

Application performance monitoring

Musa Baloyi

August 20, 2018

Table of contents

- ▶ Traditional application performance monitoring
- ▶ Machine learning model monitoring
- ▶ Data generation
 - ▶ Out of the box logging
 - ▶ Custom logging
 - ▶ Data dictionaries
 - ▶ Data types
 - ▶ File types
- ▶ Data movement primer (Kafka)
- ▶ Data transformation primer (Logstash)
- ▶ Data visualisation primer (Kibana)
- ▶ Next steps: application performance management

Logging facility for Python

- ▶ This module defines functions and classes which implement a flexible event logging system for applications and libraries.
- ▶ The key benefit of having the logging API provided by a standard library module is that all Python modules can participate in logging, so your application log can include your own messages integrated with messages from third-party modules.

Logging facility for Python

The basic classes defined by the module, together with their functions, are:

- ▶ Loggers expose the interface that application code directly uses.
- ▶ Handlers send the log records (created by loggers) to the appropriate destination.
- ▶ Filters provide a finer grained facility for determining which log records to output.
- ▶ Formatters specify the layout of log records in the final output.

Logging facility for Python

Logging serves two purposes:

- ▶ Diagnostic logging records events related to the application's operation. If a user calls in to report an error, for example, the logs can be searched for context.
- ▶ Audit logging records events for business analysis. A user's transactions can be extracted and combined with other user details for reports or to optimize a business goal.

dsds_logging guide

1. git clone
`https://<username>@tools.standardbank.co.za/bitbucket/scm/datas
packages.git`
2. `sys.path.append("../python-packages/dsds")`
3. `import dsds.dsds_logging`
4. `config, logger = dsds_spark.get_config_and_logger(sys.argv)`
5. `sc, hiveContext = dsds_spark.get_contexts(config, sys.argv)`
6. `spark_main(config, logger, sc, hiveContext)`
7. `logger.info('Feature_extraction.py', 'feature_set_6',
feature_set_6.columns)`

Sample log (.txt)

```
INFO:20180119_102504:submit:is_test=False
INFO:20180119_102504:submit:username=a231384
INFO:20180119_102504:submit:sys.platform=linux2
INFO:20180119_102504:submit:os.name=posix
INFO:20180119_102504:submit:python.version=(2, 7, 13)
INFO:20180119_102504:submit:max folder age days=2
INFO:20180119_102504:submit:folders deleted=0
INFO:20180119_102504:submit:model_name=sbgm_anomaly_classification
INFO:20180119_102504:submit:conf_environment=dev
INFO:20180119_102504:submit:config.base.project=sbgm_anomaly_classification
INFO:20180119_102504:submit:config.base.team=dsds
INFO:20180119_102504:submit:config.base.environment=dev
INFO:20180119_102504:submit:config.cluster.venv=py2-spark1
INFO:20180119_102504:submit:config.cluster.is-local=false
INFO:20180119_102504:submit:config.cluster.driver-memory=16g
INFO:20180119_102504:submit:config.cluster.num-executors=60
INFO:20180119_102504:submit:config.cluster.executor-memory=13g
INFO:20180119_102504:submit:config.cluster.executor-cores=4
INFO:20180119_102504:submit:config.steps.step-1=create_uri_summary.py
INFO:20180119_102504:submit:config.steps.step-2=create_sessions.py
INFO:20180119_102504:submit:config.steps.step-3=apply_models.py
INFO:20180119_102504:submit:config.steps.step-4=fit_models.py
INFO:20180119_102504:submit:config.data.hist_location=hdfs:///dev/data/dsds/general/history_unzip/
INFO:20180119_102504:submit:config.data.nrt_location=hdfs:///dev/data/dsds/general/nrt/
INFO:20180119_102504:submit:config.data.uri-summaries=[hive][uri_summary]
INFO:20180119_102504:submit:config.data.uri-summaries-new=[hive][uri_summary_new]
INFO:20180119_102504:submit:config.data.sessions=[hive][sessions]
INFO:20180119_102504:submit:config.data.session-summary=[hive][session_summary]
INFO:20180119_102504:submit:config.data.session-summary-new=[hive][session_summary_new]
INFO:20180119_102504:submit:config.data.fitted-models=[hive][fitted_models]
INFO:20180119_102504:submit:config.data.scored-sessions=[hive][scored_sessions]
INFO:20180119_102504:submit:config.depends.local-packages=dsds
INFO:20180119_102504:submit:config.depends.local-files=[read_log_data.py]
INFO:20180119_102504:submit:config.model.earliest_date=2017-08-05
INFO:20180119_102504:submit:config.model.last_history_date=2017-09-17
INFO:20180119_102504:submit:config.model.first_nrt_date=2017-09-20
INFO:20180119_102504:submit:config.model.run_until=2017-09-21
INFO:20180119_102504:submit:config.model.session-timeout=5minutes
INFO:20180119_102504:submit:config.model.max-session-length=30minutes
INFO:20180119_102504:submit:config.model.time_between_fits=4weeks
INFO:20180119_102504:submit:config.model.fit_length=13weeks
INFO:20180119_102504:submit:config.model.cutoff_n_sessions=50
INFO:20180119_102504:submit:config.model.tree_depth=5
INFO:20180119_102504:submit:config.model.n_trees=20
INFO:20180119_102504:submit:config.model.random_sample_per_tree=100
INFO:20180119_102504:submit:config.log.logger_type=FILE
INFO:20180119_102504:submit:config.log.log_location=../../logging
INFO:20180119_102504:submit:config.log.log_level=INFO
INFO:20180119_102504:submit:config=OK
INFO:20180119_102504:submit:step-1=create_uri_summary.py
INFO:20180119_102504:submit:step-2=create_sessions.py
INFO:20180119_102504:submit:step-3=apply_models.py
```

Sample log (.json)

```
INFO:root:{'spark': {'home': None, 'version': '2.1.0.2.6.0.3-8', 'environment': {'PYTHONHASHSEED': '0'}, 'user': 'a231384', 'conf':  
[('spark.eventlog.enabled', 'true'), ('spark.yarn.historyServer.address', 'pdshnn1p.standardbank.co.za:18081'), ('spark.history.ui.port',  
'18081'), ('spark.driver.extraLibraryPath', '/usr/hdp/current/hadoop-client/lib/native:/usr/hdp/current/hadoop-client/lib/native/Linux-  
amd64-64'), ('spark.history.kerberos.keytab', '/etc/security/keytabs/spark.headless.keytab'), ('spark.executor.id', 'driver'),  
'spark.app.id', 'local-1526633937562'), ('spark.yarn.queue', 'default'), ('spark.driver.port', '40470'), ('spark.app.name', 'pyspark-  
shell'), ('spark.executor.extraLibraryPath', '/usr/hdp/current/hadoop-client/lib/native:/usr/hdp/current/hadoop-client/lib/native/Linux-  
amd64-64'), ('spark.driver.host', '10.144.164.203'), ('spark.history.kerberos.principal', 'spark-ds_hdp_prod@ZA.SBICDIRECTORY.COM'),  
'spark.history.fs.logDirectory', 'hdfs:///spark2-history/'), ('spark.sql.catalogImplementation', 'hive'), ('spark.rdd.compress', 'True'),  
'spark.history.provider', 'org.apache.spark.deploy.history.FsHistoryProvider'), ('spark.serializer.objectStreamReset', '100'),  
'spark.master', 'local[*]'), ('spark.submit.deployMode', 'client'), ('hive.metastore.warehouse.dir', 'file:/home/a231384/rta/anomaly-  
detection/sbg-dsds-fraud-anomaly-detection/helpers/digital_anomaly_detection/monitoring/spark-warehouse'), ('spark.port.maxRetries', '100'),  
'spark.eventlog.dir', 'hdfs:///spark2-history/')], 'python': {'version': '3.4'}, 'start_time': '2018-05-18T11:03:04.926190', 'ds_env':  
'\n', 'data': {'historical': 'hdfs:///dev/data/dsds/general/history_unzip', 'near_real_time': 'hdfs:///dev/data/dsds/general/nrt',  
'list_of_hdfs_files': 'list_of_hdfs_files.txt'}, 'modules': ['IPython.core.shadows', 'sklearn.linear_model', 'sys', 'pandas', 'json',  
'logging', 'builtins', 'pickle', 'subprocess', 'time', 'requests', 'pyspark', 'types', 'py4j', 're', 'atexit', 'os', 'datetime', 'builtins',  
'platform', 'random', 'numpy', 'configparser'], 'pyspark': {'submit': {'args': '\n'}}}  
WARNING:root:{}  
ERROR:root:{}  
INFO:root:{'spark': {'home': None, 'version': '2.1.0.2.6.0.3-8', 'environment': {'PYTHONHASHSEED': '0'}, 'user': 'a231384', 'conf':  
[('spark.eventlog.enabled', 'true'), ('spark.yarn.historyServer.address', 'pdshnn1p.standardbank.co.za:18081'), ('spark.history.ui.port',  
'18081'), ('spark.driver.extraLibraryPath', '/usr/hdp/current/hadoop-client/lib/native:/usr/hdp/current/hadoop-client/lib/native/Linux-  
amd64-64'), ('spark.history.kerberos.keytab', '/etc/security/keytabs/spark.headless.keytab'), ('spark.executor.id', 'driver'),  
'spark.app.id', 'local-1526633937562'), ('spark.yarn.queue', 'default'), ('spark.driver.port', '40470'), ('spark.app.name', 'pyspark-  
shell'), ('spark.executor.extraLibraryPath', '/usr/hdp/current/hadoop-client/lib/native:/usr/hdp/current/hadoop-client/lib/native/Linux-  
amd64-64'), ('spark.driver.host', '10.144.164.203'), ('spark.history.kerberos.principal', 'spark-ds_hdp_prod@ZA.SBICDIRECTORY.COM'),  
'spark.history.fs.logDirectory', 'hdfs:///spark2-history/'), ('spark.sql.catalogImplementation', 'hive'), ('spark.rdd.compress', 'True'),  
'spark.history.provider', 'org.apache.spark.deploy.history.FsHistoryProvider'), ('spark.serializer.objectStreamReset', '100'),  
'spark.master', 'local[*]'), ('spark.submit.deployMode', 'client'), ('hive.metastore.warehouse.dir', 'file:/home/a231384/rta/anomaly-  
detection/sbg-dsds-fraud-anomaly-detection/helpers/digital_anomaly_detection/monitoring/spark-warehouse'), ('spark.port.maxRetries', '100'),  
'spark.eventlog.dir', 'hdfs:///spark2-history/')], 'python': {'version': '3.4'}, 'start_time': '2018-05-18T11:03:04.926190', 'ds_env':  
'\n', 'data': {'historical': 'hdfs:///dev/data/dsds/general/history_unzip', 'near_real_time': 'hdfs:///dev/data/dsds/general/nrt',  
'list_of_hdfs_files': 'list_of_hdfs_files.txt'}, 'modules': ['IPython.core.shadows', 'sklearn.linear_model', 'sys', 'pandas', 'json',  
'logging', 'builtins', 'pickle', 'subprocess', 'time', 'requests', 'pyspark', 'types', 'py4j', 're', 'atexit', 'os', 'datetime', 'builtins',  
'platform', 'random', 'numpy', 'configparser'], 'pyspark': {'submit': {'args': '\n'}}}  
WARNING:root:{}  
ERROR:root:}
```


Monitoring

- ▶ Managing and monitoring statistical models is crucial if your organization periodically runs a large number (say, over 10) of statistical models.
- ▶ However, these issues are important even when there are just a few of them in production.

Monitoring

Common challenges include the following:

- ▶ Keeping all the input correct and fresh.
- ▶ Making sure the outputs go to the right places, in the correct formats.
- ▶ Keeping the code organized for effective updating and maintenance.
- ▶ Creating and maintaining effective documentation.
- ▶ Assessing and tracking model performance.
- ▶ Effectively (preferably automatically) deciding when to update the model.

Monitoring: all models

- ▶ Model: name.
- ▶ Environment: continuous development and integration; software and versions; hardware statistics; environment variables; run mode; current build version; source and run location; steps; extra packages and files; run command.
- ▶ Data: historical location; near real-time location; maximum folder age; logs start and end date; last history date; model last date; next run date; first NRT date.
- ▶ Results: submit status; total runtime; FLS alerts; loglines.

Monitoring: supervised models

- ▶ Statistical process control: drift detection method (DDM); early drift detection method (EDDM).
- ▶ Sequential analysis: linear four rates (true -ve, false -ve, true +ve, false +ve) – specificity, recall, precision, accuracy; Monte Carlo sampling for significance level; Bonferoni correction for correlated tests.
- ▶ Error distribution monitoring: adaptive windowing (ADWIN)

Monitoring: unsupervised models

- ▶ Clustering/novelty detection
- ▶ Feature distribution monitoring
- ▶ Model-dependent monitoring

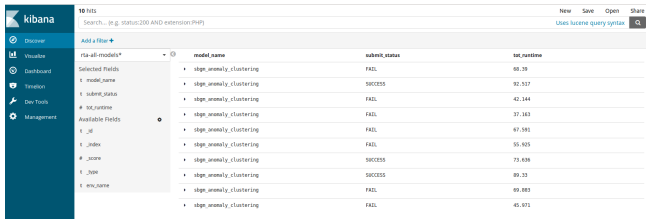
Monitoring: random forests

- ▶ Number of URI's
- ▶ Time between fits
- ▶ Samples per tree
- ▶ Model start date
- ▶ Number of sessions to score
- ▶ Previous run date
- ▶ Maximum session length
- ▶ Last URI timestamp

Monitoring: k-means

- ▶ Database name
- ▶ Results
- ▶ Alerts to FLS
- ▶ Model path
- ▶ List of features
- ▶ Clusters

Visualisation



The image shows a Kibana interface with a table of search results. The left sidebar contains navigation links: Discover, Visualize, Dashboard, Timeline, Dev Tools, and Management. The main area displays a table with 10 hits for the query 'rtm-all-models*'. The table has four columns: model_name, submitt_status, and test_runtime. The first column is expanded to show a list of selected fields: model_name, submitt_status, test_runtime, _id, _index, _score, _type, and _source. The table data shows various model names and their corresponding submitt_status and test_runtime values.

10 hits				New	Save	Open	Share
Search... (e.g. status:200 AND extension:PHP)				Uses lucene query syntax			
Add a filter							
	model_name	submitt_status	test_runtime				
Selected Fields	▸ sbge_anomaly_clustering	FAIL	88.39				
▾ model_name	▸ sbge_anomaly_clustering	SUCCESS	92.537				
▾ submitt_status	▸ sbge_anomaly_clustering	FAIL	42.144				
▾ test_runtime	▸ sbge_anomaly_clustering	FAIL	37.163				
Available Fields	▸ sbge_anomaly_clustering	FAIL	67.591				
▾ _id	▸ sbge_anomaly_clustering	FAIL	55.925				
▾ _index	▸ sbge_anomaly_clustering	SUCCESS	73.636				
▾ _score	▸ sbge_anomaly_clustering	SUCCESS	89.33				
▾ _type	▸ sbge_anomaly_clustering	FAIL	69.883				
▾ _source	▸ sbge_anomaly_clustering	FAIL	45.971				

References

1. Building Intelligent Systems: A Guide to Machine Learning Engineering. [Geoff Hulten] (Apress, 2018)
2. Trends in AI, Data Science, and Big Data. [Ben Lorica] (2017)
3. Building Evolutionary Architectures. [Rebecca Parsons; Patrick Kua; Neal Ford] (O'Reilly Media, 2017)
4. 5 things you should be monitoring. [Brian Brazil] (2018)
5. The Logstash Book. [James Turnbull] (Turnbull Press, 2013)
6. Beyond the Twelve-Factor App. [Kevin Hoffman] (O'Reilly Media, 2016)
7. Logs and real-time stream processing. [Jay Kreps] (2016)
8. I Heart Logs: Apache Kafka and Real-time Data Integration. [Jay Kreps] (2015)
9. The log: The lifeblood of your data pipeline. [Kiyoto Tamura] (2015)
10. Understanding the ELK stack. [Brian Anderson; Rafał Kuć] (2016)