ELTMaestro

Zealot Gurung
MAESTRO ANALYTICS, BOSTON, MA

Contents

ELTMaestro Overview 2
Edge Processing Architecture 2
Machine learning 3
ELTMaestro produces extensive workflow metadata 6
ELTMaestro Standard Features 8
ELTMaestro ML Features 1
Questions? 1

ELTMaestro Overview

ELTMaestro is an enterprise application that supports ELT (Extract Load Transform), ETL (Extract Transform Load), ML (Machine Learning), CDC (Change Data Capture), data quality, data integration, and data lineage. ELTMaestro works with a large and growing number of target data warehouse platforms, including Redshift, Snowflake, Yellowbrick, Exasol, Greenplum, Azure Synapse, Spark/Hadoop and Databricks.

ELTMaestro has a graphical interface that allows users to create jobs as dataflow diagrams, as is customary with traditional ETL tools such as Informatica or DataStage. ELTMaestro extracts data from standard sources such as databases and CSV files as well as more specialized sources such as cellphone apps, Salesforce, mainframes, and many others. ELTMaestro has a built-in scheduler and extensive user-customizable data quality reporting. ELTMaestro includes advanced log-based change data capture and other CDC protocols. ELTMaestro also includes a comprehensive machine learning system and supports integration of machine learning and ETL processes.

ELTMaestro's subscription-based pricing model is very advantageous for rapidly growing and evolving production environments. ELTMaestro software licenses do not restrict data volume, data sources, bandwidth, memory, processing, or number of users. ELTMaestro subscription fees do not change as you grow your data warehousing platform.

Edge Processing Architecture

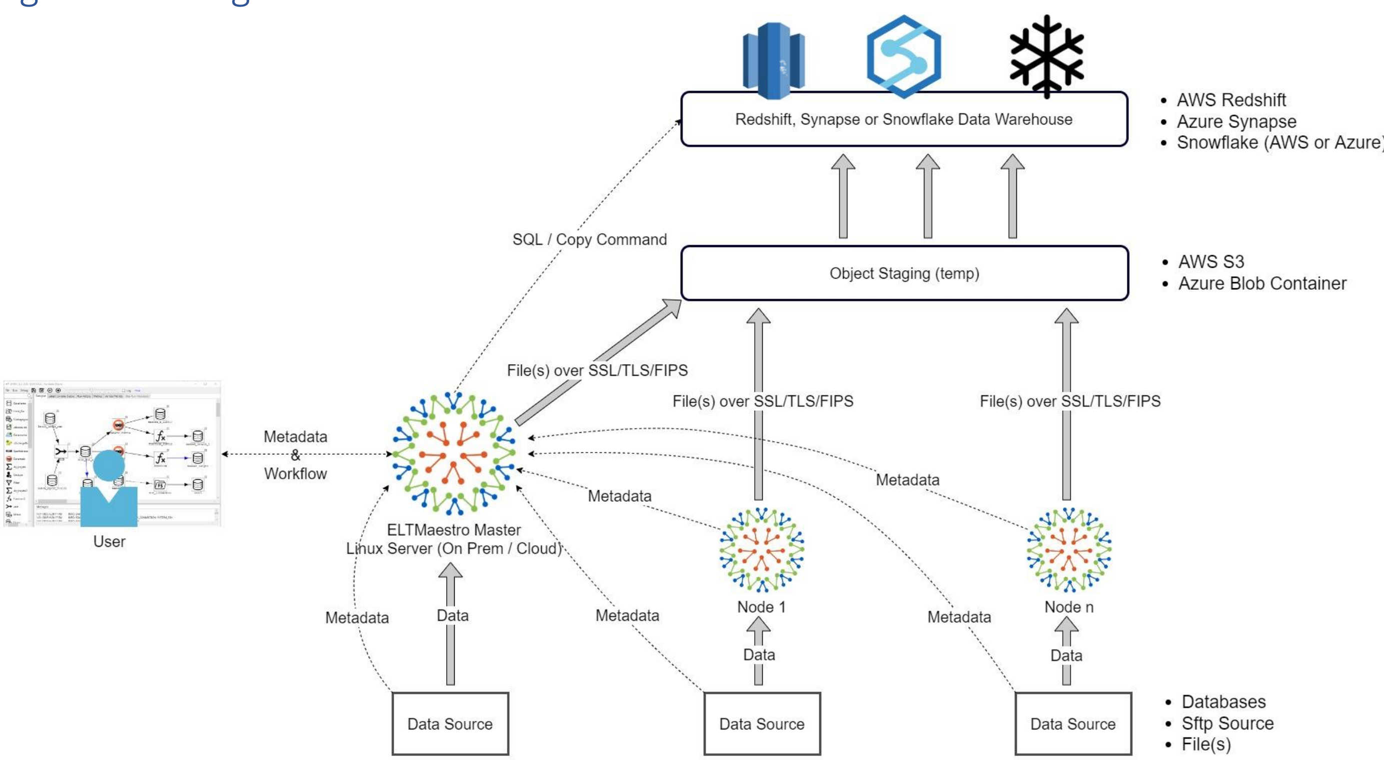


Figure 1 ELTMaestro Edge Processing Architecture

Edge processing architecture eliminates ETL server bottlenecks when source data is geo-distributed.

- Users design data load workflows using the ELTMaestro graphical client. Metadata is provided to user from the master server.
- The ELTMaestro master server then uses schedule and deployment information to decide if the workflows should be executed locally or pushed to remote nodes. In either case, data is extracted from source, compressed, and securely transferred to object storage (such as S3 or Blob). ELTMaestro then issues the appropriate SQL command (e.g., copy, load, or external table) to load files directly to target platform.

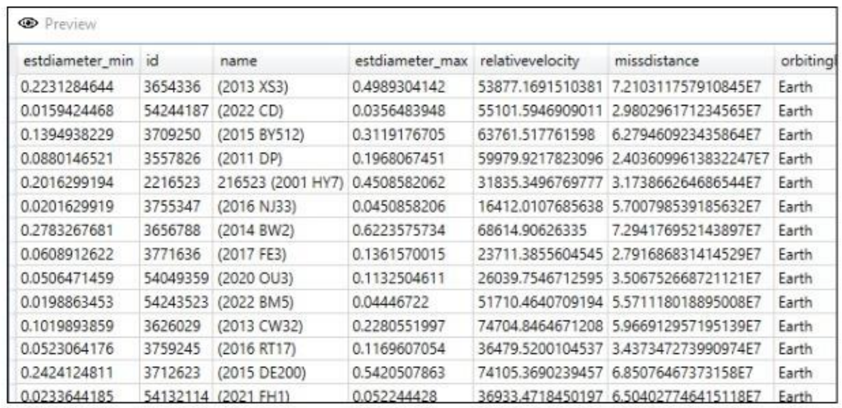
- Edge nodes allow customers to optimize data load speed when their data is distributed over multiple remote geographical locations, obviating the need to accumulate geo-distributed data on a single ETL server. This eliminates bottlenecks and saves intranet bandwidth. Moving large datasets becomes faster, more efficient, and less expensive.

Machine learning

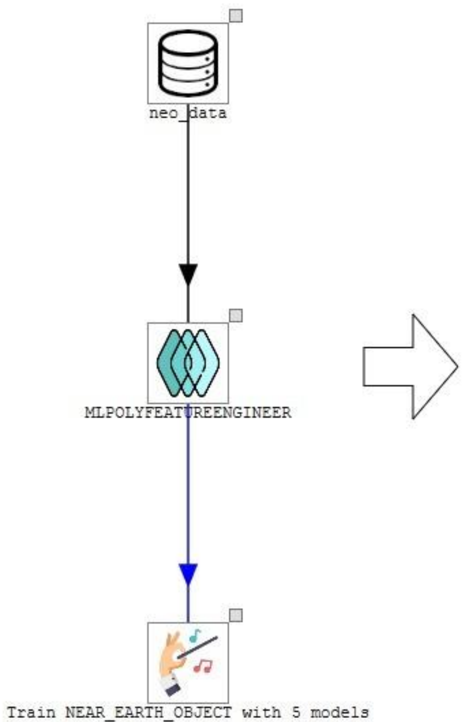
ELTMaestro includes Machine Learning as a standard feature. To build an ML model you simply point to a dataset, select and configure a model, and execute.

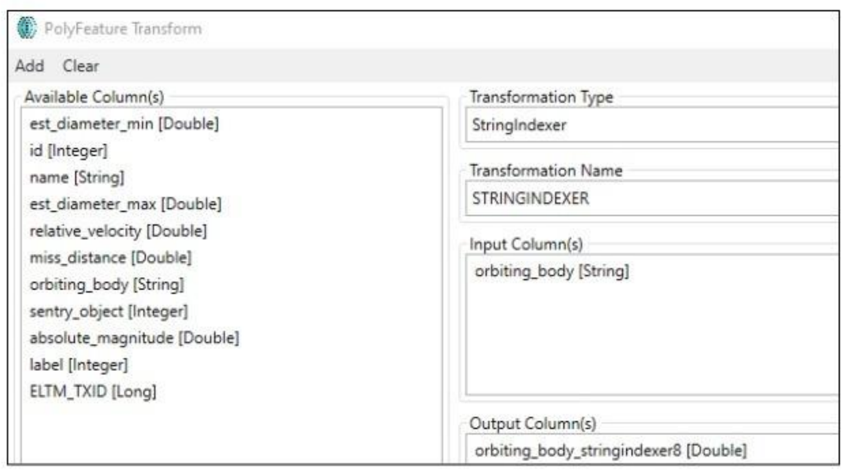
An example is shown below.

Figure: ELTMaestro ML Training Workflow


Preview

estdiameter_min	id	name	estdiameter_max	relativevelocity	missdistance	orbiting
0.2231284644	3654336	(2013 XS3)	0.4989304142	53877.1691510381	7.210311757910845E7	Earth
0.0159424468	54244187	(2022 CD)	0.0356483948	55101.5946909011	2.980296171234565E7	Earth
0.1394938229	3709250	(2015 BY512)	0.3119176705	63761.517761598	6.279460923435864E7	Earth
0.0880146521	3557826	(2011 DP)	0.1968067451	59979.9217823096	2.4036099613832247E7	Earth
0.2016299194	2216523	216523 (2001 HY7)	0.4508582062	31835.3496769777	3.173866264686544E7	Earth
0.0201629919	3755347	(2016 NJ33)	0.0450858206	16412.0107685638	5.700798539185632E7	Earth
0.2783267681	3656788	(2014 BW2)	0.6223575734	68614.90626335	7.294176952143897E7	Earth
0.0608912622	3771636	(2017 FE3)	0.1361570015	23711.3855604545	2.791686831414529E7	Earth
0.0506471459	54049359	(2020 OU3)	0.1132504611	26039.7546712595	3.506752668721121E7	Earth
0.0198863453	54243523	(2022 BM5)	0.04446722	51710.4640709194	5.571118018895008E7	Earth
0.1019893859	3626029	(2013 CW32)	0.2280551997	74704.8464671208	5.966912957195139E7	Earth
0.0523064176	3759245	(2016 RT17)	0.1169607054	36479.5200104537	3.437347273990974E7	Earth
0.2424124811	3712623	(2015 DE200)	0.5420507863	74105.3690239457	6.850764673731 58E7	Earth
0.0233644185	54132114	(2021 FH1)	0.052244428	36933.4718450197	6.504027746415118E7	Earth





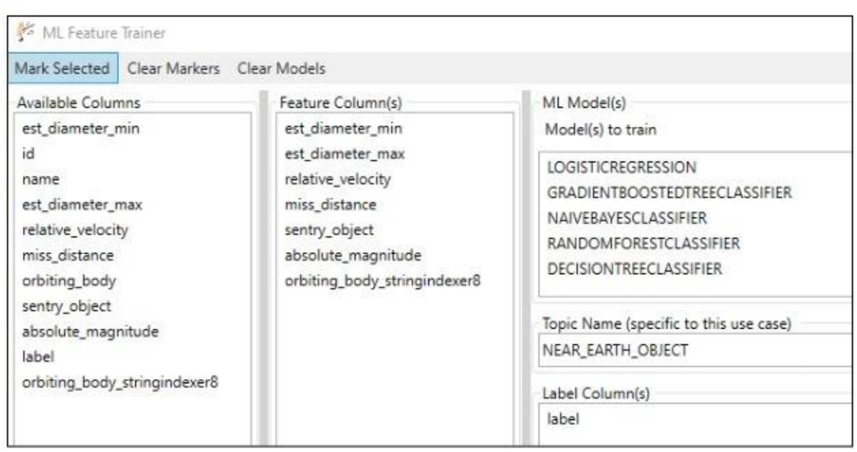


Figure 2 ELTMaestro ML Training Workflow

When the workflow is deployed and executed, ELTMaestro converts it into Spark ML code. Upon completion, ELTMaestro displays the model's evaluations and accuracy.

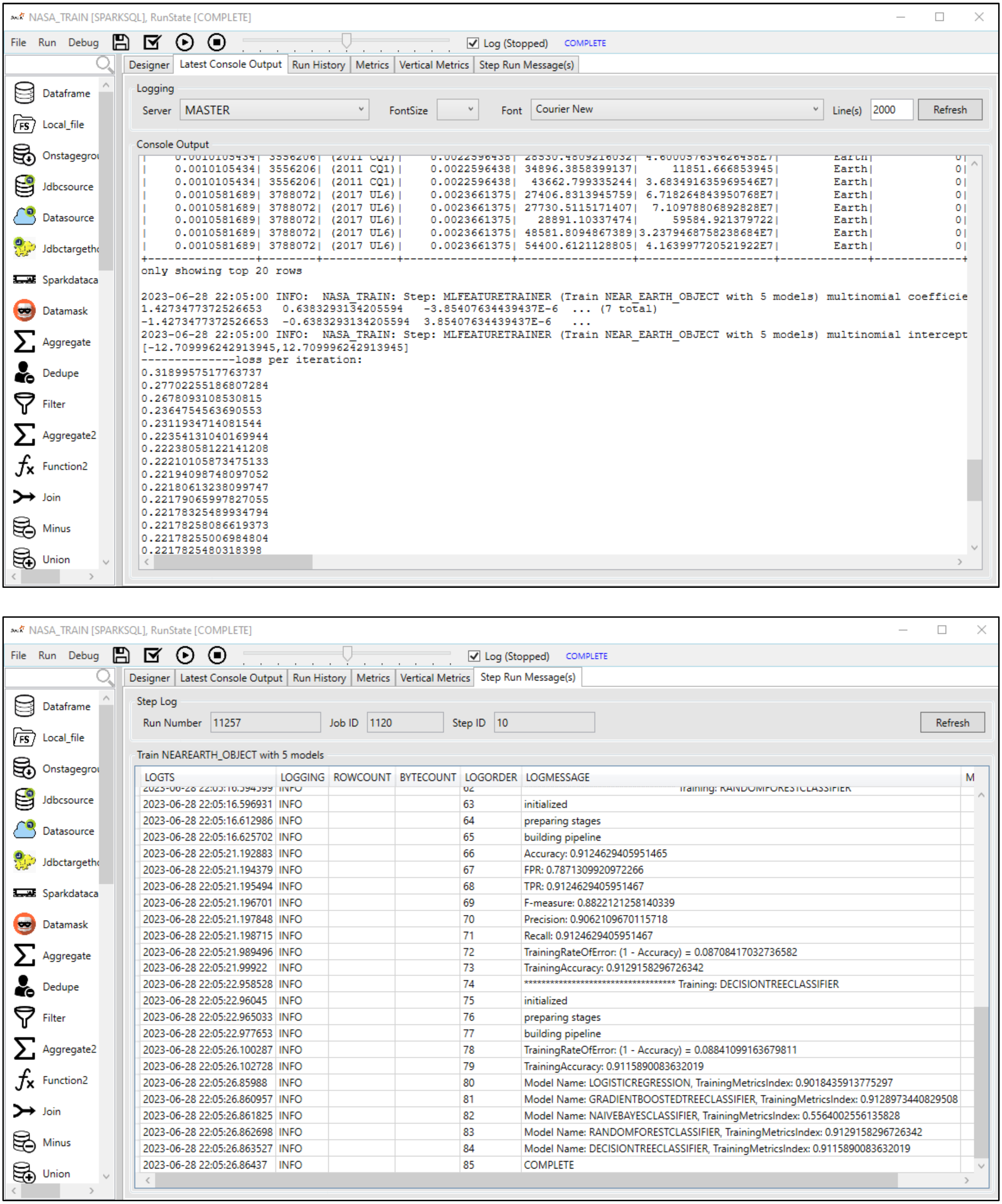


Figure 3 ML Workflow logs and accuracy

Similar pipelines can be built for predictions:

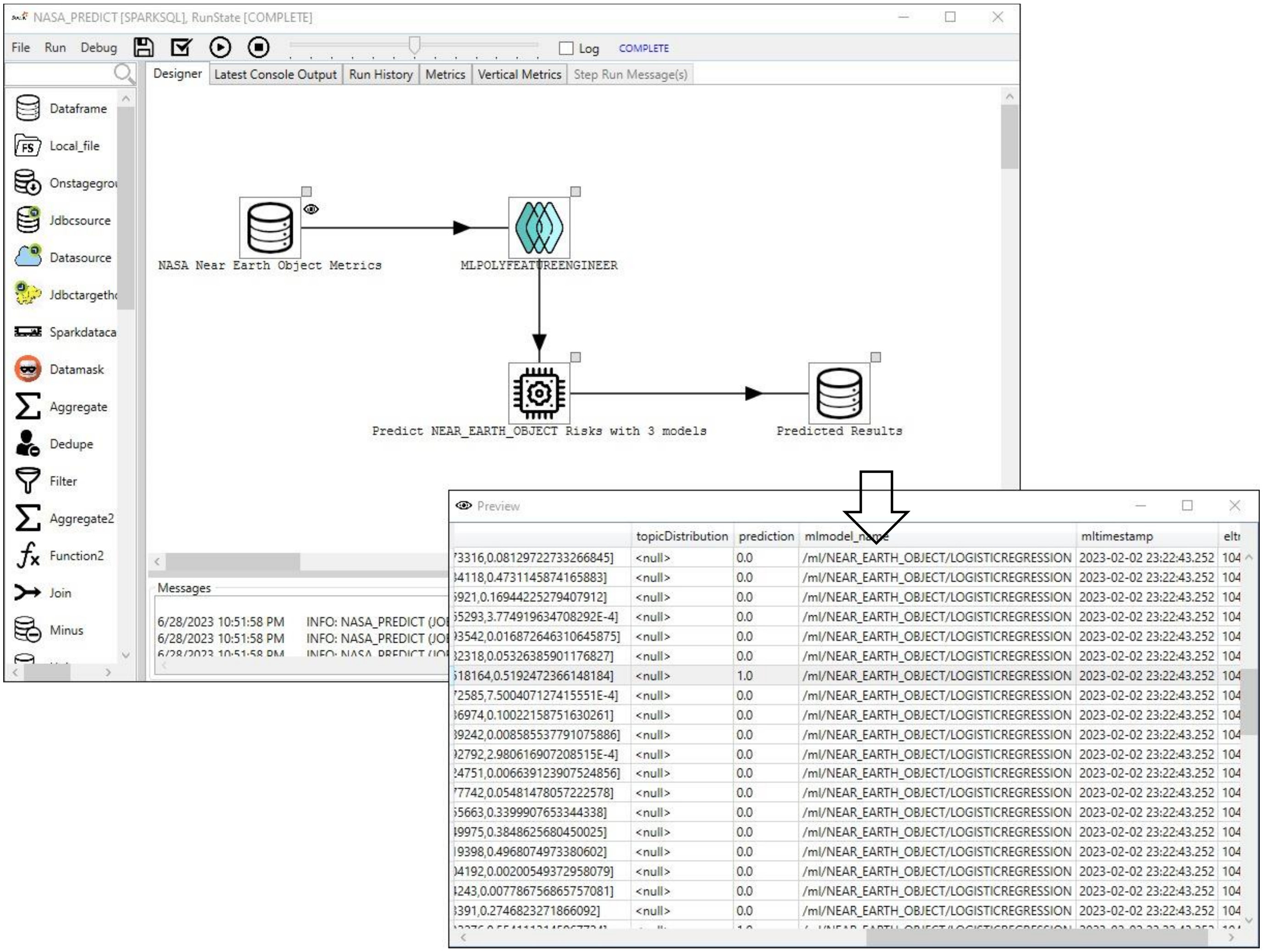


Figure 4 ELTMaestro ML Prediction Workflow

Prediction results can be graphically displayed on dashboards:

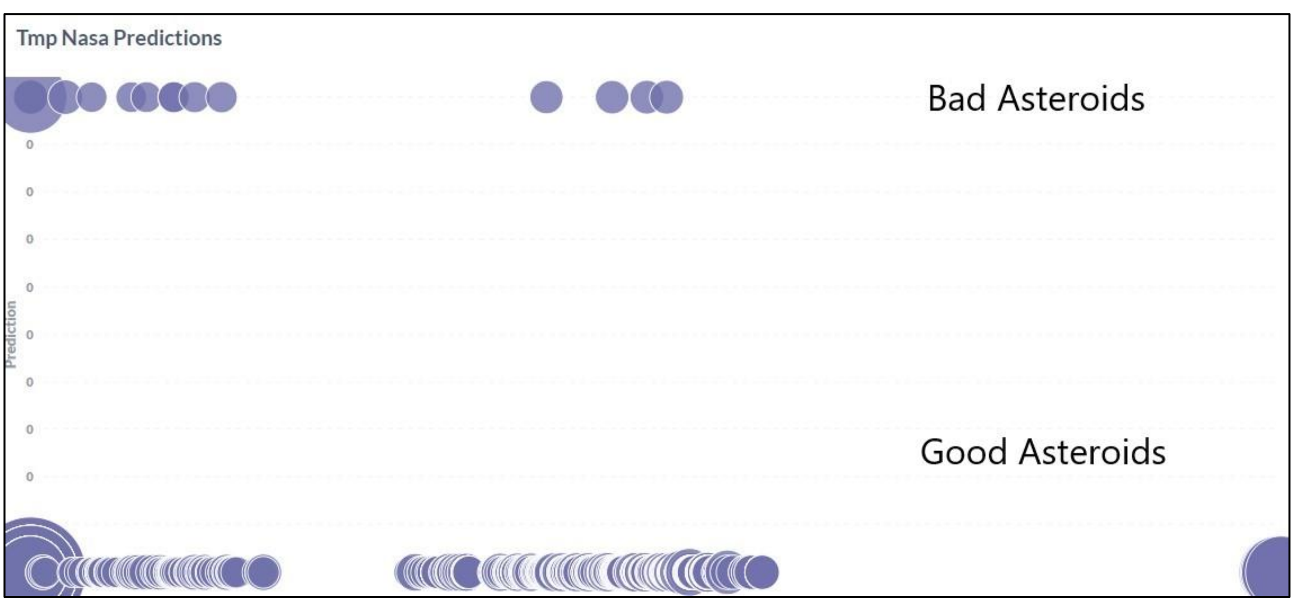


Figure 5 Predictions

ELTMaestro produces extensive workflow metadata.

ELTMaestro transforms data by converting workflows to SQL that runs against the target data warehouse. ELTMaestro keeps track of record counts, data lineage, and runtime quality metrics.

For example, the workflow below, designed by the graphical interface, is converted into SQL that executes on the data warehouse platform.

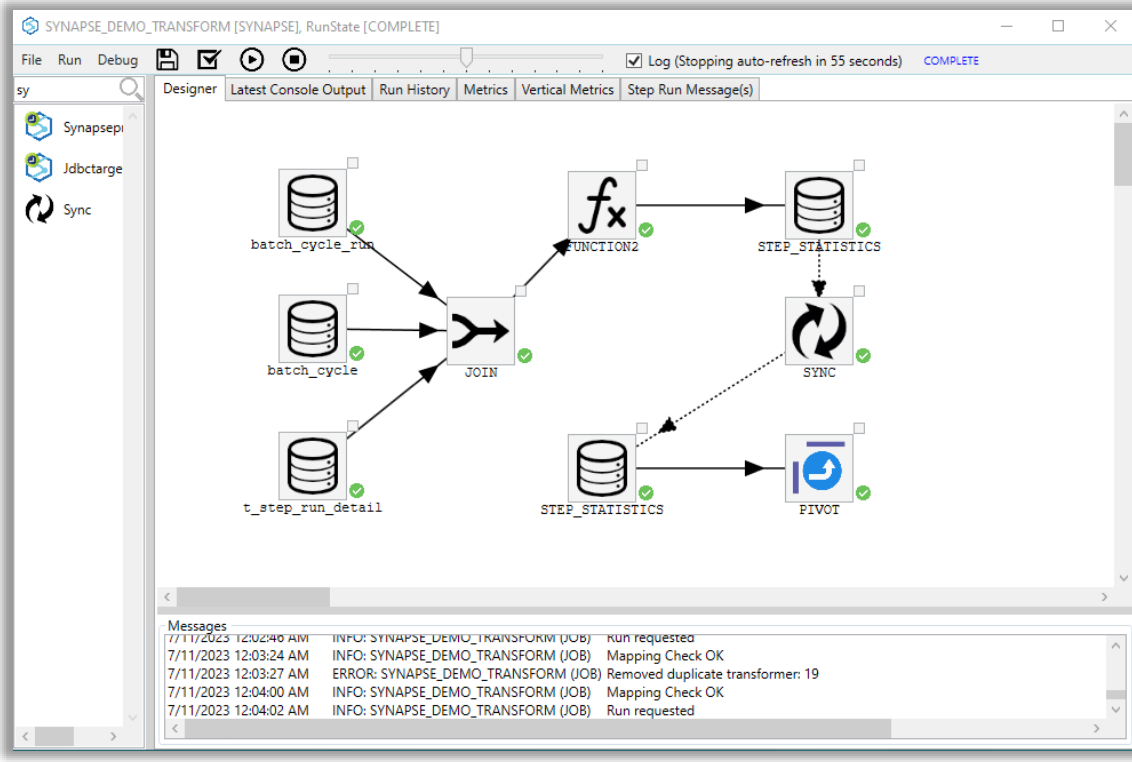


Figure 6 Workflow

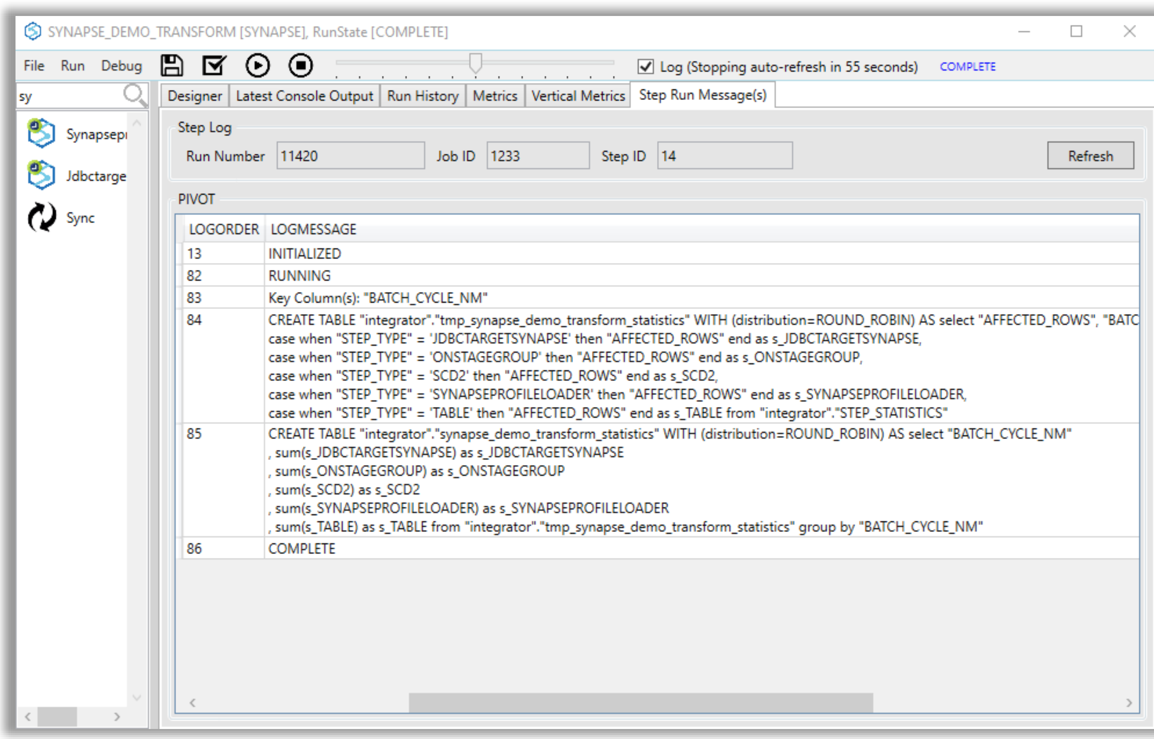


Figure 7 Workflow log

ELTMaestro maintains runtime history, including the SQL that was executed by each step:

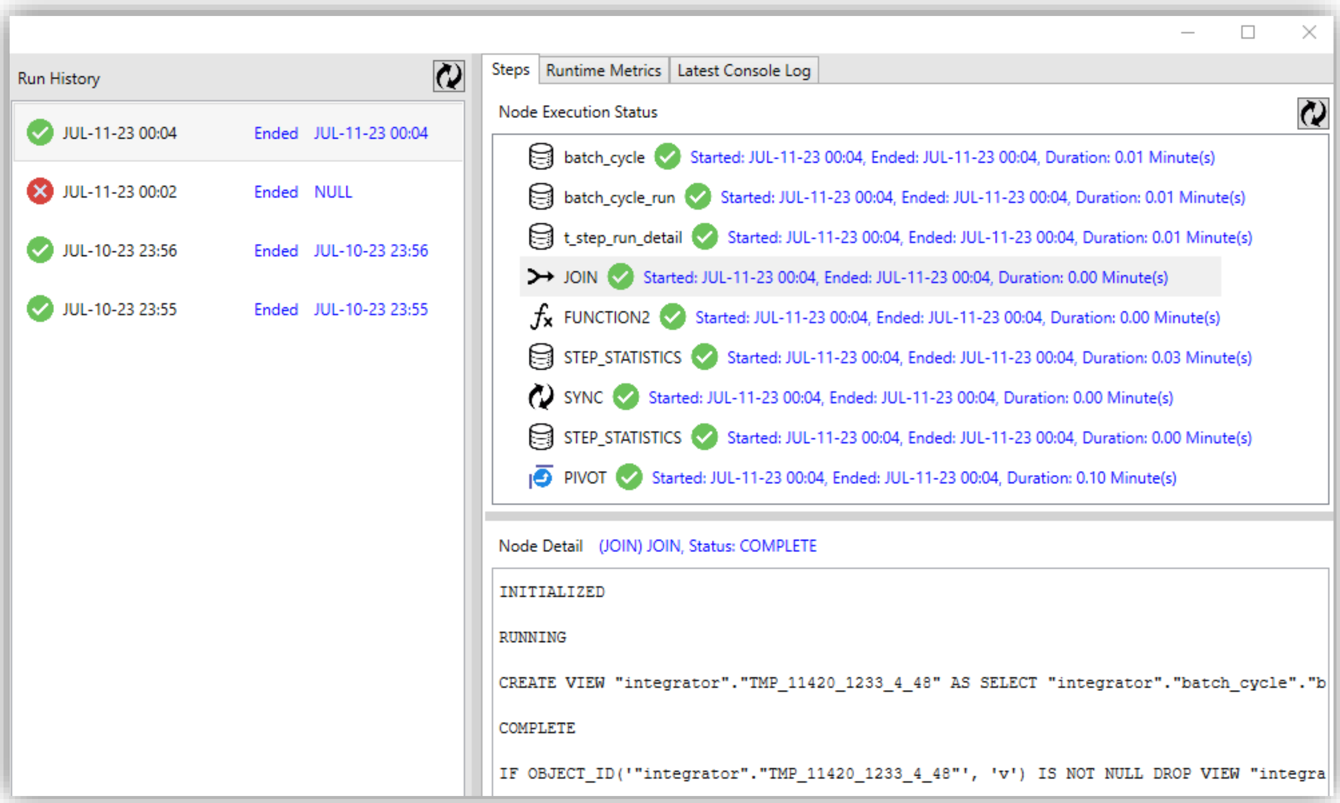
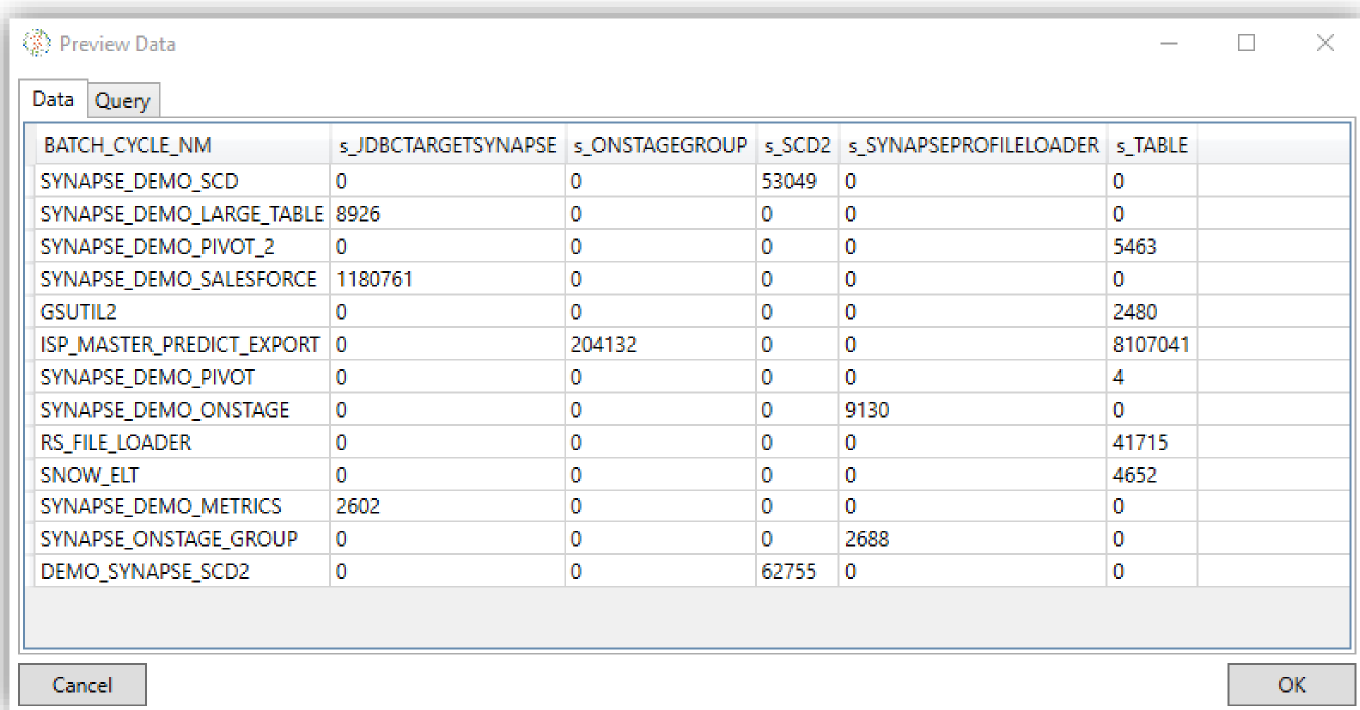


Figure 8 Runtime history

ELTMaestro maintains data quality information of all nodes:



BATCH_CYCLE_NM	s_JDBCTARGETSYNAPSE	s_ONSTAGEGROUP	s_SCD2	s_SYNAPSEPROFILELOADER	s_TABLE
SYNAPSE_DEMO_SCD	0	0	53049	0	0
SYNAPSE_DEMO_LARGE_TABLE	8926	0	0	0	0
SYNAPSE_DEMO_PIVOT_2	0	0	0	0	5463
SYNAPSE_DEMO_SALESFORCE	1180761	0	0	0	0
GSUTIL2	0	0	0	0	2480
ISP_MASTER_PREDICT_EXPORT	0	204132	0	0	8107041
SYNAPSE_DEMO_PIVOT	0	0	0	0	4
SYNAPSE_DEMO_ONSTAGE	0	0	0	9130	0
RS_FILE_LOADER	0	0	0	0	41715
SNOW_ELT	0	0	0	0	4652
SYNAPSE_DEMO_METRICS	2602	0	0	0	0
SYNAPSE_ONSTAGE_GROUP	0	0	0	2688	0
DEMO_SYNAPSE_SCD2	0	0	62755	0	0

Figure 9 Automatic data quality metrics by row count

ELTMaestro automatically maintains data lineage information:

data_context_name	data_context_name_origin	data_context_detail
/ingest/failure_types	ONSTAGE:HANA:HANA.INTEGRATOR.FAILURE_TYPES	JDBCTARGETHDFS:/ingest/failure_types
/ingest/machine_metrics	ONSTAGE:REDSHIFT:dev.public.machine_metrics	JDBCTARGETHDFS:/ingest/machine_metrics
dev.demo.dummy_98	ONSTAGE:ORACLE_CONNECTION:ORCLCDB.SYSTEM.DUMMY_98	JDBCTARGETREDSHIFT:dev.demo.dummy_98
dev.ot.dummy_98	ONSTAGE:ORACLE_CONNECTION:ORCLCDB.SYSTEM.DUMMY_98	JDBCTARGETREDSHIFT:dev.ot.dummy_98
DWH_ETL.PUBLIC.PERSON22	ONSTAGE:SQL SERVER:AdventureWorks2019.Person.Person	JDBCTARGETSNOWFLAKE:DWH_ETL.PUBLIC.PEF
DWH_ETL.PUBLIC.PASSWORD2	ONSTAGE:SQL SERVER:AdventureWorks2019.Person.Password	JDBCTARGETSNOWFLAKE:DWH_ETL.PUBLIC.PA
SYNAPSE.integrator.TMP_11421_1233_4_4	INSITU:SYNAPSE:"integrator"."batch_cycle_run"	JOIN:SYNAPSE.integrator.TMP_11421_1233_4_4
iqasnyapsepool.integrator.STEP_STATISTI	INSITU:SYNAPSE:SYNAPSE.integrator.TMP_11421_1233_11_61	TABLE:Existing Table [Create=True]
iqasnyapsepool.integrator.STEP_STATISTI	INSITU:SYNAPSE:null.null	TABLE:Existing Table [Create=False]
gpadmin.dev.batch_cycle_run	ONSTAGE:ABC:sqlmaestro.public.batch_cycle_run	JDBCTARGETGREENPLUM:gpadmin.dev.batch_c
gpadmin.public.batch_cycle	ONSTAGE:ABC:sqlmaestro.public.batch_cycle	ONSTAGEGROUP:gpadmin.public.batch_cycle
gpadmin.public.batch_cycle_run	ONSTAGE:ABC:sqlmaestro.public.batch_cycle_run	ONSTAGEGROUP:gpadmin.public.batch_cycle_r
gpadmin.isp_integration.cust_predicted_d	INSITU::gpadmin.integrator.TMP_10558_1183_1_21	TABLE:Existing Table [Create=False]
gpadmin.integrator.TMP_10637_1178_4_6	INSITU::"isp_customer"."InternetPackage"	JOIN:gpadmin.integrator.TMP_10637_1178_4_67

Figure 10 Data lineage

ELTMaestro Standard Features

Feature Type	Transformation	Orchestration Mode	Misc
Data Loading	None	Bulk Load Delta Load Watermark Load Parallel Load	Source: Database, Salesforce, Object Storage, SFTP, Files
Data Loading using CDC	SQL Replay	100% SQL	Log Mining
In-Situ Transformation	Table/DataFrame, Join, Deduplicate, Union, Minus, Filter, Data Masking,Pivot,Window/Scalar/Aggregate Functions, Slowly Changing Dimensions, +more	100% SQL	Generated on runtime
Machine Learning	Spark	100% Spark Code	Regression, Classification and Clustering
Data Quality/Lineage	None	Automatically Maintained	
Data Un-Loading	Export to file(s), object storage, databases		
Controls	Custom Metrics, File/Database/Script Watcher, Variables, SQL Script, Email Alerts, Smart Script, SSH Script	Depends on control feature	
Automatic Recovery		Job/Step/Node auto-restart on failures and threshold configuration	
Scheduling			Pre-defined and custom scheduling

ELTMaestro ML Features

Feature Engineering / Transformations	**Regression**
CountVectorizerTransform	DecisionTreeRegressor
DCTTransform	FactorizationMachinesRegressor
ElementwiseProductTransform	GeneralizedLinearRegressor
HashingTransform	GradientBoostedTreeRegressor
IDFTransform	LinearRegressor
MaxAbsScalerTransform	RandomForestRegressor
MinMaxScalerTransform	
NGramTransform	**Classification**
NormalizerTransform	DecisionTreeClassifier
PCATransform	FactorizationMachinesClassifier
PolynomialExpansionTransform	GradientBoostedTreeClassifier
RegexTokenizerTransform	LinearSupportVectorMachineClassifier
RobustScalerTransform	LogisticRegression
StandardScalerTransform	LogisticRegressionOneVsRestClassifier
TokenizerTransform	MultilayerPerceptronClassifier
VectorIndexerTransform	NaiveBayesClassifier
Word2VecTransform	RandomForestClassifier
BinarizerTransform	
BucketizerTransform	**Clustering / Neural Network**
FeatureHasherTransform	BisectingKMeansCluster
ImputerTransform	GaussianMixtureModelCluster
InteractionTransform	KMeansCluster
OneHotEncoderTransform	LDACluster
QuantileDiscretizerTransform	
StopWordsRemoverTransform	**Documentation Source**
StringIndexerTransform	https://spark.apache.org/docs/latest/ml-classification-regression.html
VectorAssemblerTransform	https://spark.apache.org/docs/latest/mllib-clustering.html

Questions?

Contact us at pen@maestro-analytics.com