**Machine Learning for Log processing**

<https://www.upwork.com/hiring/for-clients/log-analytics-deep-learning-machine-learning/>

### How is Machine Learning Applied for Log Analytics

The basic concept of Machine Learning usage for log [analytics](https://www.upwork.com/hiring/for-clients/analytics-drive-business/) can be explained with an example. As shown in fig three types of inputs are obtained. First input sources are system counters, CPU, memory, disk, and network. Now the second input source is a large amount of distributed logs from different applications around your system. Third input source is consist of error logs, crashing of executable programs, improper shutting down of applications etc. After the collection of all these input sources, a relevant type of information from these logs is extracted automatically with the use of Bayesian Algorithm. Relevant logs are obtained as an output. Machine learning is used to aggregate the logs automatically into correlated categories. Then, newly log data will automatically incorporate into the corresponding category.

Let’s take an example how machine learning can be used to detect system failure automatically. First, select the representation of features of log data and use them to fit the appropriate model according to the given dataset. Training data is used to recognize the failure within the system. Now, evaluate the performance of the model using test dataset. This is the process of supervised learning i.e. log data patterns can be defined in advance. On the contrary, if log data patterns cannot be defined in advance unsupervised learning is introduced. In this approach, most relevant patterns are taken without the need of training dataset provided by the human being.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<https://opensource.com/article/18/9/quiet-log-noise-python-and-machine-learning>

# Quiet log noise with Python and machine learning

## Logreduce saves debugging time by picking out anomalies from mountains of log data.

Red Hat

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ML for DevOps https://www.kdnuggets.com/2018/02/applying-machine-learning-devops.html

Some of the key examples of applying Machine Learning to DevOps include:

1. Tracking Application Delivery

The activity data from ‘DevOps tools’ such as Git, SonarQube, Jira, Ansible and many others provide delivery process visibility. Application of ML on these tools uncover the anomalies in that data- substantial code volumes, long build times, late code check-ins, slow release rates to identify many of the software development wastes, including excessive task switching, gold plating, inefficient resourcing, partial work, or process slowdowns.

1. Ensuring Application Quality

ML, by analyzing output from testing tools can intelligently review QA results, efficiently build a test pattern library based on discovery. This understanding of a ‘known good release’ helps to ensure comprehensive testing on every release, even for novel defects, increasing the quality of delivered applications.

1. Securing Application Delivery

Like fingerprints, User behavior patterns can be unique. Applying Machine Learning to Dev and Ops user behaviors helps in identifying anomalies, which represents dangerous activity. For instance, the access of anomalous patterns to repos, deployment activity, automation routines, test execution, system provision, and more can highlight users were exercising ‘familiar bad patterns’ in a rapid pace both intentionally or accidentally. These patterns include deploying unauthorized code, coding back doors, stealing intellectual property, etc.

1. Managing Production

Machine Learning comes into its own by analyzing an application in production, due to larger data volumes, transactions, etc. occurs in prod when compared to dev or test. The teams of DevOps use ML to analyze general patterns including resource utilization, user volumes, etc. and finally to detect abnormal patterns like memory leaks, DDOS conditions, and race conditions.

1. Managing Alert Storms

The practical and best-value application of ML is in managing the gigantic flood of alerts, which occurs in the systems of production. It can be more complicated such as “training systems overtime to identify ‘known well’ and inadequate warnings, thus enabling filtering to reduce alert storms and fatigue.

1. Troubleshooting and Triage Analytics

The other area where Ml technologies shine today is in triage analytics. It can automatically detect and can triage known issues and some unknown ones even. These tools can detect anomalies in general processing and can analyze the release logs to correlate with new deployments. Even other automation tools can use ML to raise a ticket, alert operations and assign them to the exact source.

1. Preventing Production Failures

In the prevention of failures, ML can go beyond the straight-line capacity planning. It can map utilizing patterns to predict. The required configuration for a desired level of performance, the percentage of clients can use a brand new feature, infrastructure necessities for a brand new promotion, an outage can impact the engagement of customers. ML sees opaque early indicators in applications and systems, allowing Ops to avoid problems faster with quick response times.

1. Analyzing Business Impact

In DevOps, to achieve success, understanding the impact of code release on business goals is critical. ML systems can detect good and bad patterns by analyzing the metrics of the user thus generate an early warning system to business teams and coders when a problem arises in applications.

Conclusion

Though ML has no short or easy button, and there is no alternate for intelligence, creativity, experience, and struggle. But today we see many of its applications and the sky is the limit as we continue to push the boundaries.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<https://www.kdnuggets.com/2017/08/data-version-control-analytics-devops-paradigm.html>

### Why DevOps Matters

The eternal dream of almost every Data Scientist today is to spend all (well, almost all) the time in the office exploring new datasets, engineering decisive new features, inventing and validating cool new algorithms and strategies. However, reality is often different. One of the unfortunate daily routines of a Data Scientist work is to do raw data pre-processing. It usually translates to the challenges to

1. **Pull all kinds of necessary data from a variety of sources**

a. Internal data sources like ERP, CRM, POS systems, or data from online e-commerce platforms

b. External data, like weather, public holidays, Google trends etc.

2.**Extract, transform, and load the data**

a. Relate and join the data sources

b. Aggregate and transform the data

3.**Avoid technical and performance drawbacks** when everything ends up in “one big table” at the end

4. **Facilitate continuous machine learning and decision-making in a business-ready framework**

Utilize historic data to train the machine learning models and algorithms

b. Use the current, up-to-date data for decision-making

c. Export back the resulting decisions/recommendations to review by business stakeholders, either back into the ERP system or some other data warehouse

Another big challenge is to organize **collaboration and data/model sharing** inside and across the boundaries of teams of Data Scientists and Software Engineers.

DevOps skills as well as effective instruments will certainly be beneficial for industrial Data Scientists as they can address the above-mentioned challenges in a self-service manner.

### Can DVC Be a Solution?

Data Version Control (<https://dataversioncontrol.com/>) or simply DVC comes to the scene whenever you start looking for effective DevOps-for-Analytics instruments.

DVC is an open source tool for data science projects. It makes your data science projects reproducible by automatically building data dependency graph (DAG). Your code and the dependencies could be easily shared by Git, and data — through cloud storage (AWS S3, GCP) in a single DVC environment.

Although DVC was created for machine learning developers and data scientists [originally](https://blog.dataversioncontrol.com/data-version-control-beta-release-iterative-machine-learning-a7faf7c8be67), it appeared to be useful beyond it. Since it brings proven engineering practices to not well defined ML process, I discovered it to have enormous potential as an Analytical DevOps instrument.

It clearly helps to manage a big fraction of DevOps issues in daily Data Scientist routines

1. **Pull all kinds of necessary data from a variety of sources**. Once you configure and script your data extraction jobs with DVC, it will be persistent and operable across your data and service infrastructure

2. **Extract, transform, and load the data**. ETL is going to be easy and repeatable once you configure it with DVC scripting. It will become a solid pipeline to operate without major supportive effort. Moreover, it will track all changes and trigger an alert for updates in the pipeline steps via DAG.

3. **Facilitate continuous machine learning and decision-making.** The part of the pipeline facilitated through DVC scripting can be jobs to upload data back to any transactional system (like ERP, ERM, CRM etc.), warehouse or data mart. It will then be exposed to business stakeholders to make intelligent data-driven decisions.

4. **Share your algorithms and data**. Machine Learning modeling is an iterative process and it is extremely important to keep track of your steps, dependencies between the steps, dependencies between your code and data files and all code running arguments. This becomes even more important and complicated in a team environment where data scientists’ collaboration takes a serious amount of the team’s effort. DVC will be the arm to help you with it.

One of the ‘juicy’ features of DVC is ability to support multiple technology stacks. Whether you prefer R or use promising Python-based implementations for your industrial data products, DVC will be able to support your pipeline properly. You can see it in action for both [Python-based](https://blog.dataversioncontrol.com/data-version-control-beta-release-iterative-machine-learning-a7faf7c8be67) and [R-based](https://blog.dataversioncontrol.com/r-code-and-reproducible-model-development-with-dvc-1507a0e3687b) technical stacks.

As such, DVC is going to be one of the tools you would enjoy to use if/when you embark on building continual analytical environment for your system or across your organization.

<https://blog.dataversioncontrol.com/data-version-control-in-analytics-devops-paradigm-35a880e99133>

DVC tutorial; https://dvc.org/doc/dvc-philosophy/collaboration-issues