Asharul Islam, Ali Al-Badi (2020) Open source machine learning frameworks for industrial internet of things. Procedia Computer Science 170: 571-577.
11. Nokeri Tshepo Chris, Tshepo Chris Nokeri (2022) Big Data, Machine Learning, and Deep Learning Frameworks. Data Science Solutions with Python: Fast and Scalable Models Using Keras, PySpark MLlib, H2O, XGBoost, and Scikit-Learn 7-14.
12. Nashif, Shadman, Md Rakib Raihan, Md Rasedul Islam, Mohammad Hasan Imam (2018) Heart disease detection by using machine learning algorithms and a real-time cardiovascular health monitoring system. World Journal of Engineering and Technology 6: 854-873.
13. Zhang, Dongdong, Changchang Yin, Jucheng Zeng, Xiaohui Yuan, Ping Zhang (2020) Combining structured and unstructured data for predictive models: a deep learning approach. BMC medical informatics and decision making 20: 1-11.
14. Famili A, Wei-Min Shen, Richard Weber, Evangelos Simoudis (1997) Data preprocessing and intelligent data analysis. Intelligent data analysis 1: 3-23.
15. Xu Yan, Kai Hong, Junichi Tsujii, Eric I-Chao Chang (2012) Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries. Journal of the American Medical Informatics Association 19: 824-832.

16. Mishra Satwik (2017) Handling imbalanced data: SMOTE vs. random undersampling. Int Res J Eng Technol 4: 317-320.
17. Zhang, Dongdong, Changchang Yin, Jucheng Zeng, Xiaohui Yuan & Ping Zhang et al. (2020) Combining structured and unstructured data for predictive models: a deep learning approach. BMC medical informatics and decision making 20: 1-11.
18. Meng, Xiangrui, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, et al. (2016) Mllib: Machine learning in apache spark.The Journal of Machine Learning Research 17: 1235-1241.
19. Sindhu SS, T Santhanam (2017) Implementing machine learning models as RESTful services using Flask in python.2017 International conference on inventive communication and computational technologies (ICICCT).
20. Christy A, Vaithyasubramanian S, Jesudoss A, Anto Praveena MD (2020) Multimodal speech emotion recognition and classification using convolutional neural network techniques. International Journal of Speech Technology 23: 381-388.
21. Ho Sung Yang, Kimberly Phua, Limsoon Wong, Wilson Wen Bin Goh (2020) Extensions of the external validation for checking learned model interpretability and generalizability. Patterns 1.
22. Wong Tzu-Tsung, Po-Yang Yeh (2019) Reliable accuracy estimates from k-fold cross validation. IEEE Transactions on Knowledge and Data Engineering 32: 1586-1594.
23. Yang Li, Abdallah Shami (2020) On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing 415: 295-316.
24. Bhattacharjee Baishakhi, et al. (2019) IBM deep learning service containers. IBM Journal of Research and Development 63: 3-1.
25. Bosch Thomas, et al. (2022) MLOps: DevOps for machine learning. arXiv preprint arXiv:2201.05828.
26. Aly Mohamed (2005) Survey on multiclass classification methods. Neural Networks.
27. Ausgewiesene Wert, Leistungsnachweis (2022) WI001180: Tech Challenge| Tech Challenge. TUM School of Life Sciences Technische Universität München https://www.ph.tum.de/academics/org/cc/course/950605895/.
28. Hardin Caroline D (2020) Learning to program using online forums a comparison of links posted on Reddit and Stack Overflow. Diss. MS thesis, Univ. of Wisconsin-Madison http://ictwiki.org/carolinescastle/wp-content/uploads/2016/02/CarolineHardinThesisSept2015.pdf.
29. Yeung Christopher, David Ho, Benjamin Pham, Katherine T Fountaine, Aaswath P Raman (2022) Enhancing Adjoint Optimization-Based Photonic Inverse Design with Explainable Machine Learning. ACS Photonics 9: 1577-1585.
30. Claesen, Marc, Bart De Moor (2015) Hyperparameter search in machine learning. arXiv preprint arXiv:1502.02127.
31. Hawkins, Douglas M (2004) The problem of overfitting. Journal of chemical information and computer sciences 44: 1-12.
32. Buitinck Lars, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, et al. (2013) API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238.
33. Opitz David, Richard Maclin (1999) Popular ensemble methods: An empirical study. Journal of artificial intelligence research 11: 169-198.
34. Guyon Isabelle, André Elisseeff (2003) An introduction to variable and feature selection. Journal of machine learning research 1157-1182.
35. Bruch Marcel, Monperrus Martin, Mezini Mira (2009) Learning from examples to improve code completion systems. In Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering 213–222.
36. Nam Jaechang, Sunghun Kim, Wei Fu, Tim Menzies, Lin Tan (2018) Heterogeneous defect prediction. IEEE Transactions on Software Engineering 44: 159-172.
37. Han DongGyun, et al. (2020) PerfLearner: predicting performance regression for deep learning workloads using machine learning. Proceedings of the ACM International Conference on Parallel Architectures and Compilation Techniques.
38. Md Rakib Raihan, Md Rasedul Islam, Mohammad Hasan Imam (2018) Heart disease detection by using machine learning algorithms and a real-time cardiovascular health monitoring system. World Journal of Engineering and Technology 6: 854-873.
39. Ozgur Ceyhun, Taylor Colliau, Grace Rogerset, Zachariah Hughes (2017) MatLab vs. Python vs. R. Journal of data Science 15: 355-371.