FACTORS AFFECTING MACHINE LEARNING ALGORITHMS AND LIBRARIES SELECTION

Boris Zibitsker, PhD, BEZNext; Dominique Heger, PhD, Data Analytica;



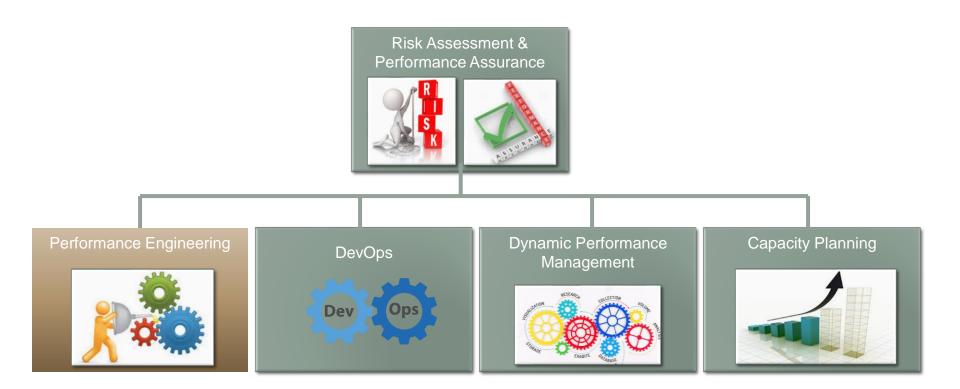
ICPE 2018

Outline

- Introduction
- Challenges
- Objective
- Benchmarking and Modeling
- Collaboration
- Methodology
- Data Collection Results
- Modeling Results
- Recommender
- Summary



Introduction Performance Engineering and Performance Assurance



Challenges



- Business Requirements to IT: Accuracy, Timeliness, Throughput, Scalability
- Selection of ML Algorithms affects ability of IT meeting Business Goals
- ML Algorithms are "Atomic Components" of IT
- ML Algorithm performance depends on Number of observations / rows, number of columns / predictors, hardware and software configuration

Challenges Selection of ML Algorithms Affects Business and IT IT Business

- Users
 - · Customers, Local, Partners, Vendors
- Measurement Data
 - Performance, Resource Usage
 - Accuracy
 - Scalability
 - Cost
- Hardware and Software Systems
 - Big Data Clusters, EDW, Data Center, Cloud
- Workload
- Application
- Data
- ML Algorithm / ML Library

- Business Measures
 - P&L, Balance Sheet, etc.
- Business Plan
 - Growth
- Service Level Goals
- Line of Business

Regression Algorithms	Method		Python
			Python Function
	Ordinary Least Squares Regression	sklearn.linear_model	LinearRegression()
Focus on the highlighted MLAs	Lasso (least absolute shrinkage and selection operator)	sklearn.linear_model	linear_model.Lasso()
Focus on the highlighted MLAS	Multi task lasso	sklearn.linear_model	linear_model.MultiTaskLasso()
	Elastic Net	sklearn.linear_model	linear_model.ElasticNet()
Ridge Regression		sklearn.linear_model	linear_model.Ridge()
Ridge Regression	Ridge	sklearn.linear_model	linear_model.Ridge()
	Lasso	sklearn.linear_model	linear_model.Lasso()
	LassoLars	sklearn.linear_model	linear_model.LassoLars()
		sklearn.linear_model	linear_model.MultiTaskLasso()
	Elastic Net	sklearn.linear_model	linear_model.ElasticNet()
Elastic Net	Na drifte a latte a ria Na a	- Lite and Barrier and state	line and the state of the state
1			linear_model.MultiTaskElasticNet()
Lars		sklearn.linear_model	linear_model.Lars()
		sklearn.linear_model	
	brm DevenienDidee	sklearn.linear_model	linear model Devecier Didae()
Bayesian Regression		sklearn.linear_model	linear_model.BayesianRidge()
		sklearn.linear_model	linear_model.ARDRegression()
		PyMC3	
	m any	bayespy	
	ke nel Riuge	sklearn.kern an idge	kernel_ridge.KernelRidge()
kernel regression	lemel Ridge	mlpy	kernelRidge()
			nonparametric.kernel_regression.Kern
	nonpa ametr : Kern I i eg	a smodels	elReg()
	ksmort	sklearn.linear_model	
SVR (Suppport Vector Regression)	2 m		svm.SVR(), SVR()
		sklearn.linear_model	
SGD (Stochastic Gradient Descent	<u> </u>		linear_model.SGDRegression()
			gaussian_process.GaussianProcessR
Gaussion Processs Regress on	GaussianProce sRe ressor		egressor()
Regression Tree		sklearn.tree	tree.DecisionTreeRegressor()
Bagging		sklearn.linear_model	
		sklearn.linear_model	
		sklearn.linear_model	
Random Forest		sklearn.linear_model	
C	rando i Fore t	sklearn.linear_model	
AdaBoosting	enser Ible Ac 1B and Regressor	sklearn.linear_model	
Gradient Boosted Regression Trees	U U	sklearn.linear_model	
- · · · · · · · · · · · · · · · · · · ·	gbm -	sklearn.linear_model	
Neural Network	neural_network.MLPRegressor	sklearn.linear_model	neural_network.MLPRegressor()
K Neariest Neighbours	KNeighborsRegressor	sklearn.linear model	neighbors.KNeighborsRegressor()
-	Least Angle regression	skleam.linear_model	
		sklearn.linear_model	
		sklearn.preprocessing sklearn	
		sklearn.linear_model	
All Pights Posserved			
All Rights Reserved	Isotonic Regression	sklearn.linear_model	

Classification		Pytho	on
Algorithms	Algorithm	Library	Function
	Binary Logistic Regression	sklearn.linear_model	LogisticRegression()
	Multinomial Logistic Regression	sklearn.linear_model	LogisticRegression()
Linear Classifier	Ordinal Logistic Regression	N/A	N/A
	Fisher's Linear Discriminant Analysis	sklearn.discriminant_analysis	LinearDiscriminantAnalysis()
	Naive Bayes classifier	sklearn.naive_bayes	MultinomialNB(), BernoulliNB(), GaussianNB()
Quadratic Classifier	Quadratic Discriminant Analysis	sklearn.discriminant_analysis	QuadraticDiscriminant/ nalysis
Decision Trees	Classification Tree	s (lea n.t. e	DecisionTreeClassifier()
	Bagging	sk learn.cosemble	EaggingClassifier()
	Random Forest	sklearn on semble	Random Fol estClassifier()
	Gradient Boosting	splearn.ensemble	G.adientBoostingClassifier()
Ensemble Model	Ada Boosing Pticat	on Algori	
	chilo	sklearn.ensemble	AdaBoostClassifier()
r c	1255.		
Support Vector Machines		sklearn.svm	SVC(), LinearSVC(), NuSVC()
Neural Network	-	sklearn.neural_network	MLPClassifier()
Stochastic Gradient Descent		sklearn.linear_model	SGDClassifier()
Nearest Neighbors	KNeighbors	sklearn.neighbors	KNeighborsClassifier()
	RadiusNeighbors	sklearn.neighbors	RadiusNeighborsClassifier()
Gaussian Processes All Rights Reserved		sklearn.gaussian_process	GaussianProcessClassifier()

Clustering	Almorithm	Python			
Algorithms	Algorithm	Library	Function		
Centroid-based	K-means clustering	sklearn.cluster	KMeans(), MiniBatchKMeans()		
Distributed-based	Gaussian mixture models	k earn-mixture	Gaussian Aixture()		
	LDA Challen	gens im nodels			
Connectivity-based	Hierarchical clustering OOS	sklearn.cl ister	AgglomerativeClustering()		
10	al Optimit	sklearn.cluster	DBSCAN()		
Density-based	EVE CIUSIO		Birch()		
oft	Spectual clustering	sklearn.cluster	SpectralClustering()		

Challenges



How to choose Right Machine Learning Algorithm and ML Library



How to choose Right Hardware and Software configuration

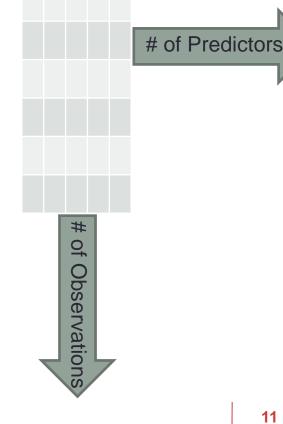
Objectives of our project

- Develop Methodology
 - How to select appropriate ML algorithm and ML library providing sufficient accuracy, response time, scalability, minimize usage of resources and cost
 - Select Data Set
 - How to collect performance measurement data
- Use Models
 - Develop Models expanding measurement results
 - Use Models to compare benchmarks results done in different environment
- Organize Collaboration
 - Organize a Collaboration to benchmark different algorithms in parallel
 - Use common methodology to benchmark ML algorithms and libraries
- Develop Recommender for Data Scientists and Application Developers
 - Create knowledge base and web application
 - Develop algorithm selecting appropriate ML algorithm and ML library based on business requirements

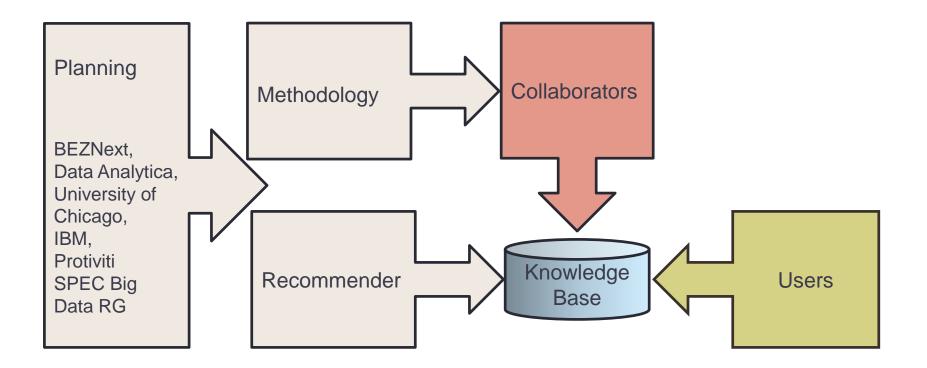
- Determine minimum data set size to achieve desired level of accuracy and time of 10 $^{\rm All\,Rights}$ model training

Role of Benchmarks and Modeling

- Evaluate performance, usage of resources, scalability and accuracy of ML algorithms and libraries
- Analyze the impact of the size of the data set on time of • the models training
- Benchmarking process includes the following steps: •
 - Preparing the Data Sets with different numbers of observations and predictors
 - Preparing and running the Benchmark test:
 - Writing Python programs
 - Creating the benchmark environment
 - Collecting measurement data: response time, accuracy, CPU, memory usage, I/O rate and network utilization
- Modeling:
 - Building models to predict performance characteristics of the different ML algorithms and ML Libraries for different sizes of the data sets not included into the benchmarks
- Model validating:
 - Comparing the prediction results with actual measurement data for data set with different number of observations and predictors.



Collaborative Approach



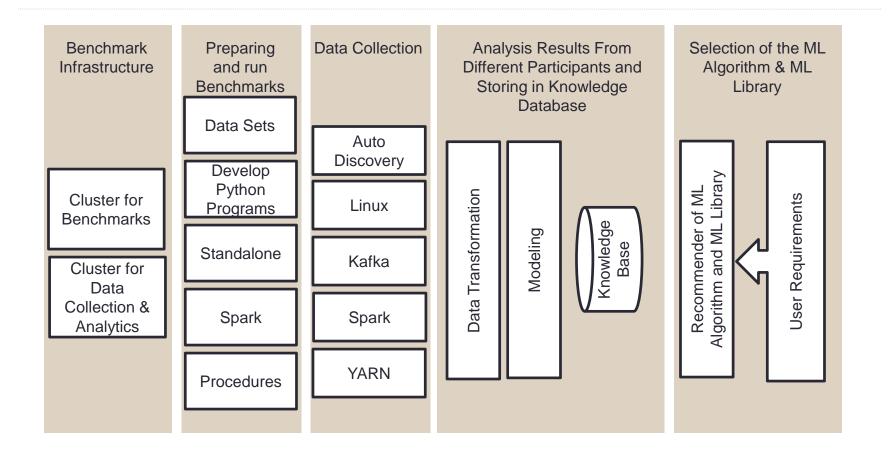
Example of Business Requirements

Туре	Relative Weight
Accuracy	0.2
Response Time	0.4
CPU Utilization	0.2
Memory Utilization	0.2
Total	1



Relative importance of the different requirements to new application

Benchmarking Process



Infrastructure Spark Cluster with 4 Data Nodes Provided by IBM

Host	uchicago-hadoop-host-	uchicago-hadoop-host-	uchicago-hadoop-host-	uchicago-hadoop-host-
	01.bigdatauniversity.com	02.bigdatauniversity.com	03.bigdatauniversity.com	05.bigdatauniversity.com
IP	10.114.57.125	10.114.57.122	10.114.57.115	10.114.57.117
Linux	Ubuntu 16.04.2 LTS (4.4.0)	Ubuntu 16.04.2 LTS (4.4.0)	Ubuntu 16.04.2 LTS (4.4.0)	Ubuntu 16.04.2 LTS (4.4.0)
CPU	8x2.6 GHz	8x2.6 GHz	8x2.6 GHz	8x2.6 GHz
RAM	15.7 GB	15.7 GB	15.7 GB	15.7 GB
Space	4TB	4TB	4TB	
<mark>Services</mark>	datanode-3 : 10.42.82.202	datanode-2 : 10.42.47.208	datanode-1: 10.42.121.65	datanode-4: 10.42.219.248
	yarn-nodemanager-3 : 10.42.12.72	yarn-nodemanager-4 : 10.42.175.132	yarn-nodemanager-1 : 10.42.59.195	yarn-nodemanager-2 : 10.42.18.233
	spark-worker-2: 10.42.205.153	spark-worker-3 : 10.42.36.4	spark-worker-1: 10.42.13.255	spark-worker-4: 10.42.194.67
		jupiter-1 : 10.42.204.60	hue-1:10.42.73.11	
		zeppelin-1: 10.42.255.224		

15

Infrastructure Spark Cluster with Four Nodes are used for Data Collection and Management

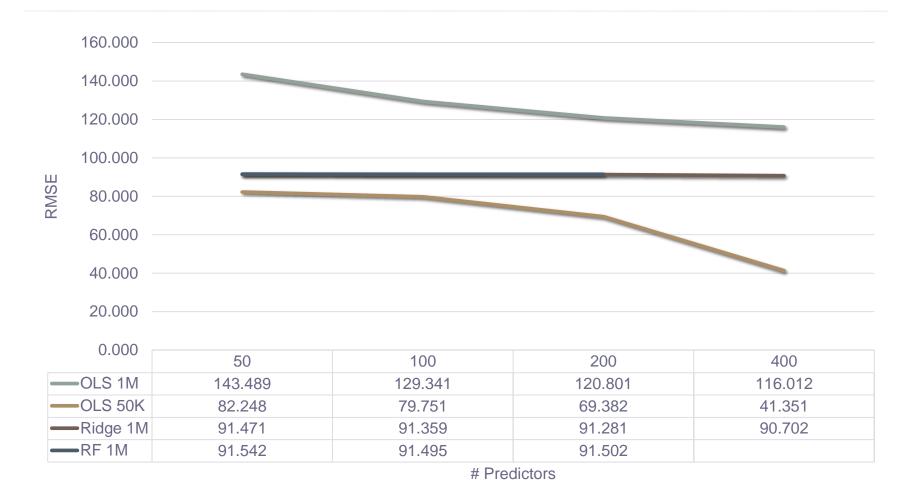
Host	uchicago-misc-host- 01.bigdatauniversity.com	uchicago-hadoop-host- 04.bigdatauniversity.com	uchicago-spark-host- 01.bigdatauniversity.com	uchicago-spark-host- 02.bigdatauniversity.com
IP	10.114.57.118	10.114.57.116	10.114.57.126	10.114.57.114
Linux	Ubuntu 16.04.2 LTS (4.4.0)	Ubuntu 16.04.2 LTS (4.4.0)	Ubuntu 16.04.2 LTS (4.4.0)	Ubuntu 16.04.2 LTS (4.4.0)
CPU	4x2.6 GHz	8x2.6 GHz	4x2.6 GHz	4x2.6 GHz
RAM	7.83 GB	15.7 GB	7.83 GB	7.83 GB
Space	4TB	4TB	4TB	4TB
Services	HDFS Client1Spark clientBEZVisionBEZVision RepositoryOracleAnalytics WebAppWebApp RepositoryMariaDBAnalytics Python scriptsBenchmarks scripts	HDFS namenode-primary-1 : 10.42.150.176 yarn-resourcemanager-1 : 10.42.9.96 Spark jobhistory-server-1 : 10.42.85.10 spark-master-1 : 10.42.171.182 Kafka broker-1 : 10.42.114.64	spark-master-1: 10.42.188.22 spark-worker-2: 10.42.107.244 datanode-2:10.42.88.226 <u>namenode-primary-1</u> : 10.42.245.223 <u>yarn-resourcemanager-1</u> : 10.42.82.156 <u>jobhistory-server-1</u> : 10.42.170.253	spark-worker-1: 10.42.242.8 datanode-1:10.42.204.132 yarn-resourcemanager-1: 10.42.152.78 BEZ Agent Manager Linux agents, YARN and Spark agents UDT/YLT/BVT
All Rights Reserve		Zookeeper zk-1 : 10.42.52.9		

Data Collection Process

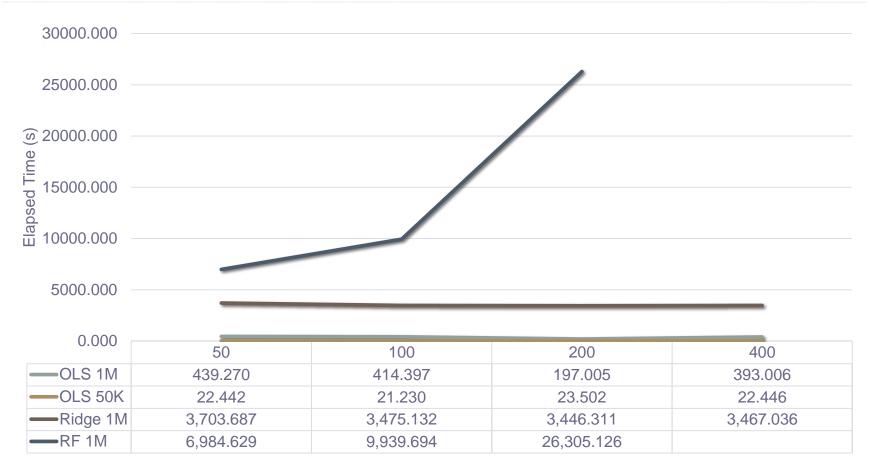
- Benchmarking Ordinary Least Squares Regression (OLS), Ridge Regression (Ridge) and Random Forest (RF)
 - Standalone Python and Python on Spark
 - Each algorithm was tested for 3 dataset sizes of 5k, 50k and 1m observations and for 4 different number of predictors
- Measurement data include
 - Accuracy, Response time, CPU utilization, Memory usage, I/O rate and Network throughput



Accuracy RMSE vs # Predictors

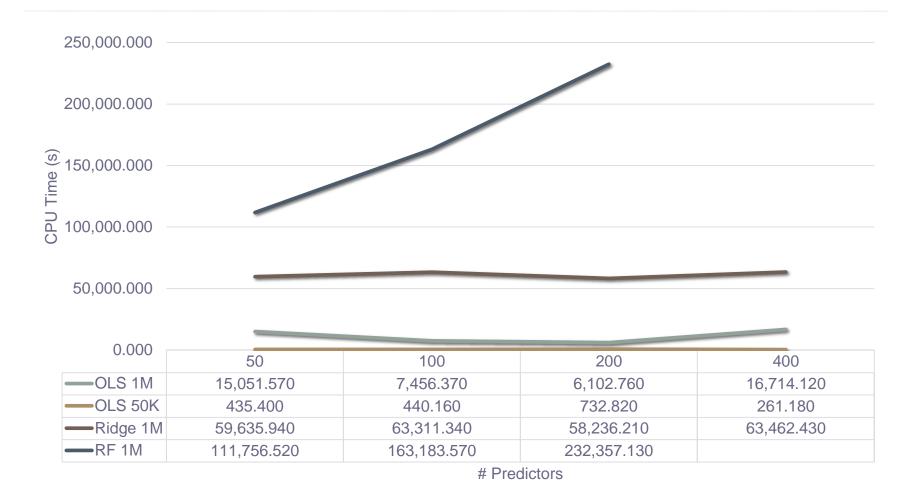


Elapsed Time vs. #Predictors

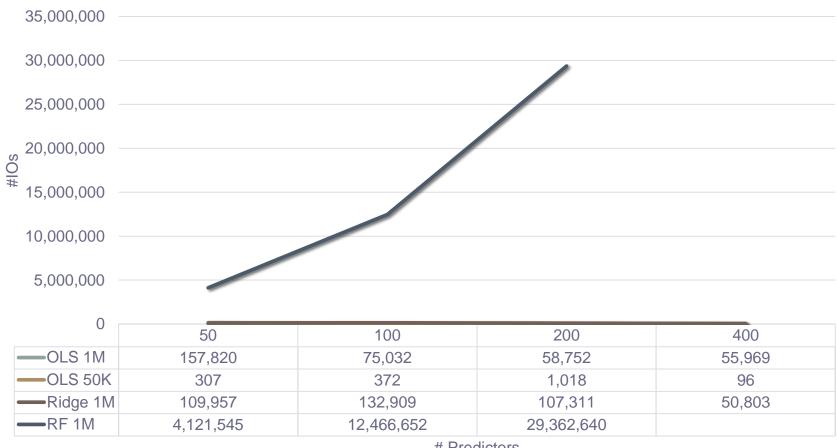


Predictors

CPU Time Across All Nodes vs # Predictors

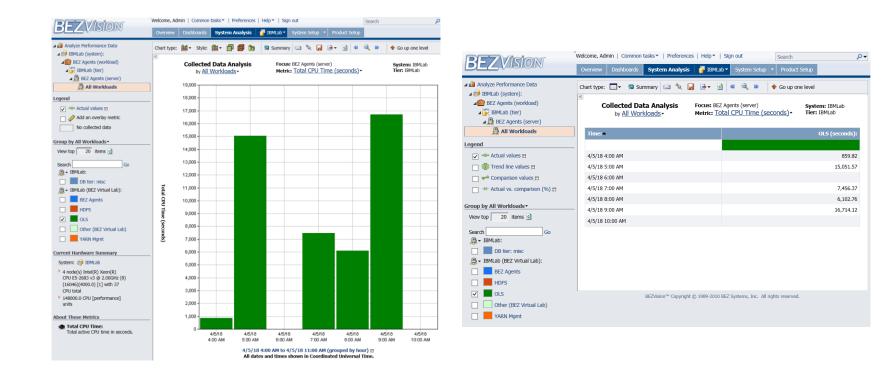


Total I/O across All Nodes vs #Predictors



Predictors

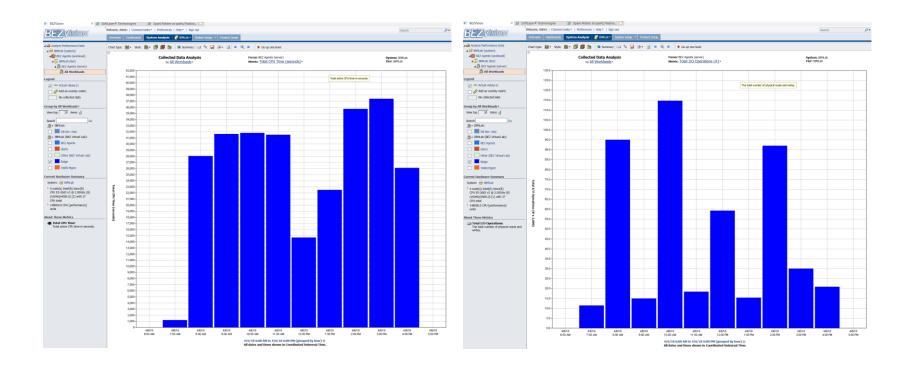
Total CPU time used for OLS 1M rows and 50, 100, 200, 400 predictors starting at 5, 7, 8 and 9 AM



Total IO count for OLS 1M rows and 50, 100, 200, 400 predictors for Benchmark starting at 5, 7, 8, and 9 AM

BMLab (system):	e via	🕼 - 🞒 👹 🎁 🖄 Summary 🖾 🍬 🔒 🕒 😒					Welcome, Admin Common tasks * Preferences Help * Sign out	Search
Contract (workload)		Collected Data Analysis by All Workloads -	Focusi BEZ Agents (server) Metric: Total I/O Operations (#)*		Systems IBMLab Tior: IBMLab	BEZVISION		
BEZ Agents (server)	180.0						Overview Dashboards System Analysis 🚰 IBMLab - System	Setup 👻 Product Setup
Q All Workloads	175.0							
Actual volves 53	170.0			The total number of physical reads and writes,		🔺 🎒 Analyze Performance Data	Chart type: 🔲 🗸 🥱 Summary 🖂 🍾 🛃 🅞 🛛 🍕 🍳	Go up one level
nti an overlay metric	165.0					⊿ 🥩 IBMLab (system):		
No collected data	160.0					BEZ Agents (workload)		
γ All Workloads • p 20 hams)	155.0					⊿ IBMLab (tier)	by All Workloads • Focus: BEZ Agents (server Metric: Total I/O Ope	
	150.0					BEZ Agents (server)	by <u>All Workloads</u> • Metric: <u>Total 1/0 Ope</u>	iduoiis (#)* inel ioneau
h Go IBMLab:	140.0							
DB tier: misc IBMLab (BEZ Virtual Lab):	135.0					🚊 All Workloads	Time:▲	OLS (#)
BEZ Agents	130.0					Legend		
HOPS	125.0						-	
OLS Other (BEZ Virtual Lab)	120.0					🔽 🔶 Actual values 🖯	4/5/18 4:00 AM	1,39
YARN Mgmt	115.0					🔲 🐺 Trend line values 🕫	4/5/18 5:00 AM	157,82
Hardware Summary	110.0					Comparison values 5	4/5/18 6:00 AM	
m: 🧀 TEMLab	5 100.0							
vode(s) Intel(R) Xeon(R) U E5-2683 v3 @ 2.00GHz (8)	1/0 950					🔄 🐣 Actual vs. comparison (%) 🖯	4/5/18 7:00 AM	75,03
9046)(4000.0) [1] with 37 U total	900-						4/5/18 8:00 AM	58,75
(8000.0 CPU [performance] sits	3005 85.0-					Group by All Workloads*	4/5/18 9:00 AM	55,969
These Metrics	80.0 10 75.0					View top 20 items 🖄		
Total I/O Operations: The total number of physical reads and	70.0						4/5/18 10:00 AM	
writes.	65.0					Search Go		
	60.0							
	55.0					DB tier: misc		
	50.0							
	45.0							
	35.0					BEZ Agents		
	30.0					HDFS		
	25.0							
	20.0						BEZVision™ Copyright © 1989-2010 BEZ Systems, Ir	rc. All rights reserved.
	15.0-					Other (BEZ Virtual Lab)		
	50					YARN Mgmt		
	00-	45/10 4.5/10	45/10 45/10	4510	45/10 4/5/10			

Total CPU time and IO count: Ridge 1M rows and 500 predictors Start at 8, 10 AM, 12PM 2PM with 2 hours each run with most of I/O happens during the first minute

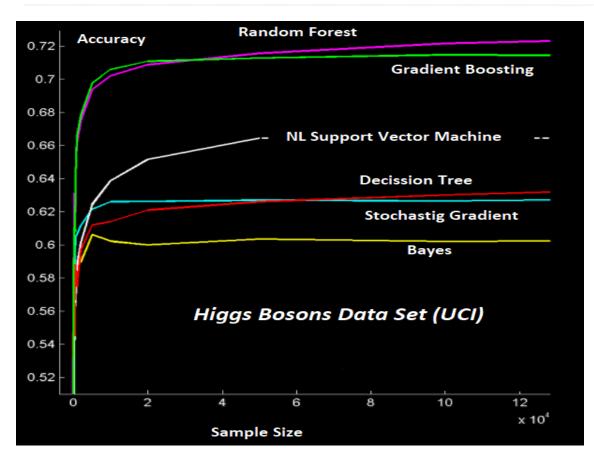


Surprises

- Spark response time exceeds
 Standalone Python
- CPU Utilization and #I/Os
- Memory utilization
- ML algorithms accuracy depends on training Data Set size

- Partitioning
- Size of the data size
- Machine Learning Libraries Implementation
- Docking containers

ML algorithms accuracy depends on training Data Set size

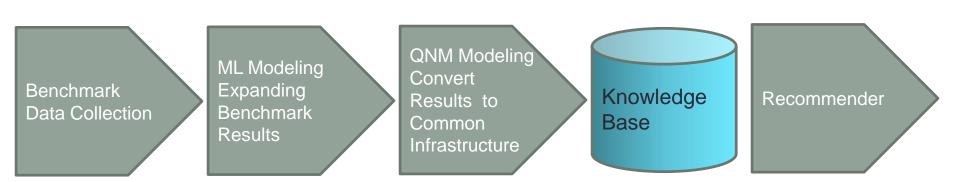


- Higgs Boson Data Set (Cern Benchmark) with 11m rows and 28 predictors https://archive.ics.uci.edu/m I/datasets/HIGGS
- Results of testing ML algorithms on various training data set sizes.
- Training data size affects training and running time.

Modeling Expands the Results of Benchmarks

- Preparation and conducting the benchmark test is time consuming. In our case each benchmark test run for 2 hours. Therefore, the benchmark results include the limited number of data points.
- In order to expand the results of the benchmark tests we applied ML and Queueing Network Models.
- For ML based models the measurement data were split into two data sets: 80% of data were used for model training and 20% of data were used to compare the actual measurement data with prediction results. Trained models are used to predict Response Time, CPU Utilization, I/O rate, Memory Utilization, and Accuracy for each ML Algorithm and ML Library

Role of Models



- Benchmarks are done by collaborators in parallel;
- ML models are used to expand benchmark results;
- Queueing network models (QNM) are used to convert measurement data collected by different collaborators on different clusters into baseline configuration
- QNM are used to justify capacity management measures

Modeling Expands Results of the Benchmark Tests

- Number of benchmarks tests is limited
- ML models can be used to fill gaps and enable evaluation of scenarios where tests were not performed
- Different ML algorithms are evaluated to predict the metrics and estimate the accuracy by comparing prediction results with measurement data
- Measurement data were split into two data sets:
 - 80% of data were used for model training and
 - 20% of data were used to compare the actual measurement data with prediction results.
 - Trained models are used to predict Response Time, CPU Utilization, I/O rate, Memory Utilization, and Accuracy for each ML Algorithm and ML Library.



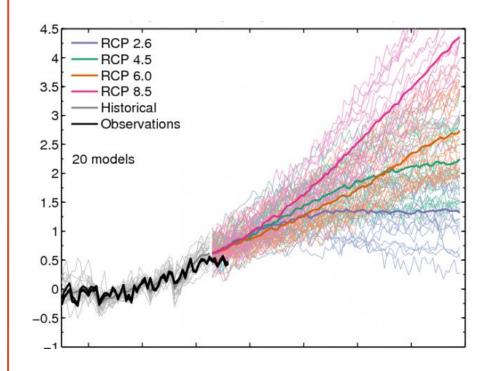
Prediction Models

Accuracy = f(Algorithm, Library, # of Observations, log(log(# of Features));

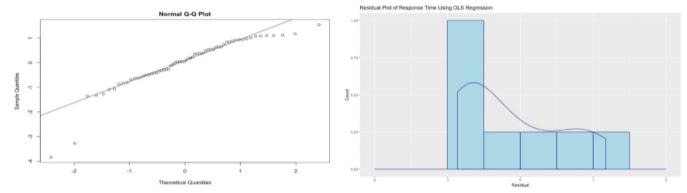
CPU Utilization= f(Algorithm, Library, # of Observations, # of Features);

log(Memory Usage) = f(Algorithm, Library, # of Observations, # of Features);

log(Response Time) = f(Algorithm, Library, # of Observations, # of Features);

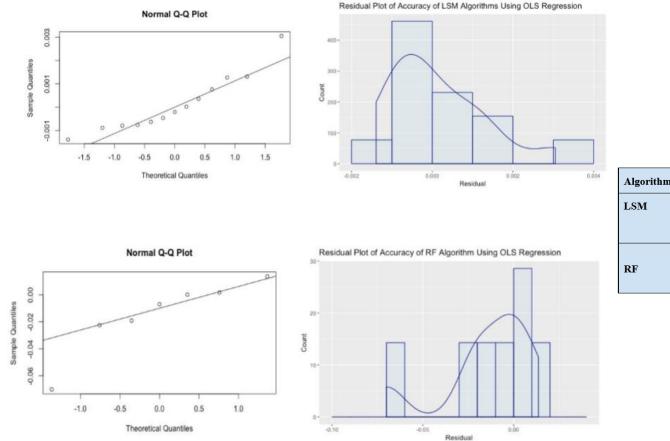


Ordinary Least Squares Regression outperforms Random Forest and Regression Tree Models in Predicting Response Time



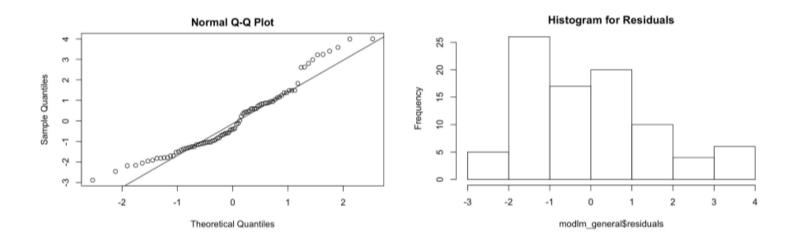
Model	Train R2	Test R2	Final Model
OLS	89%	91%	Yes
RF	70%	96%	No
Tree	80%	88%	No

Ordinary Least Squares Regression Models Provide highest Accuracy for LSM and RF Algorithms



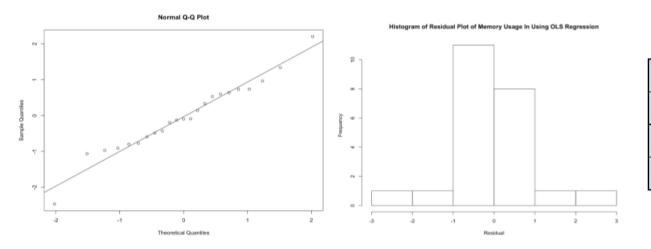
Algorithms	Model	Train R2	Test R2	Final Model
LSM	OLS	87%	84%	Yes
	RF	83%	95%	No
DE	OLS	99%	99%	Yes
RF	RF	48%	89%	No

RF Model to Predict CPU Utilization



Regression Tree model provides higher accuracy of CPU Utilization prediction (the difference between train and test results is smaller).

Regression Tree Models provide better accuracy in Memory Usage prediction



Model	Train R2	Test R2	Final Model
OLS	78%	87%	No
Tree	87%	81%	Yes
RF	60%	62%	No

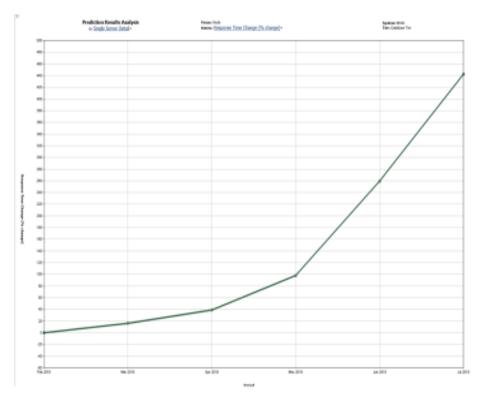
Regression Tree model provides better accuracy of Memory Usage prediction

Comparing algorithms:

Ordinary Least Squared provides better results in prediction of Accuracy and Response Time. Regression Tree algorithm provides better results in predicting CPU and Memory utilization.

Measure	Algorithms	Model	Train R2	Test R2
Accuracy	LSM	OLS	87%	84%
	RF	OLS	99%	99%
Response Time	LSM	OLS	89%	91%
	RF	OLS	73%	85%
Memory Usage	LSM & RF	Tree	87%	81%
CPU Utilization	LSM & RF	Tree	97%	96%

Predicted impact of the #Observations increase by 25% each period on OLS elapsed time

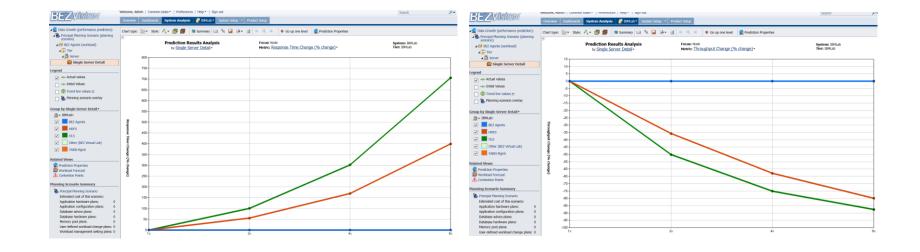


Waiting for I/O will become a bottleneck for OLS

Predicted Elapsed Time Components for Data Set Size Growth 25% per step

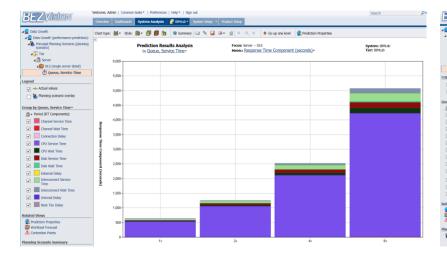
Unia Ind Double (parlaments)	inerne Mr. Inie 📾	r 🗗 🗰 🎠 🗮 henney 🗔	N 🖬 (H) 📓 H (H) (• • incernation 😫	taktor insperior					
Annual Parsing Lonard (daming annual)	In the native lacitors changed to Temperous Face Emprovementy Temperously Temperou									
- 10 mm	Profession Results Analysis Press for a contrast to Dates, Sorton Trans- Press Regions, Trac Concerned.				entrust e Time Component (becamb	a-		Spallers I	HOL day Tar	
Concern Simple server Setted	634		-							
ad a second s	122									
+ total store	47									
. Farming constitution										
	1.00									
In Geren, Inveloa Time 1	1.54									
Perind (KT Consumption) Council Service Texa	640							_		
	10									
	1.00							_		
								_		
CRU Half Terre End Decision Terre	1.54									
End Daring Term	6.0									
	10							_		
External Entry International Actions	1.40							_		
Take	1.45							_		
Intercent Kat Text								_		
Internal Indep	10									
Not far tiday	1 10									
el Viren	1 10									
raktu nuutu	0									
Partial Invent.	1.00									
Contention Prints	1									
ing formario formany	1.0									
Hinduit Planning Domarks	k									
Edwarded and of the constraints Application frankmant plants: 1	12									
lapitote-ordente-ofen 1	1.00									
Intelescenting plans: 1	622									
Naturbase fund-source practice of a Hearmony good planes: 4	10				_	_				
mar Advant variated charge place: 1	10									
Wolford recognised and by plants if										
d Randward Jammery	6.60									
uni 🤃 000	8.54			_						
House and the second	6.0									
004.0000.0000.000	87									
seffermatory units	1.00									
These Bullets					-					
Register Test Component: The average free greet processing the southed surgesters.	100- 100-									
		Aug 324	Her 2018	Apr 2010		No 201	26.324		x/20	

Predicted Elapsed Time and Throughput Change (%) for 50, 100, 200 and 400 Predictors



Predicted Elapsed Time Components OLS and HDFS for for 50, 100, 200 and 400 Predictors

OLS

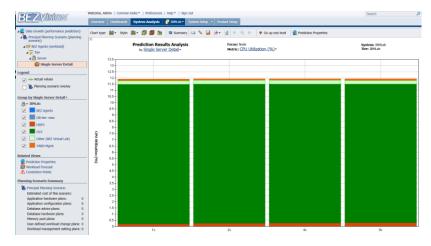


HDFS

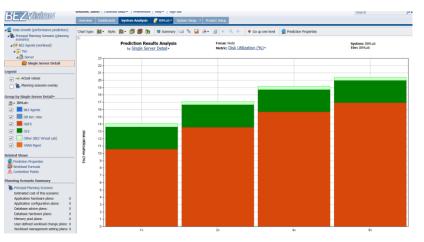
BEZVISION		is* Preferences Help* Sign out			Search	۶÷
	Overview Dashboards 5	ystem Analysis 🛛 🚰 18MLab 👻 System Setu	p 👻 Product Setup			
Data Growth Data Growth (performance prediction): A Principal Planning Scenario (planning	Chart type: M = Style: M			p one level 👷 Prediction Properties		
scenario)	w me metric has been	sanged to Response Time Component. 10	was previously response time change,	. Due that metric is not available in this contex	6	
A D Server		rediction Results Analysis	Focus: Server HDFS		System: IBMLab	
AB HDFS (single server detail)	· ·	by Queue, Service Time -	Metric: Response Time (Component (seconds)+	Tier: IBMLab	
Queue, Service Time						
egend	20,000					
🖌 🔶 Actual values	19,000 -					
Planning scenario overlay	18,000					
	17,000					
iroup by Queue, Service Time *	16,000					
- Period (RT Components):						
Channel Service Time	15,000 -					
Channel Wait Time	14,000 -					
Connection Delay	8 13,000					
CPU Service Time	g 12,000					
CPU Wait Time	B 11.000					
Disk Service Time						
Disk Wait Time	10,000					
External Delay Interconnect Service	8 9,000					
Time	8,000					
 Interconnect Wait Time 	(se 7,000 -					
 Internal Delay 	B 6.000					
Next Tier Delay	5,000					
telated Views	4.000					
Rediction Properties						
Workload Forecast	3,000					
	2,000 -					
lanning Scenario Summary	1,000-					
Principal Planning Scenario Estimated cost of this scenario: Acceleration body and acceleration	۰.	1×	źx	áx.	Øx	

Predicted Resource Utilization for OLS and HDFS for 50, 100, 200 and 400 Predictors

CPU

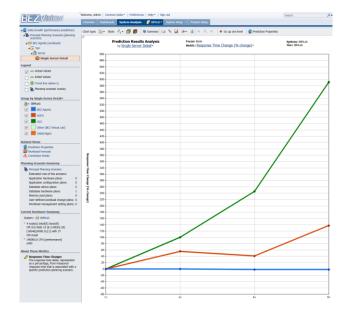


Disk

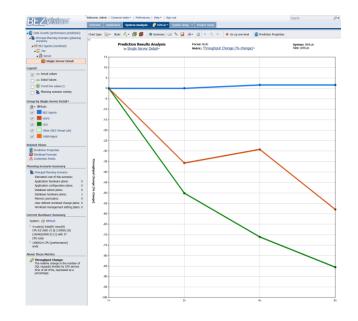


Predicted Impact of Increase Number of Nodes on OLS and HDFS Workloads Elapsed Time and Throughput

Elapsed Time relative change



Throughput relative change



Predicted Impact of Increase Number of Nodes on OLS and HDFS Elapsed Time Components

OLS

BEZVISION	Welcome, Admin Commo Overview Dashboard	on tasks • Preferences Help	• Sign out 4.45 • System Setup • Prode			Search
Data Growth			turnary 🖂 🍬 🖬 🕞-			
Data Growth (performance prediction):	Chart type: . Styk	· · · · · · · · · · · ·	ummary 🖾 🧐 🖬 🦉 •	21 · · · · · · · · · · · · · · · · · · ·	ne sevel 1 Prediction Pro	spanes
Principal Planning Scenario (planning	a la constante de la constante		ne Component." It was previou			
scenario)	W The metric has t	icen changed to mesponse in	ne component. ² It was previou	all	NAT INSURVE IS NOT AVAILABLE	in this context.
A D Server						
CLS (single server detail)		rediction Results Analy ty Queue, Service Time-		ever – CLS Response Time Component	(seconds) -	System: 201.ab Ter: 1090.ab
Queue, Service Time		ty gatal, all the line		supprise time comparent.	farmer and -	
pend	5,000					
😥 🔶 Actual values	4.500					
	4,000-					
Planning scenario overlay	4,500					
oup by Queue, Service Time *						
- Period (IET Components):	4,400					
Channel Service Time						
Channel Watt Time	4,200					
Connection Delay	4.000					
CPU Service Time						
CPU Wat Time	3,800-					
Dek Service Time						
Dek Wolt Time	3,600-					
External Delay	3,400					
	0,400					
Time	3,200					
 Interconnect wait Time 						
Triternal Delay	3,000-					
Next Tier Delay	2,800-					
lated Views	8 2,000					
Prediction Properties	2,500-					
Workload Porecast	i i					
Contention Points	2,000 2,000					
nning Scenario Summary	2,200-					
Principal Planning Scenario	8					
Estimated cost of this scenario:	2,000-					
Application hardware plans: 0						
Application configuration plans: 0 Database advice plans: 0	1,800-					
Database hardware plans: 1	1.600					
Hemory pool plans: 0	1,000					
User-defined workload change plans: 0 Workload management setting plans: 0	1,400-					
rent Hardware Summary	1,200-					
iyatemi 😝 IIIMLab	1.000					
4 node(s) Intel(R) Xeon(R) CPU E5-2680 v3 (8 2.00GHz (8)						
(16046)(4000.0) [1] with 37 CPU total	800-					
148000.0-CPU [performance] units	600-		-			
ut These Metrics	400-					
Response Time Component: The average time sperit processing the	200-					
specified component.						
	0.1	Ťx.	2×		4x	6×

HDFS

BEZVISION	Overview	Selfcords	System An	alysis	🚰 IEMLab 🕶	System Se	tup • Produc	Setup				
Cets Growth	Chart type:	4- syle:	B- (F)	0 b	Summa	ny 🖙 🎕	🖬 🕞 - 🗟		🔶 Go up one les	el 🧕 Prediction	Properties	
Data Growth (performance prediction): A Principal Planning Scenario (planning	4											
scenario)	🍁 The m	stric has be	en changed to	a 'Respo	nse Time Co	reponent."	t was previous	ly 'Response T	ime Change,' but the	t metric is not ava	illable in this context.	
A GR Tier												
a 💆 Server			ediction R					ver HDPS			System: IBHLab	
HDFS (single server detail)			by Queue, S	iervice."	Time -		Metric: 🖯	isponse Tim	e Component (seci	ands) •	Tiert 18MLab	
C Queue, Service Time	Ι.	.500										
ogond		400										
🖵 🔶 Actual values		200										
Renning scenario overlay		,000										
		,800										
Sroup by Queue, Service Time*		,500										
Period (RT Components):		,400										
🕑 📕 Channel Service Time		,200										
Charmel Well Time		,000										
Connection Delay		500										
CRU Service Time		400										
CPU Wat Time		200										
Disk Service Time		.000										
Cisk Wat Time		,800										
		,500										
External Delay		,400									- 10 C	
Time Interconnect Service		,200										
 Interconnect Wait Time 		,000										
Internal Delay		,800									- 10 C	
Not Ter Delay		,600										
	lă 1	,400										
Related Views		,200										
Prediction Properties		.000										
Workload Forecast Contention Points		100										
	13 .	400										
Planning Scenario Summary	А,	,200										
Principal Planning Scenario	(seconds)	,000										
Estimated cost of this scenario: Application herdware plans: 0	1 A 1	,800										
Application configuration plans: 0		,600										
Database advice plans: 0		A00-										
Database hardware plans: 1		,200-										
Memory pool plans: 0 User-defined workload change plans: 0		1,000 -										
Workload management setting plans: 0		800-										
		400-										
Current Hardware Summary		200-										
System: 😝 25MLab		,000-										
 4 node(s) Intel(R) Xeon(R) CPU E5-2683 v3 (8 2.00GHz (8) 	1	,000-			_							
(16046)(4000.0) [1] with 37		,000-				-						
CPU total		,400-										
 \$48000.0 CPU [performance] units 		,200-										
	1	,000-										
About These Metrics		800-										
Response Time Component: The average time spent processing the		400-										
specified component.		200-										
		~1										

Example of Business Requirements

 The Score takes into consideration the type of ML algorithm, Number of Observations and Features / Predictors in Data Set, the relative importance of the different criteria, like response time, Accuracy, CPU Utilization, Memory utilization, Number of I/O operations, and other parameters:

Score = w1 * Accuracy + w2 * Response Time + w3 * CPU Utilization + w4 * Memory Utilization +w5 * Scalability , etc

Where the weighting coefficients wi represent business priorities between 0 and 1.

Type of Requirement	Relative
	Weight
Accuracy	0.2
Response Time	0.4
CPU Utilization	0.2
Memory Utilization	0.2
Total	1

Example of Recommendation

- Response Time can vary between 0 and infinity. We transform the response time as 1 / (1 + RT) to make it as a number between 0 and 1, where 1 is better. In addition to calculating the score we check if predicted CPU Utilization and Memory Usage are less than 1. -
- Value of score is used to recommend the appropriate ML algorithm and ML Library.

Algorithm	library	pred_score	pred_rank	true_score	true_rank	
OLS	Python Sklearn	0.962057911	1	0.936165261	1	$\langle __$
OLS	Pyspark ML	0.876712666	2	0.753752225	2	
Ridge	Python Sklearn	0.781980143	3	0.725268522	3	
Ridge	Pyspark ML	0.722426161	4	0.659234146	4	
RF	Python Sklearn	0.476284999	5	0.429752013	5	
RF	Pyspark ML	0.465422159	6	0.415271967	6	

• ML OLS Algorithm using Python Sklearn ML library is the most appropriate algorithm to satisfy business requirements presented in example above.

Summary

- Business Requirements
- Challenges
- Collaboration
- Modeling Expands Benchmarks
- ML Models and QNM Models
- Surprises
- Recommender



References

- Daniel A. Menasce "Software, Performance, or Engineering?" WOSP '02 Proceedings of the 3rd international workshop on Software and performance Pages 239-242
- Max Kuhn, Kjell Johnson, Applied Predictive Modeling, Springer, 2013
- B. Zibitsker, IEEE Conference, Delft, Netherlands, March 2016, Big Data Performance Assurance
- B. Zibitsker, Key note presentation "Role of Big Data Predictive Analytics" Big Data Predictive Analytics Conference, Minsk 2015
- B. Zibitsker, Key Note Presentation on "Big Data Advanced Analytics", Big Data Advanced Analytics Conference, Minsk 2016,
- Dominique A. Heger "Big Data Predictive Analytics, Applications, Algorithms and Cluster Systems" ISBN:978-1-61422-951-3
- Dr. Joseph Hellerstein, *Microsoft Corp*, Yixin Diao, *IBM* Engineering Performance Using Control Theory
- B. Zibitsker, Proactive Performance Management for Data Warehouses with Mixed Workload, Teradata Partners, 2008, 2009
- J. Buzen, B. Zibitsker, CMG 2006, "Challenges of Performance Prediction in Multi-tier Parallel Processing Environments"
- B. Zibitsker, CMG 2008, 2009 "Hands on Workshops on Performance Prediction for Virtualized Multi-tier Distributed Environments"
- Heger, D., "Introduction to Apache YARN Schedulers & Queues", Data Analytica, www.mlanalytica.com, 2016
- Heger, D. "Machine Learning in the Realm of Big Data Analytics", Fundcraft Publication, ISBN 978-0-578-19095-2, March 2017.

THANK YOU!