MATLAB EXPO 2018

Tackling Big Data Using MATLAB

Alka Nair
Application Engineer

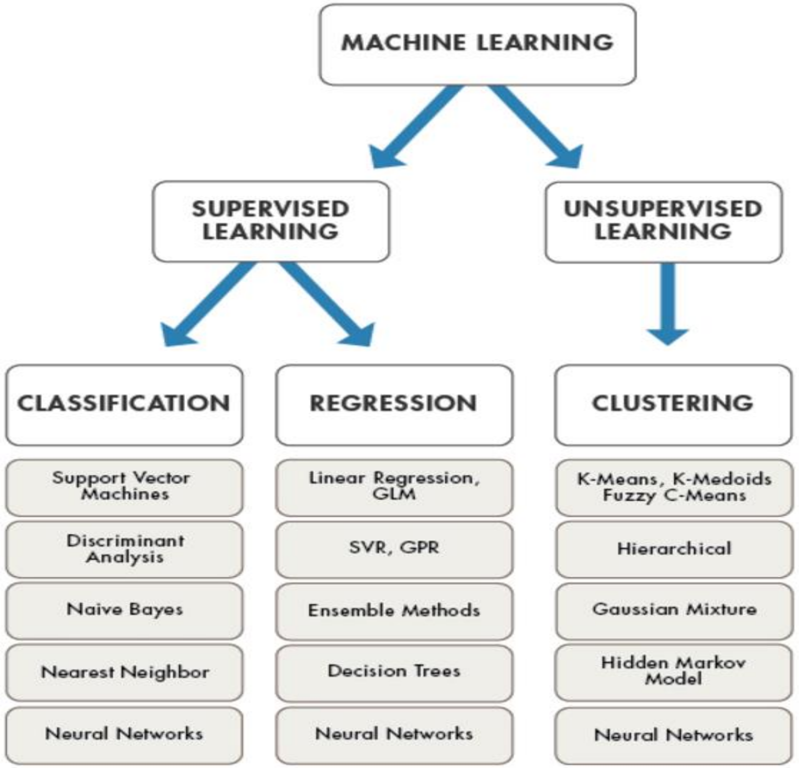
Building Machine Learning Models with Big Data

Access

Preprocess, Exploration & Model Development

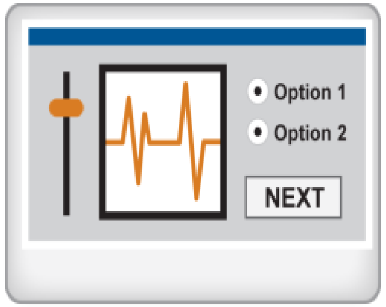
Scale up & Integrate with Production Systems

hadoop

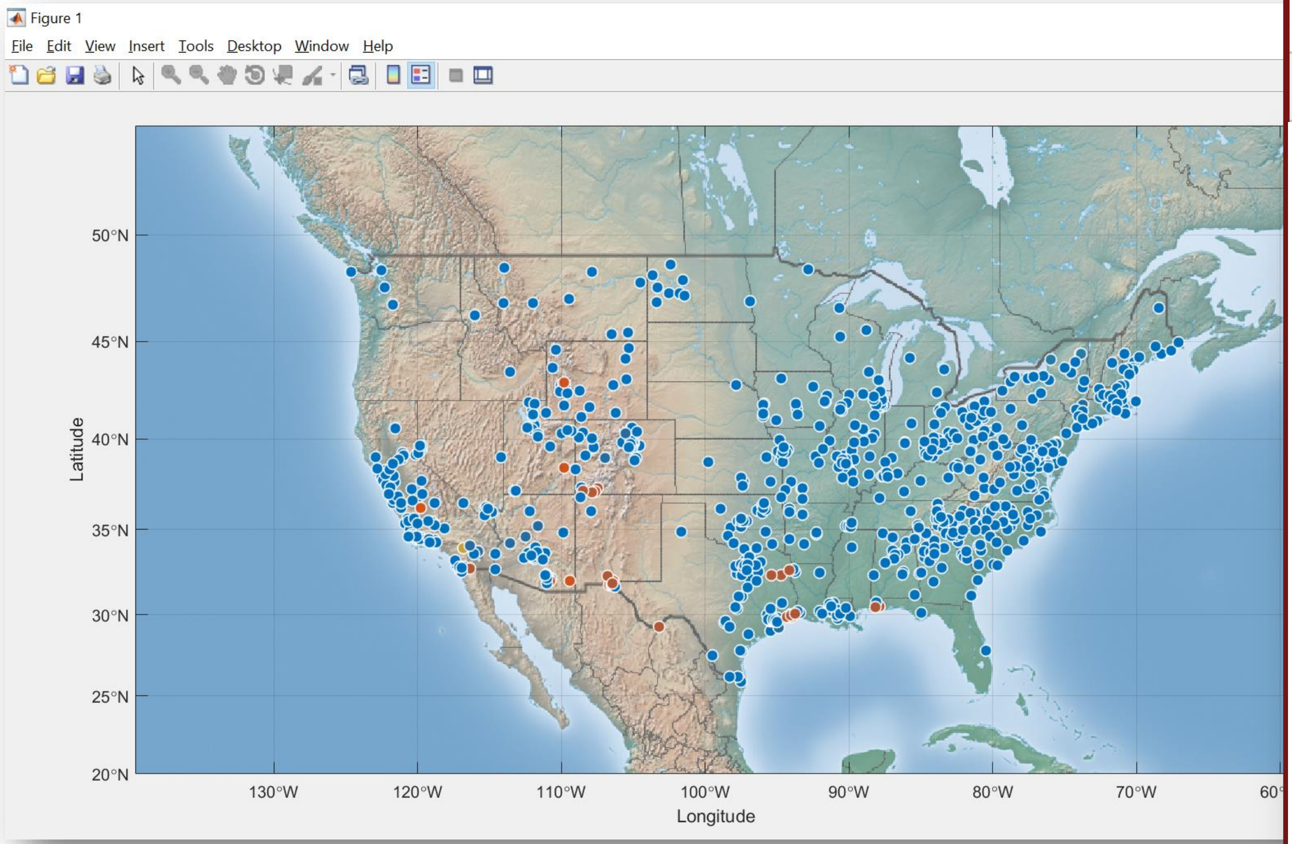


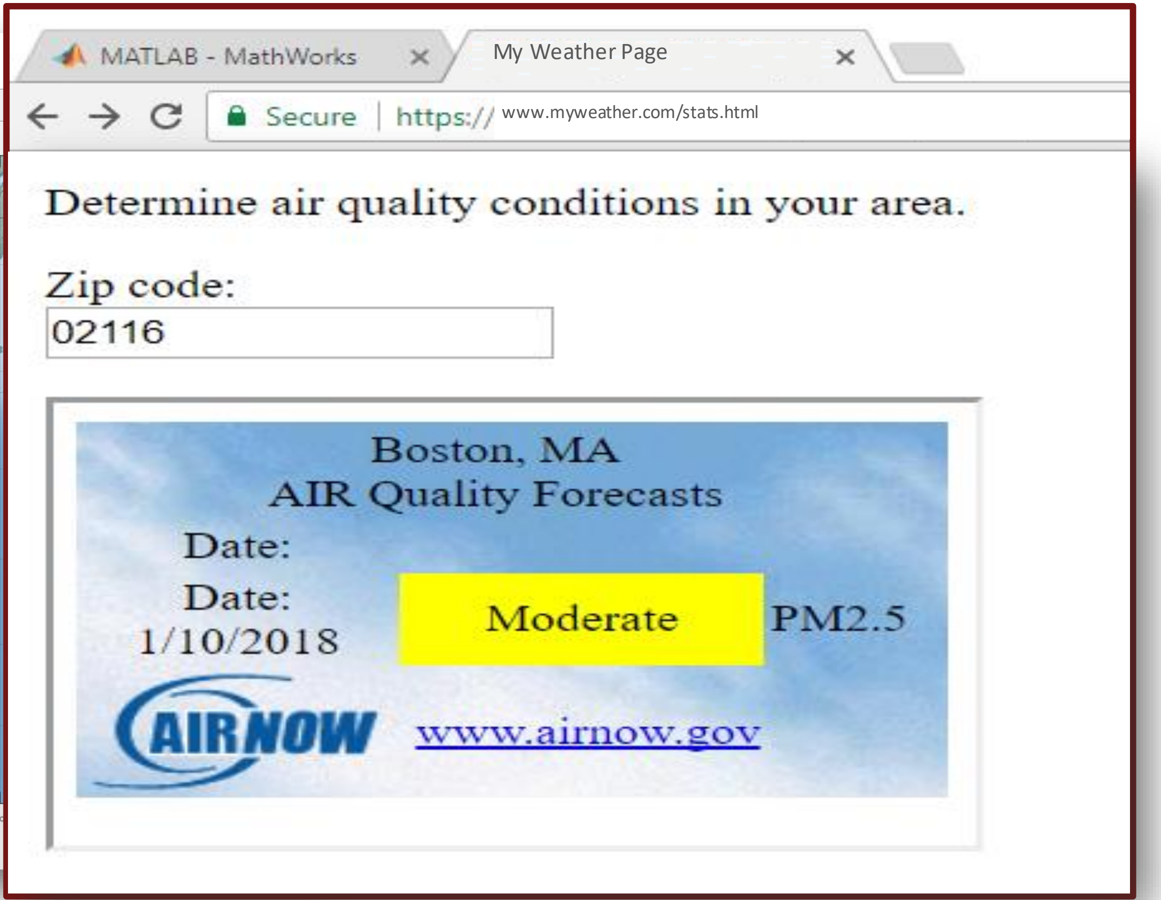
Spark

MATLAB Excel .NET .exe C/C++ Java .dll



Case study: Predict Air Quality





Hadoop Overview Datanodes Snapshot Startup Progress Utilities

1TB Data

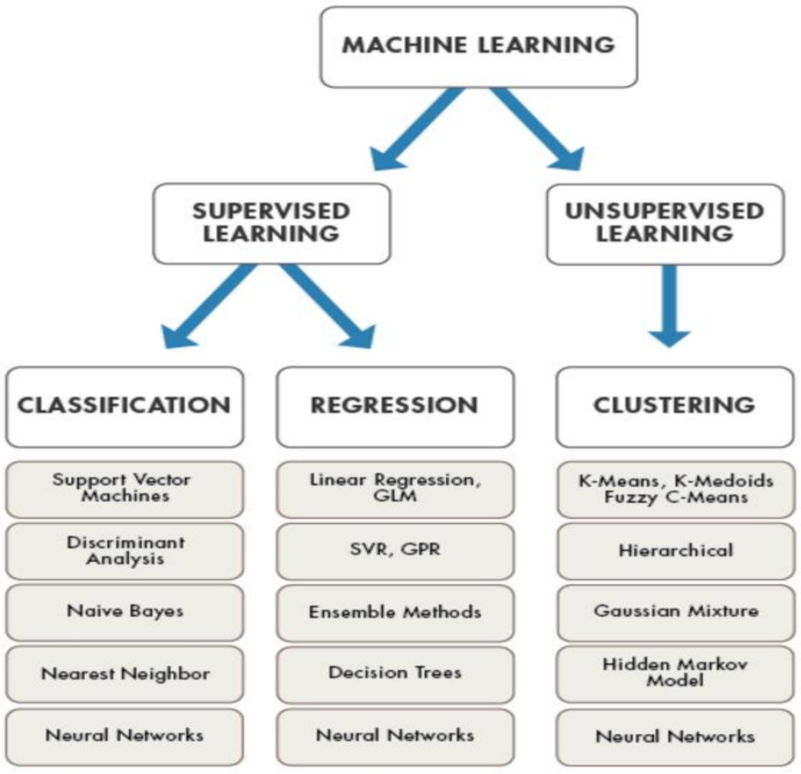
Browse Directory

/datasets/AirQuality/hourlyData Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hgorr	supergroup	673.54 MB	7/5/2017, 2:25:08 PM	3	128 MB	hourly_42101_1980.csv
-rw-r--r--	hgorr	supergroup	738.79 MB	7/5/2017, 2:25:29 PM	3	128 MB	hourly_42101_1981.csv
-rw-r--r--	hgorr	supergroup	799.69 MB	7/5/2017, 2:25:55 PM	3	128 MB	hourly_42101_1982.csv
-rw-r--r--	hgorr	supergroup	806.46 MB	7/5/2017, 2:27:24 PM	3	128 MB	hourly_42101_1983.csv
-rw-r--r--	hgorr	supergroup	788.51 MB	7/5/2017, 2:28:19 PM	3	128 MB	hourly_42101_1984.csv

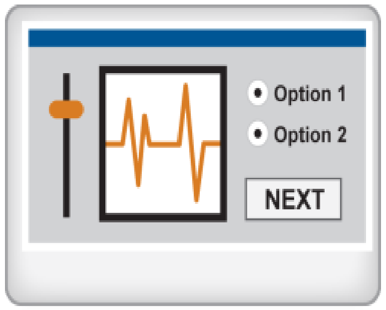
Building Machine Learning Models with Big Data

Access → **Preprocess, Exploration & Model Development** → **Scale up & Integrate with Production Systems**



Spark

MATLAB Excel .NET .exe C/C++ Java .dll



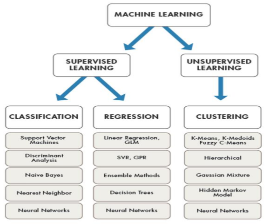
Challenges in Modeling and Deploying Big Data Applications

Access

- Distributed Data Storage
- Different Data Sources & Types

Managing Different APIs for Data Sources and Data Formats

Preprocess, Exploration & Model Development



- Preprocessing and Visualizing Big Data
- Parallelizing Jobs and Scaling up Computations to Cluster

- Rewriting Algorithms to Use Big Data Platforms
- Parallelizing Code to Scale up to Use Cluster and Cloud Compute

Scale up & Integrate with Production Systems

- Enterprise level deployment

Overhead in Moving the Algorithm to Production

Wouldn’t it be nice if you could:

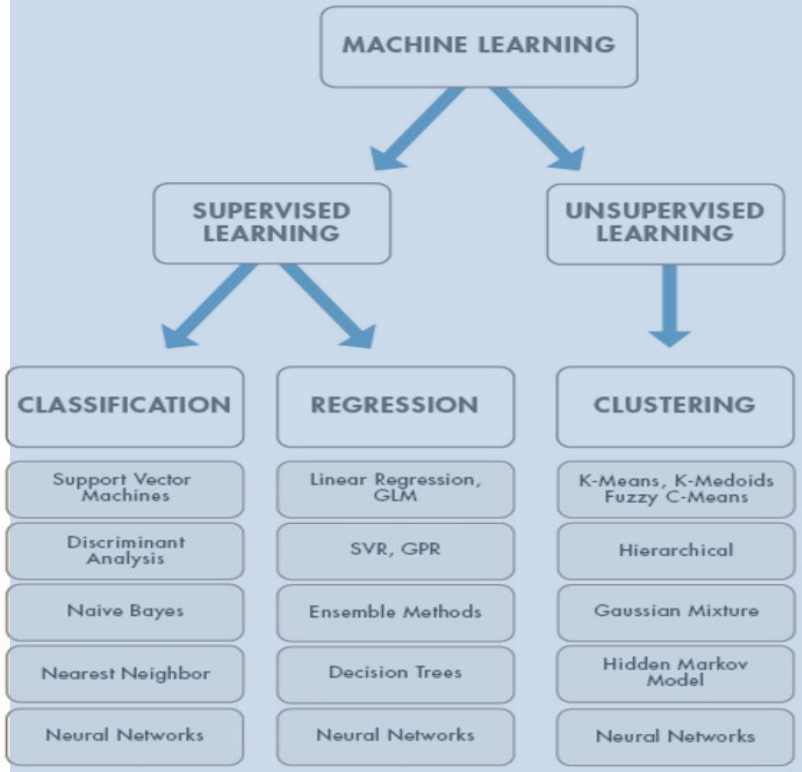
- Easily access data however it is stored
- Prototype algorithms quickly using small data sets
- Scale up to big data sets running on large clusters
- **Using the same intuitive MATLAB syntax you are used to**

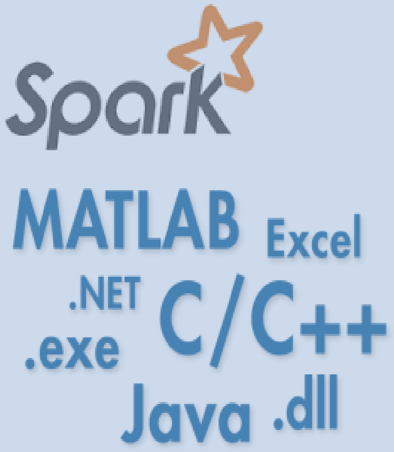
Building machine learning models with big data

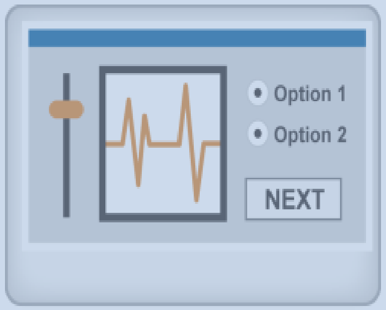
Access

Preprocess, Exploration & Model Development

Scale up & Integrate with Production Systems







Access and Manage Big Data

Different Data Types

- Text
- Images
- Spreadsheet
- Custom File Formats

Different Data Sources

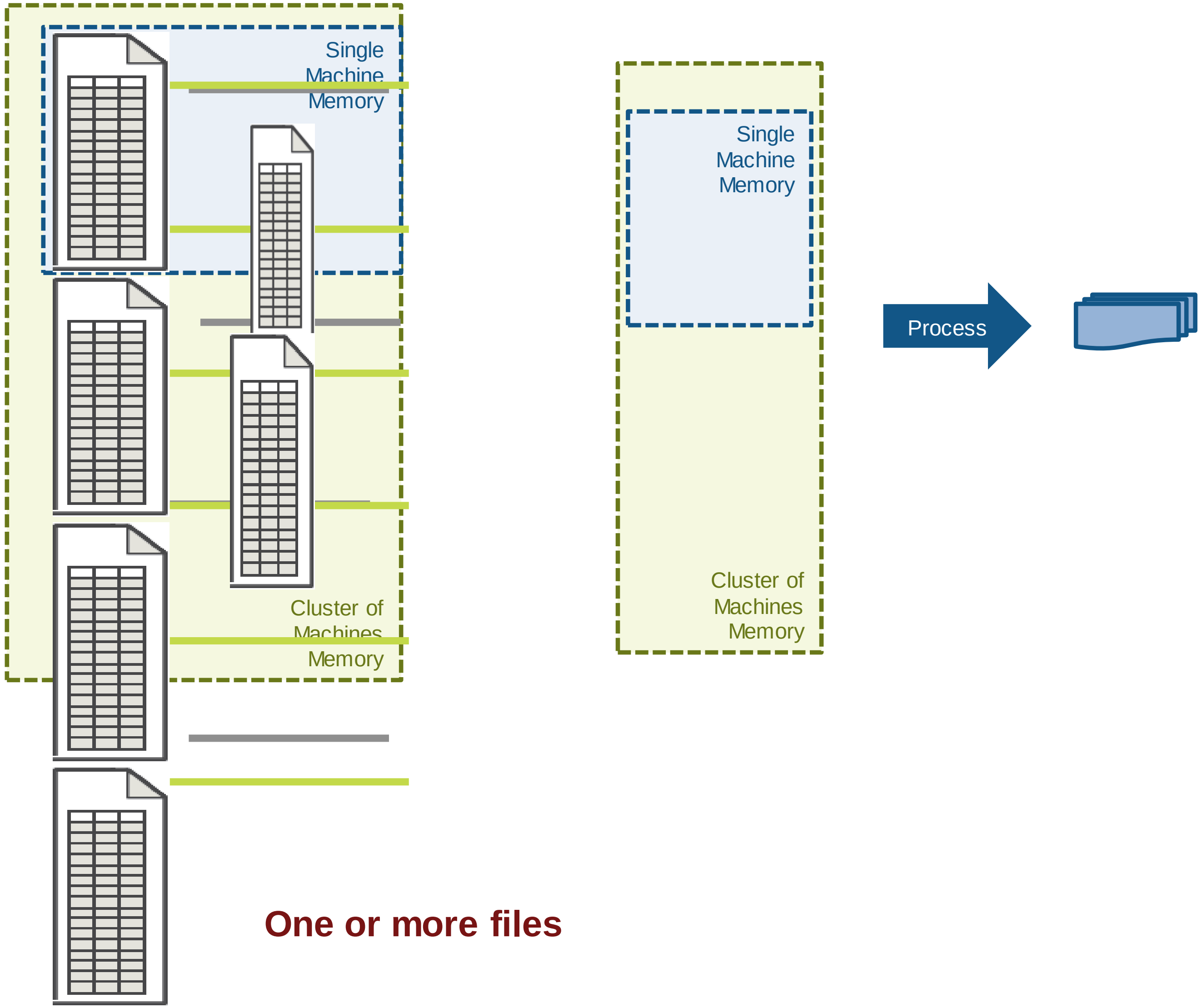
- Hadoop Distributed File System (HDFS)
- Amazon S3
- Windows Azure Blob Storage
- Relational Database
- HDFS on Hortonworks or Cloudera

Different Applications

- MapReduce
- Image Segmentation
- Image Classification
- Denoising Images
- Predictive Maintenance

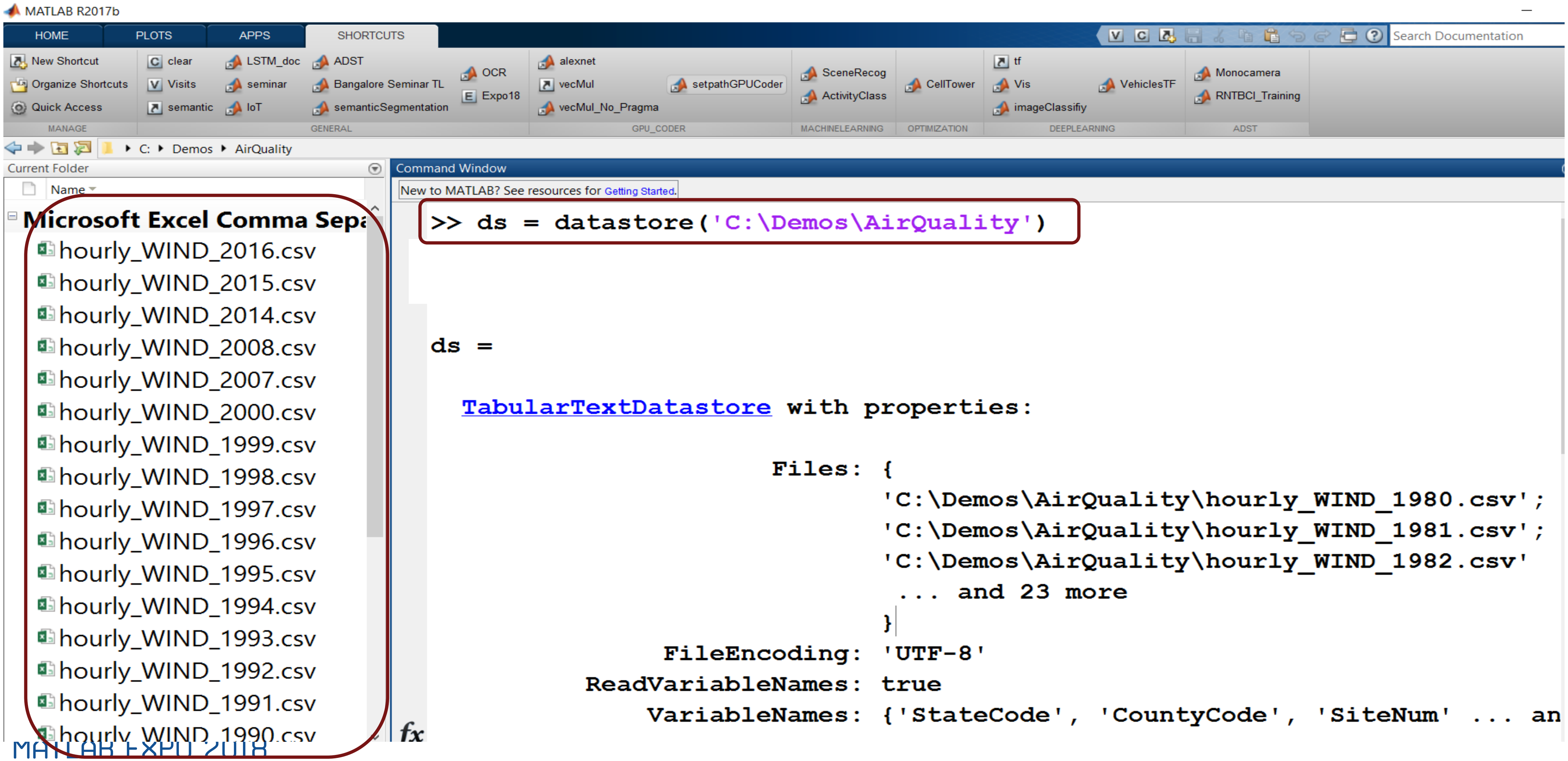
Datastores

Datastore



Single
Machine
Memory
Cluster of
Machines
Memory
Single
Machine
Memory
Cluster of
Machines
Memory
Process
One or more files

Air Quality Data on Local Folder



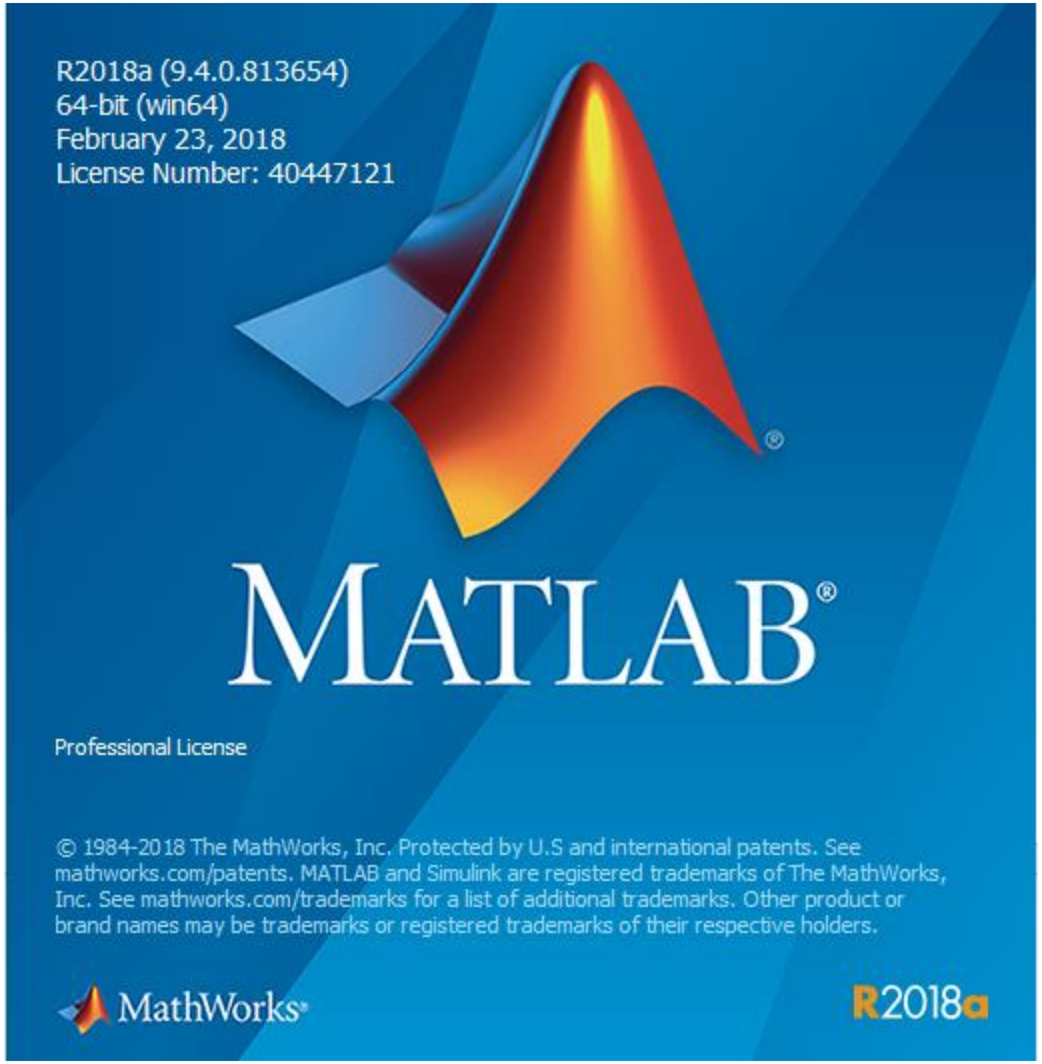
Accessing and Processing different types of data

Image Collection

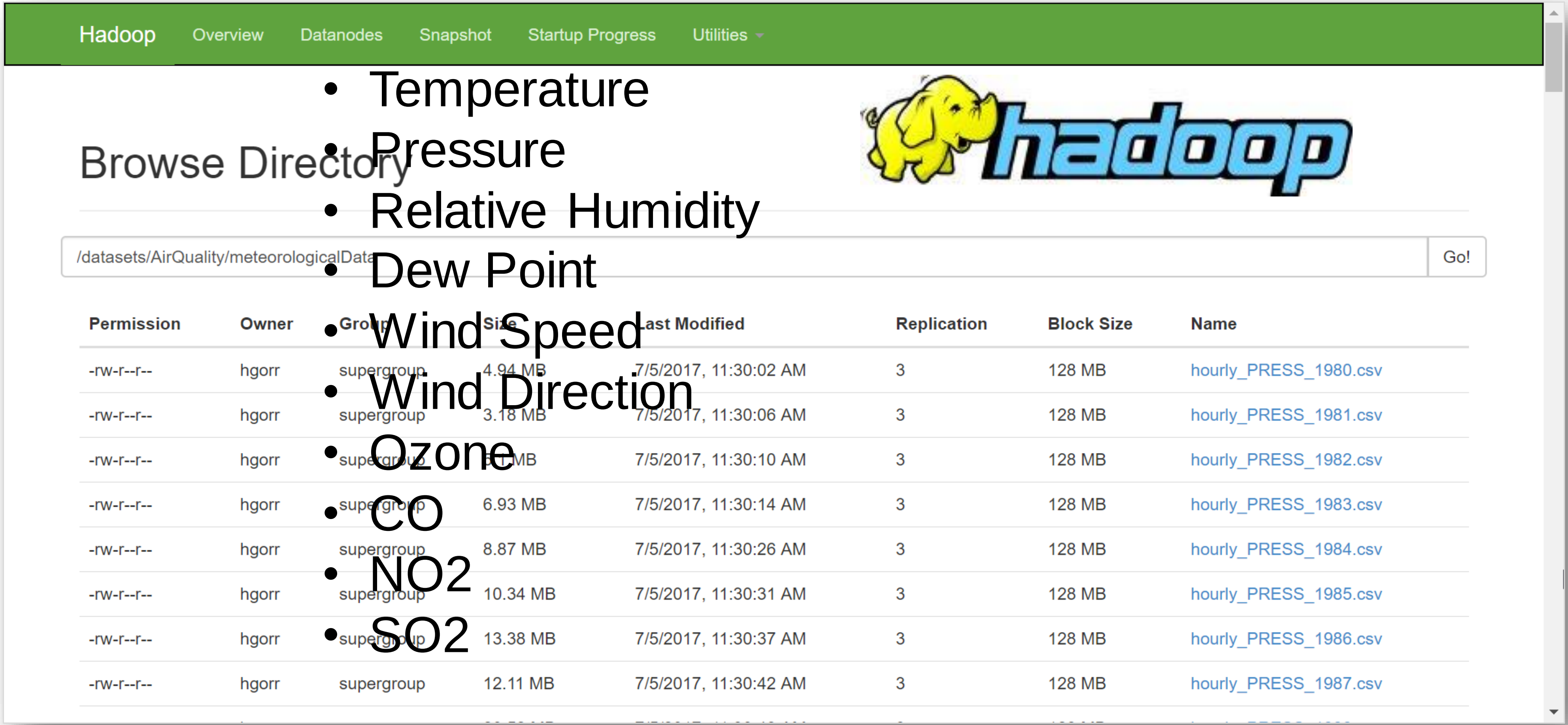
MDF Files

TabularTextDatastore	Text files containing column-oriented data, including CSV files
ImageDatastore	Image files, including formats that are supported by imread such as JPEG and PNG
SpreadsheetDatastore	Spreadsheet files with a supported Excel® format such as .xlsx
MDFDatastore	Datastore for collection of MDF files
Custom Datastore	Datastore for custom or proprietary format

You have 1 TB of data you've never seen before. How do you access this data?



Historical files are on HDFS and real time data are available through an API



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hgorr	supergroup	4.94 MB	7/5/2017, 11:30:02 AM	3	128 MB	hourly_PRESS_1980.csv
-rw-r--r--	hgorr	supergroup	3.18 MB	7/5/2017, 11:30:06 AM	3	128 MB	hourly_PRESS_1981.csv
-rw-r--r--	hgorr	supergroup	[illegible]	7/5/2017, 11:30:10 AM	3	128 MB	hourly_PRESS_1982.csv
-rw-r--r--	hgorr	supergroup	6.93 MB	7/5/2017, 11:30:14 AM	3	128 MB	hourly_PRESS_1983.csv
-rw-r--r--	hgorr	supergroup	8.87 MB	7/5/2017, 11:30:26 AM	3	128 MB	hourly_PRESS_1984.csv
-rw-r--r--	hgorr	supergroup	10.34 MB	7/5/2017, 11:30:31 AM	3	128 MB	hourly_PRESS_1985.csv
-rw-r--r--	hgorr	supergroup	13.38 MB	7/5/2017, 11:30:37 AM	3	128 MB	hourly_PRESS_1986.csv
-rw-r--r--	hgorr	supergroup	12.11 MB	7/5/2017, 11:30:42 AM	3	128 MB	hourly_PRESS_1987.csv

- Temperature
- Pressure
- Relative Humidity
- Dew Point
- Wind Speed
- Wind Direction
- Ozone
- CO
- NO2
- SO2

Access air quality data using `datastore`

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse Directory

/datasets/AirQuality/dailyData

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hgorr	supergroup	80.79 MB	9/10/2017, 5:45:44 AM	3	128 MB	daily_42101_1980.csv
-rw-r--r--	hgorr	supergroup	88.24 MB	9/10/2017, 5:45:55 AM	3	128 MB	daily_42101_1981.csv
-rw-r--r--	hgorr	supergroup	95.82 MB	9/10/2017, 5:46:07 AM	3	128 MB	daily_42101_1982.csv
-rw-r--r--	hgorr	supergroup	96.62 MB	9/10/2017, 5:46:16 AM	3	128 MB	daily_42101_1983.csv
-rw-r--r--	hgorr	supergroup	94.21 MB	9/10/2017, 5:46:28 AM	3	128 MB	daily_42101_1984.csv
-rw-r--r--	hgorr	supergroup	93.5 MB	9/10/2017, 5:46:41 AM	3	128 MB	daily_42101_1985.csv
-rw-r--r--	hgorr	supergroup	90.62 MB	9/10/2017, 5:46:51 AM	3	128 MB	daily_42101_1986.csv
-rw-r--r--	hgorr	supergroup	95.16 MB	9/10/2017, 5:47:04 AM	3	128 MB	daily_42101_1987.csv

```
files = 'hdfs://hadoop01glnxa64:54310/datasets/AirQuality/daily_44201_*.csv';
ds5 = datastore(files,'TextType','string');
```

Preview the data and adjust properties to best represent the data of interest

```
ds.SelectedVariableNames = vars;
```

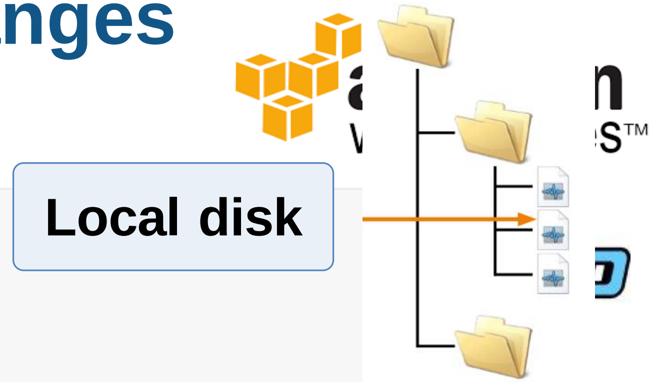
```
preview(ds)
```

```
ans =  8×6 table
```

	DateLocal	UnitsOfMeasure	ArithmeticMean	AQI	StateName	CountyName
1	"1980-04-04"	Parts per m…	0.0475	"67"	Alabama	Autauga
2	"1980-04-05"	Parts per m…	0.0366	"67"	Alabama	Autauga
3	"1980-04-06"	Parts per m…	0.0558	"84"	Alabama	Autauga
4	"1980-04-07"	Parts per m…	0.0439	"61"	Alabama	Autauga
5	"1980-04-08"	Parts per m…	0.0442	"49"	Alabama	Autauga
6	"1980-04-09"	Parts per m…	0.0428	"58"	Alabama	Autauga
7	"1980-04-10"	Parts per m…	0.0340	"67"	Alabama	Autauga
8	"1980-04-11"	Parts per m…	0.0416	"49"	Alabama	Autauga

Access data from anywhere with minimal changes

```
setenv('AWS_ACCESS_KEY_ID', 'ACCESS_KEY_ID')
setenv('AWS_SECRET_ACCESS_KEY', 'ACCESS_KEY')
```




```
fileLoc = 'datasets/FoodImages';
```

```
ds = imageDatastore(fileLoc);
```

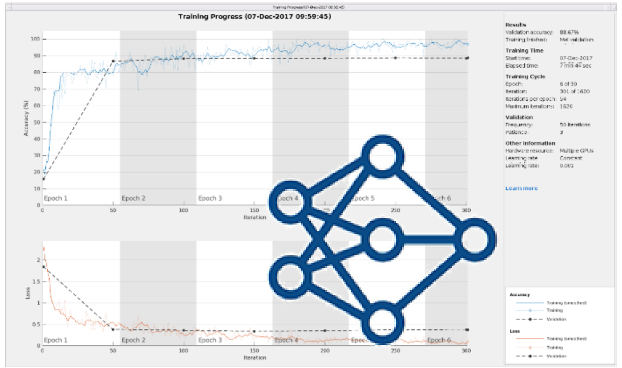
Datastores enable big data workflows

Deep Learning

```
ds = imageDatastore(fileLoc);
```

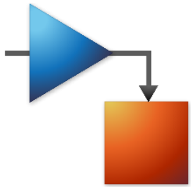
```
[trainDS,valDS,testDS] = splitEachLabel(ds,...
      0.7,0.15,0.15,'randomized');
```

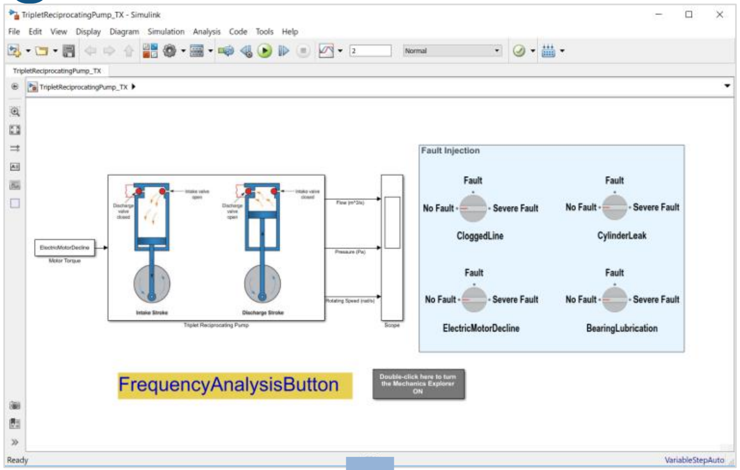
```
net = trainNetwork(trainDS,layers,trainOpts);
```



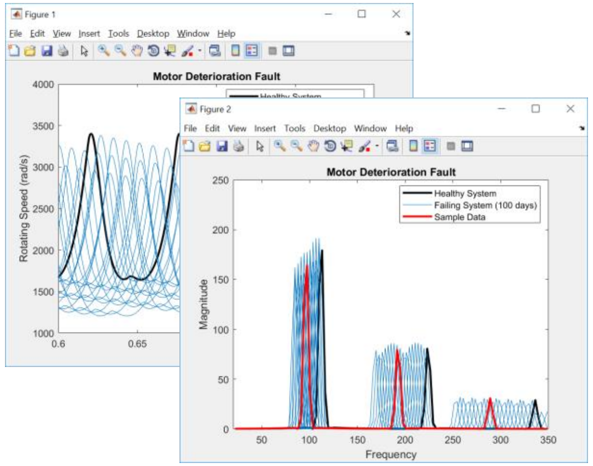
Datastores enable big data workflows

Predictive Maintenance





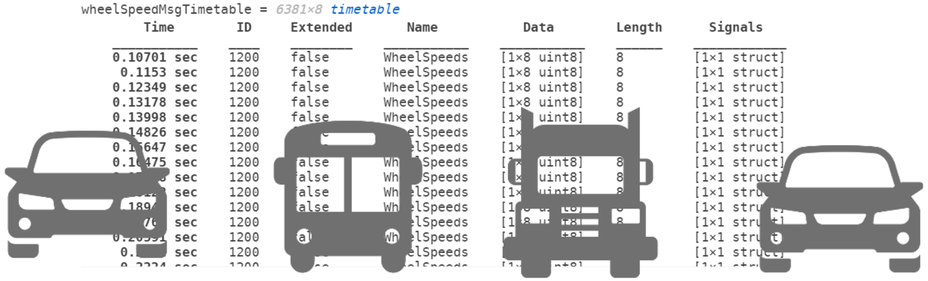

```
ds = simulationEnsembleDatastore(location)
```



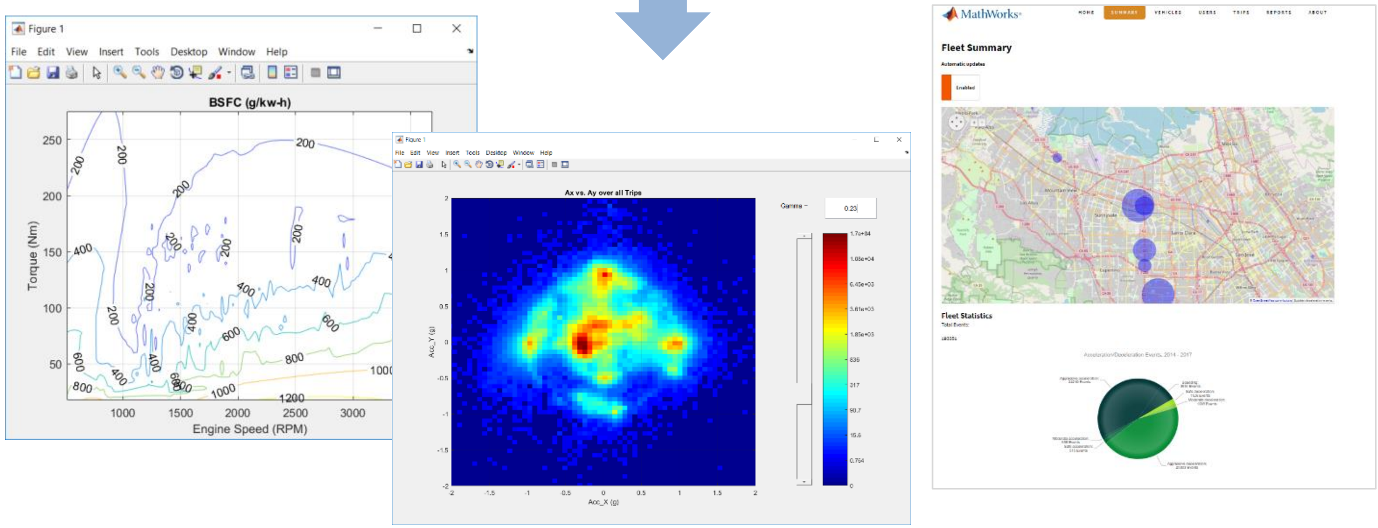

```
Days to Failure = 33.728 days
```

Datastores enable big data workflows

Fleet Analytics




```
ds = mdfDatastore(fileLoc);
```



Datastores: Access Big Data with Minimal Changes

Different Data Types

- Text
- Images
- Spreadsheet
- Custom File Formats

✓

Different Data Sources

- Hadoop Distributed File System (HDFS)
- Amazon S3
- Windows Azure Blob Storage
- Relational Database
- HDFS on Hortonworks or Cloudera

✓

Different Applications

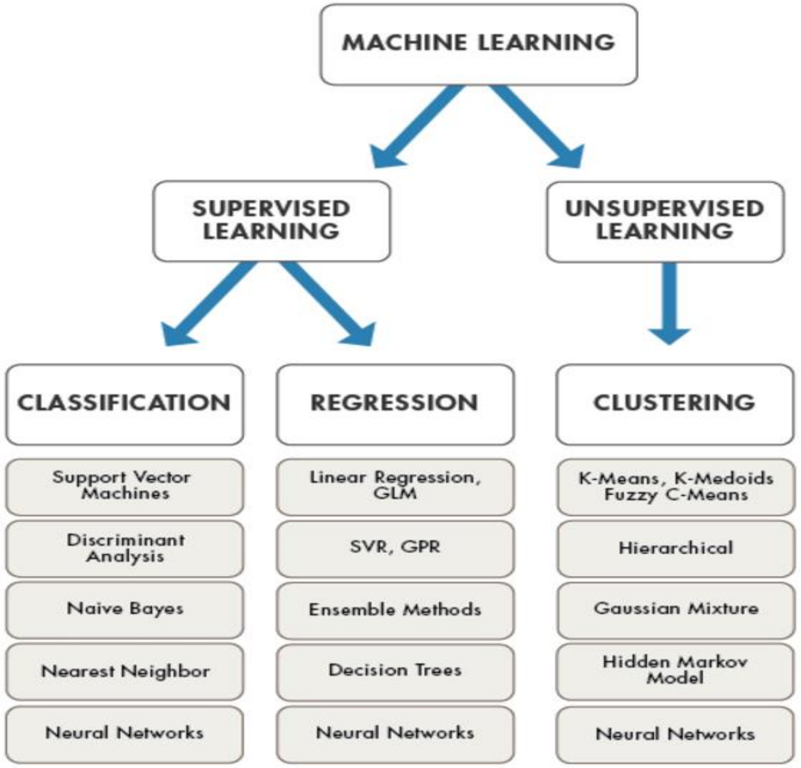
- MapReduce
- Image Segmentation
- Image Classification
- Denoising Images
- Predictive Maintenance

✓

Building machine learning models with big data

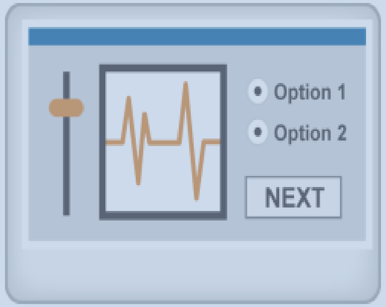
Access

Preprocess, Exploration & Model Development

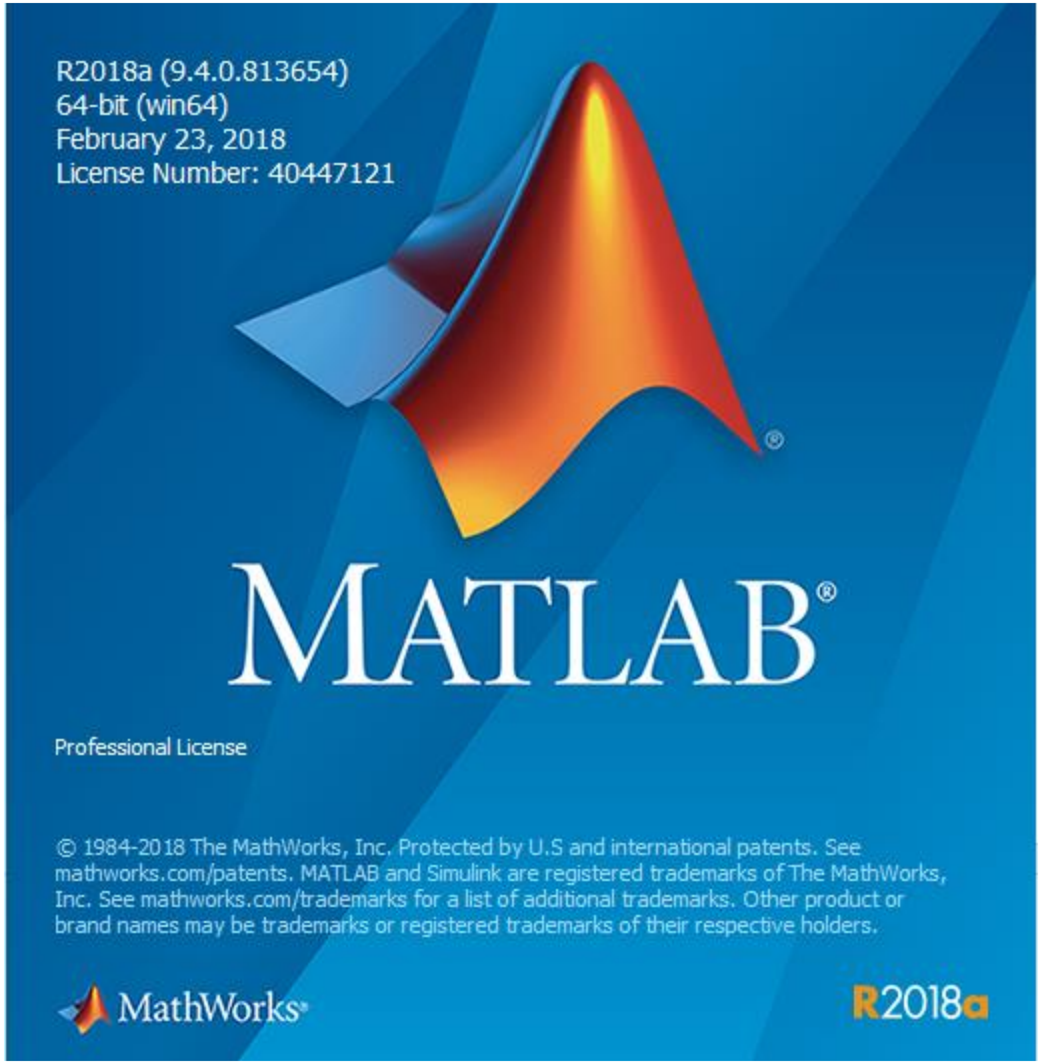


Scale up & Integrate with Production Systems

Spark
MATLAB Excel
.NET
.exe C/C++
Java .dll



You have 1TB of data you've never seen before. How do you visualize and process the data?



Use `tall` arrays to work with the data like any MATLAB array

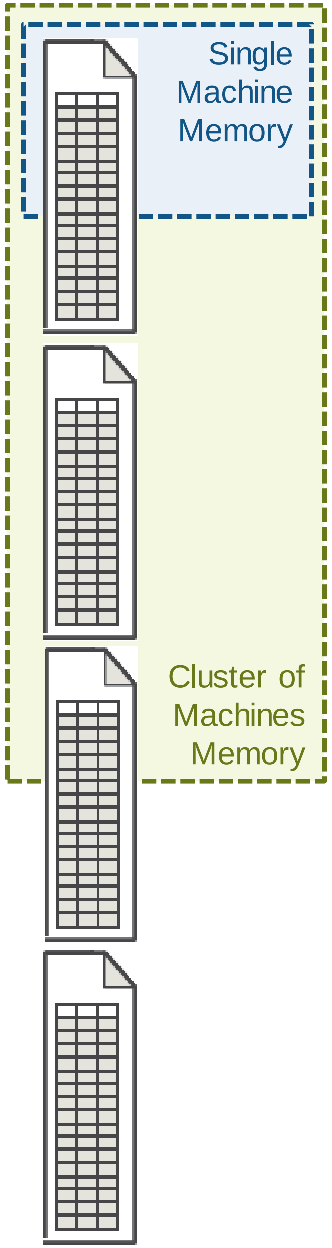
- **Introduction to Tall Arrays**
- **Tall Arrays for Big Data Visualization and Preprocessing**
- **Machine Learning for Big Data Using Tall Arrays**

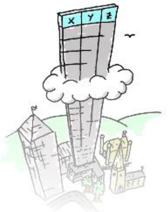
Tall arrays

- Data is in one or more files
- Files stacked vertically
- Typically tabular data

Challenge

- Data doesn't fit into memory
  (even cluster memory)
- Takes a lot of time for even simple
  operations on data





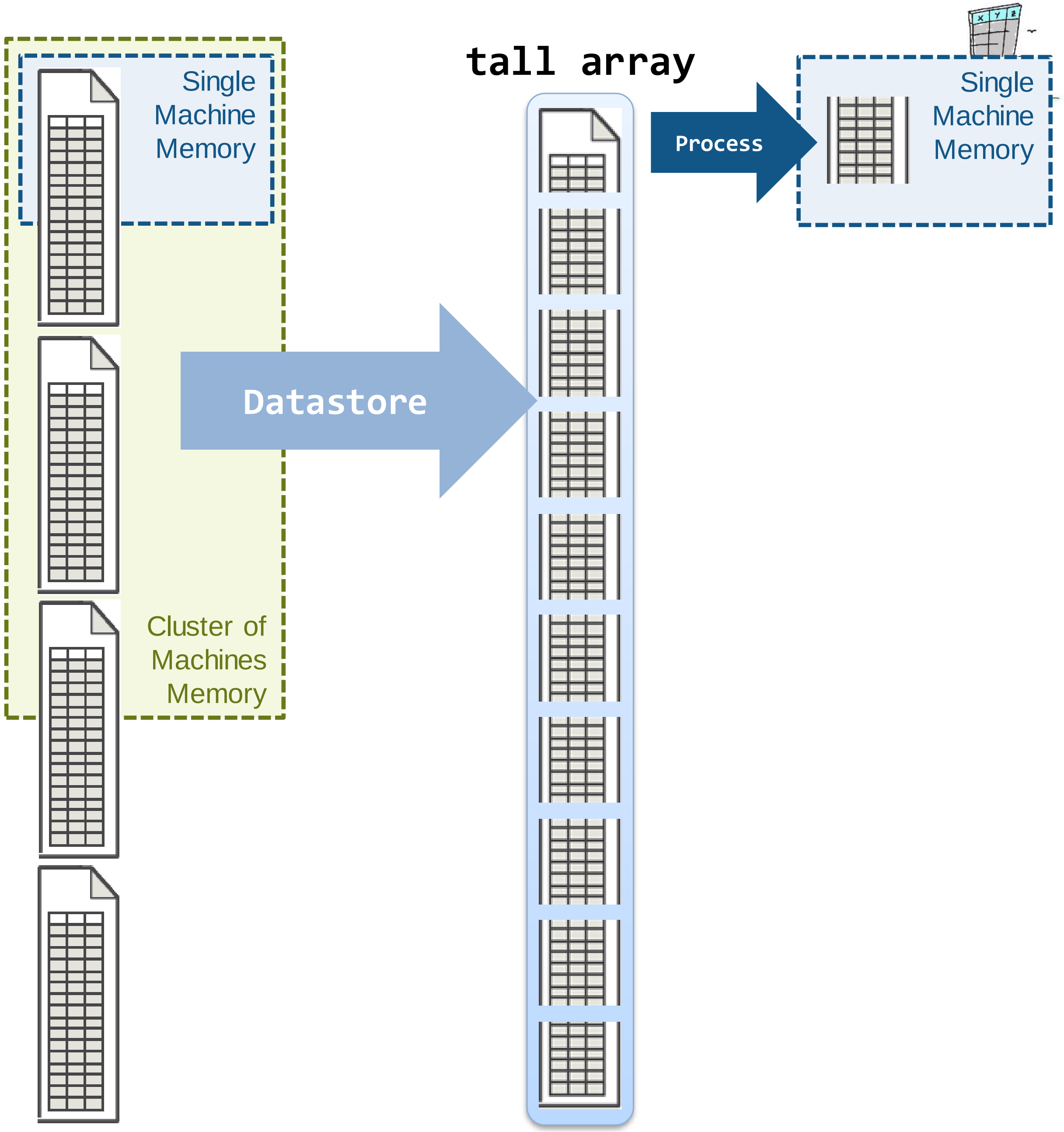
Tall arrays (new R2016b)

- Create tall table from datastore

```
ds = datastore('*.csv')
tt = tall(ds)
```

- Operate on whole tall table
  just like ordinary table

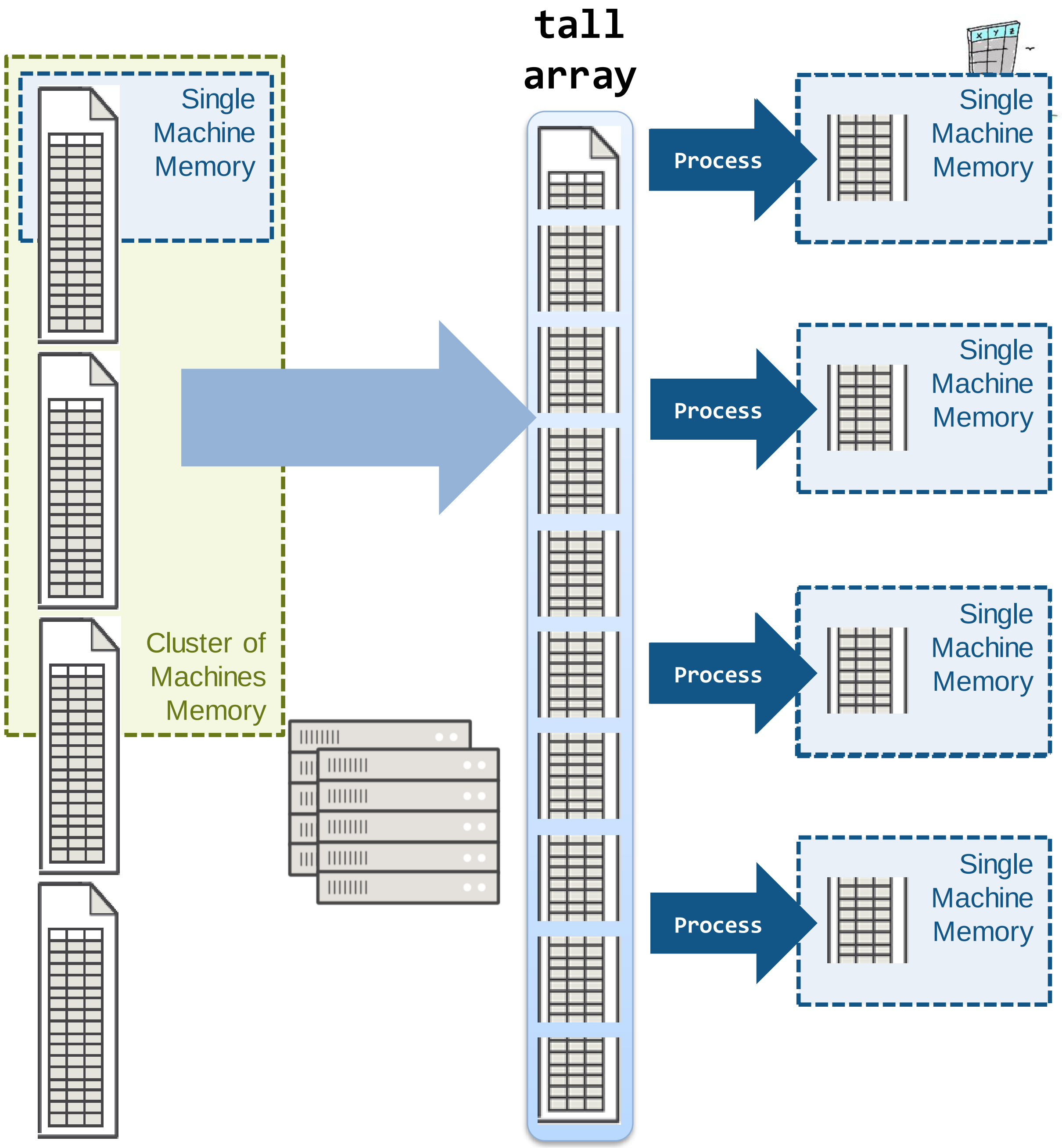
```
summary(tt)

max(tt.EndTime - tt.StartTime)
```



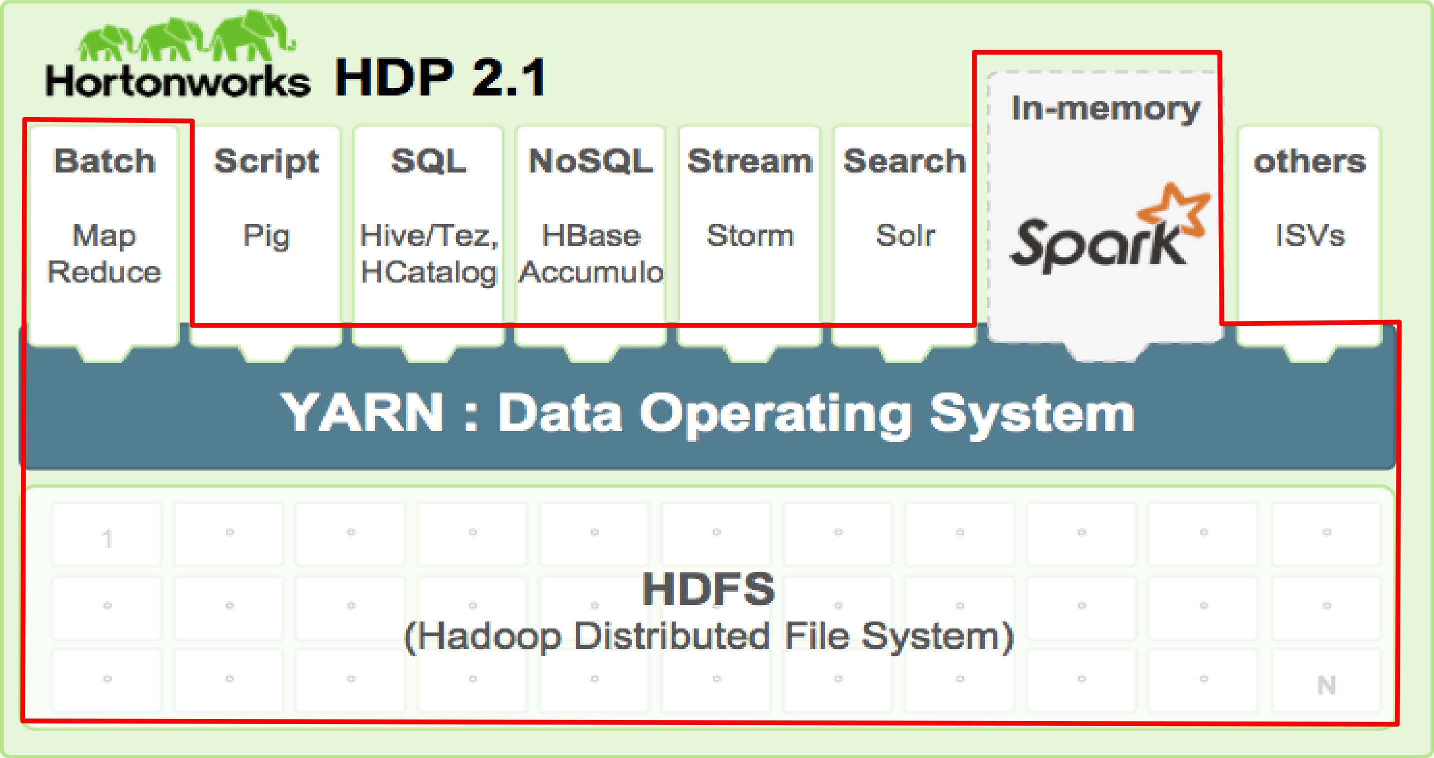
tall arrays R2016b

- With Parallel Computing Toolbox, process several “chunks” at once

- Can scale up to clusters with MATLAB Distributed Computing Server



Use a Spark-enabled Hadoop cluster and MATLAB



Support for many other platforms through reference architectures

It’s easy to run MATLAB code on Spark + Hadoop

MATLAB R2017b

Live Editor - C:\Demos\Demos\LTC\Visits\Expo2018Prep\HeatherSlides\AirQuality_Tall_17b.mlx

Set Environment to Spark - Enabled Hadoop Cluster

```
setenv('HADOOP_HOME','/mathworks/AH/hub/apps_PCT/LS_Hadoop_hadoop01glnxa64/current')
setenv('SPARK_HOME','/mathworks/hub/3rdparty/R2017a/1998143/share/spark/2.0.0-2.6/')

numWorkers = 32;
cluster = parallel.cluster.Hadoop;
cluster.SparkProperties('spark.executor.instances') = num2str(numWorkers);
mr = mapreducer(cluster);
```

Spark Connection

Cluster Config for Spark

Create datastore for data on HDFS.

```
files = 'hdfs://hadoop01glnxa64:54310/datasets/AirQuality/hourlyData/hourly_44201_*.csv'; % Ozone measurements
warning('off','MATLAB:table:ModifiedVarnames')
% files = ['data',filesep,'hourly_44201_2016.csv'];
ds = datastore(files,'TextType','string');
```

Hadoop Access

MATLAB Documentation for

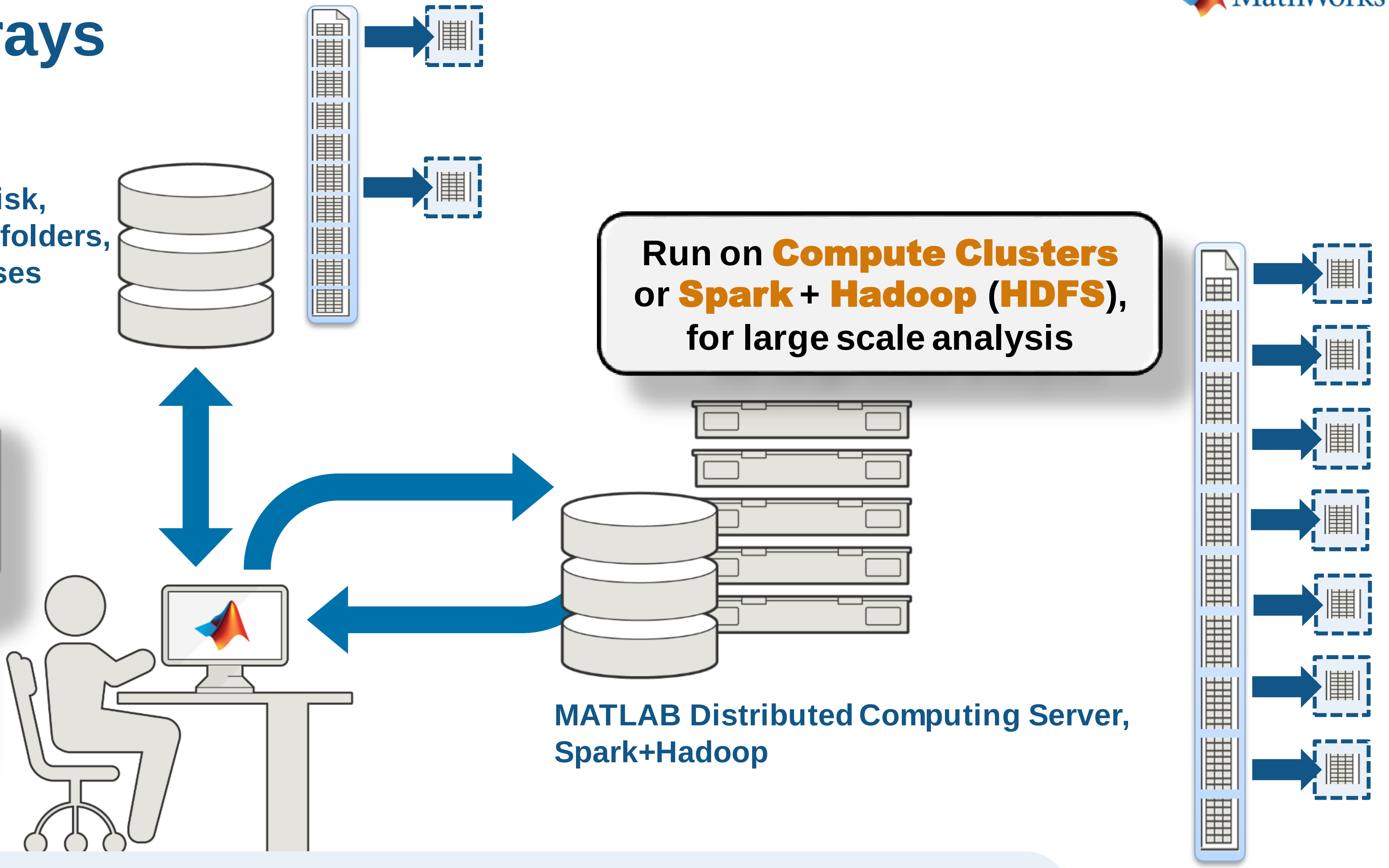
Build Effective Algorithms with MapReduce

Example Link	Primary File	Description	Notable Programming Techniques
Find Maximum Value with MapReduce	`MaxMapReduceExample.m`	Find maximum arrival delay	One intermediate key and minimal computation.
Compute Mean Value with MapReduce	`MeanMapReduceExample.m`	Find mean arrival delay	One intermediate key with intermediate state (accumulating intermediate sum and count).
Create Histograms Using MapReduce	`VisualizationMapReduceExample.m`	Visualize data using histograms	Low-volume summaries of data, sufficient to generate a graphic and gain preliminary insights.
Compute Mean by Group Using MapReduce	`MeanByGroupMapReduceExample.m`	Compute mean arrival delay for each day of the week	Perform simple computations on subgroups of input data using several intermediate keys.
Compute Maximum Average HSV of Images with MapReduce	`HueSaturationValueExample.m`	Determine average maximum hue, saturation, and brightness in an image collection	Analyzes an image datastore using three intermediate keys. The outputs are filenames, which can be used to view the images.
Simple Data Subsetting Using MapReduce	`SubsettingMapReduceExample.m`	Create single table from subset of large data set	Extraction of subset of large data set to look for patterns. The procedure is generali using a parameterized map function t in the subsetting criteria

Summary for `tall` arrays

Process out-of-memory data on your **Desktop** to explore, analyze, gain insights and to develop analytics

Use **Parallel Computing Toolbox** for increased performance



Develop your code locally using Tall Arrays or MapReduce only once

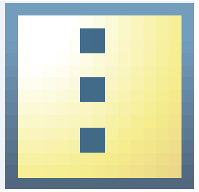
Use the same code to scale up to cluster

Create a `tall` array for each datastore

```
ozone = tall(ds)
```

```
Starting a Spark Job on the Hadoop cluster. This could take a few minutes ...done.
ozone =
  M×4 tall table
     DateLocal       ArithmeticMean    AQI      StateName
    ____________     ______________    ____     _________
    "1980-04-04"       0.0475          "67"     Alabama
    "1980-04-05"     0.036588          "67"     Alabama
    "1980-04-06"     0.055824          "84"     Alabama
    "1980-04-07"     0.043941          "61"     Alabama
    "1980-04-08"     0.044235          "49"     Alabama
    "1980-04-09"     0.042765          "58"     Alabama
    "1980-04-10"        0.034          "67"     Alabama
    "1980-04-11"     0.041647          "49"     Alabama
    :                :                 :        :
    :                :                 :        :
```

ozone



Execution model makes operations more efficient on big data

tt : tall array

```
a = tt.Month;
b = tt.DayofMonth;
c = mean(tt.DayofMonth);
d = std(tt.DayOfWeek);
e = numel(tt.AirTime);
f = tt.TaxiOut;
f(isnan(f)) = 0;
g = movmean(tt.ArrDelay,10);

calc3 = (a + b).*c + d.*f.*g;

calc3_result = gather(calc3);
```

- Deferred evaluation
  - Commands are not executed right away
  - Operations are added to a queue
- Execution triggers include:
  - `gather` function
  - `summary` function
  - Machine learning models
  - Plotting

Execution model makes operations more efficient on big data

```
a = tt.Month;
b = tt.DayofMonth;
c = mean(tt.DayofMonth);
d = std(tt.DayOfWeek);
e = numel(tt.AirTime);
f = tt.TaxiOut;
f(isnan(f)) = 0;
g = movmean(tt.ArrDelay,10);

calc3 = (a + b).*c + d.*f.*g;

calc3_result = gather(calc3);
```

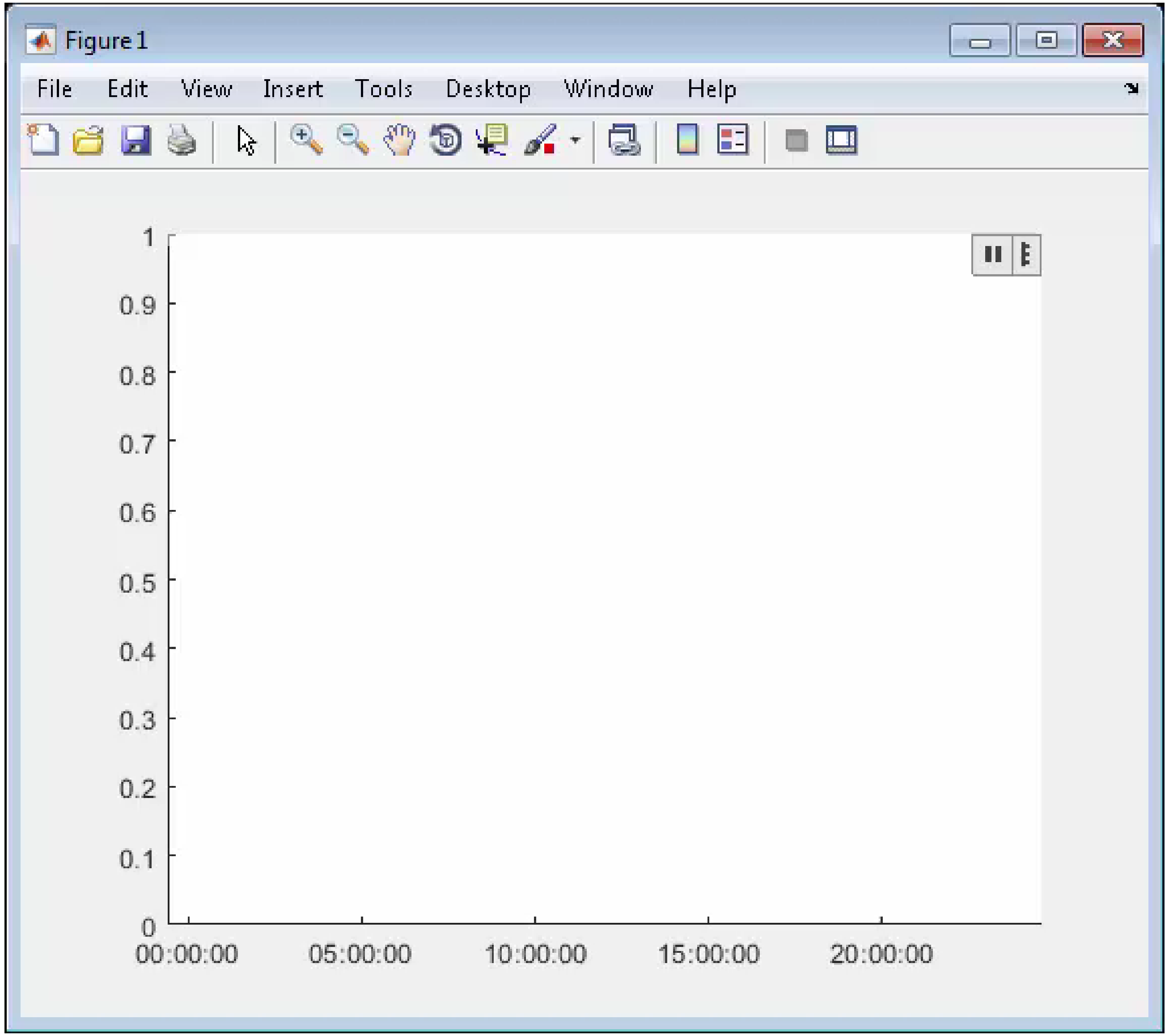
```
Evaluating tall expression using the Parallel Pool 'local':
- Pass 1 of 2: Completed in 3 sec
- Pass 2 of 2: Completed in 3 sec
Evaluation completed in 7 sec
```

```
e =
  tall double
    ?
Preview deferred. Learn more.
```

Unnecessary results are not computed

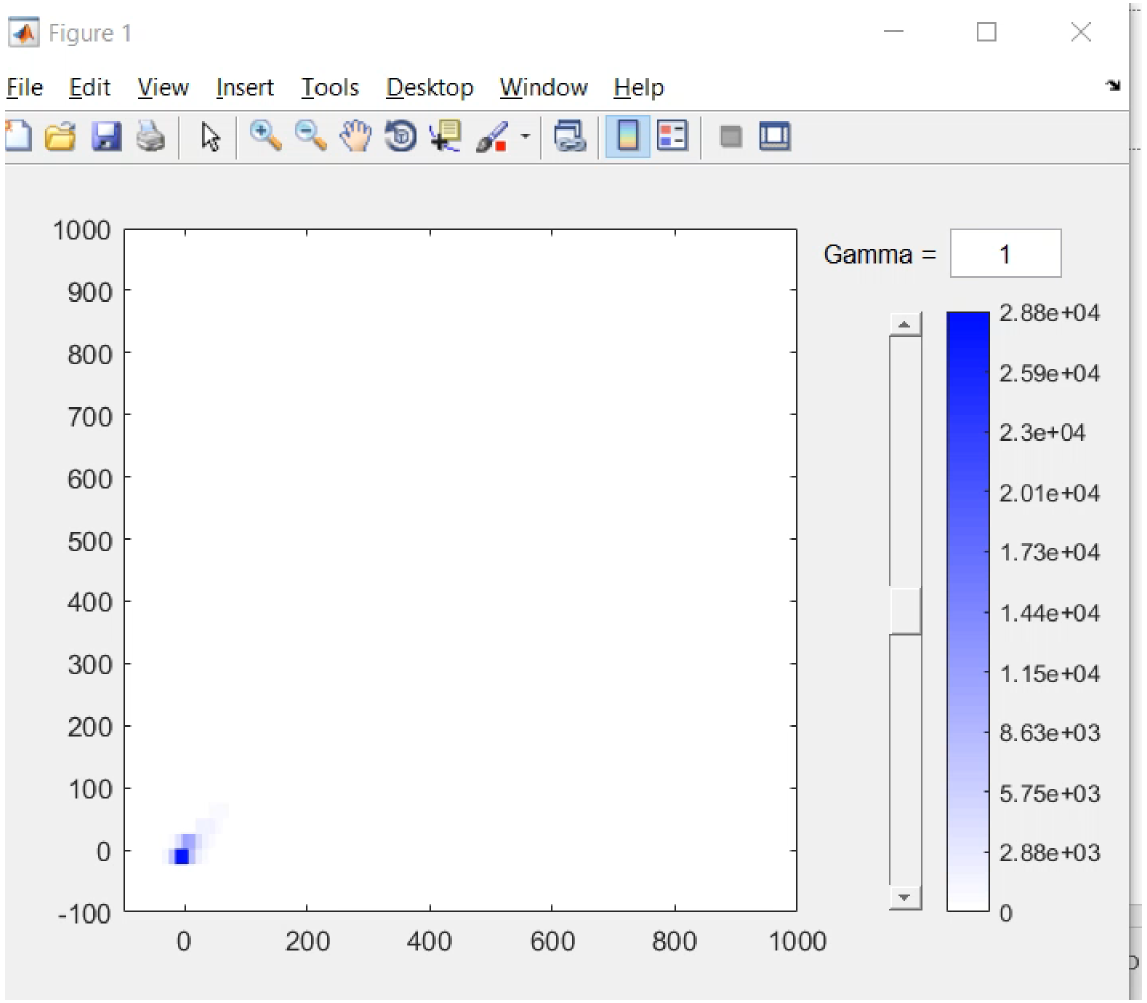
- ✓ **Introduction to Tall Arrays**
- **Tall Arrays for Big Data Visualization and Preprocessing**
- **Machine Learning for Big Data Using Tall Arrays**

Explore Big Data with Tall Visualizations




```
plot
scatter
binscatter
histogram
histogram2
ksdensity
```

Explore Big Data with Tall Visualizations

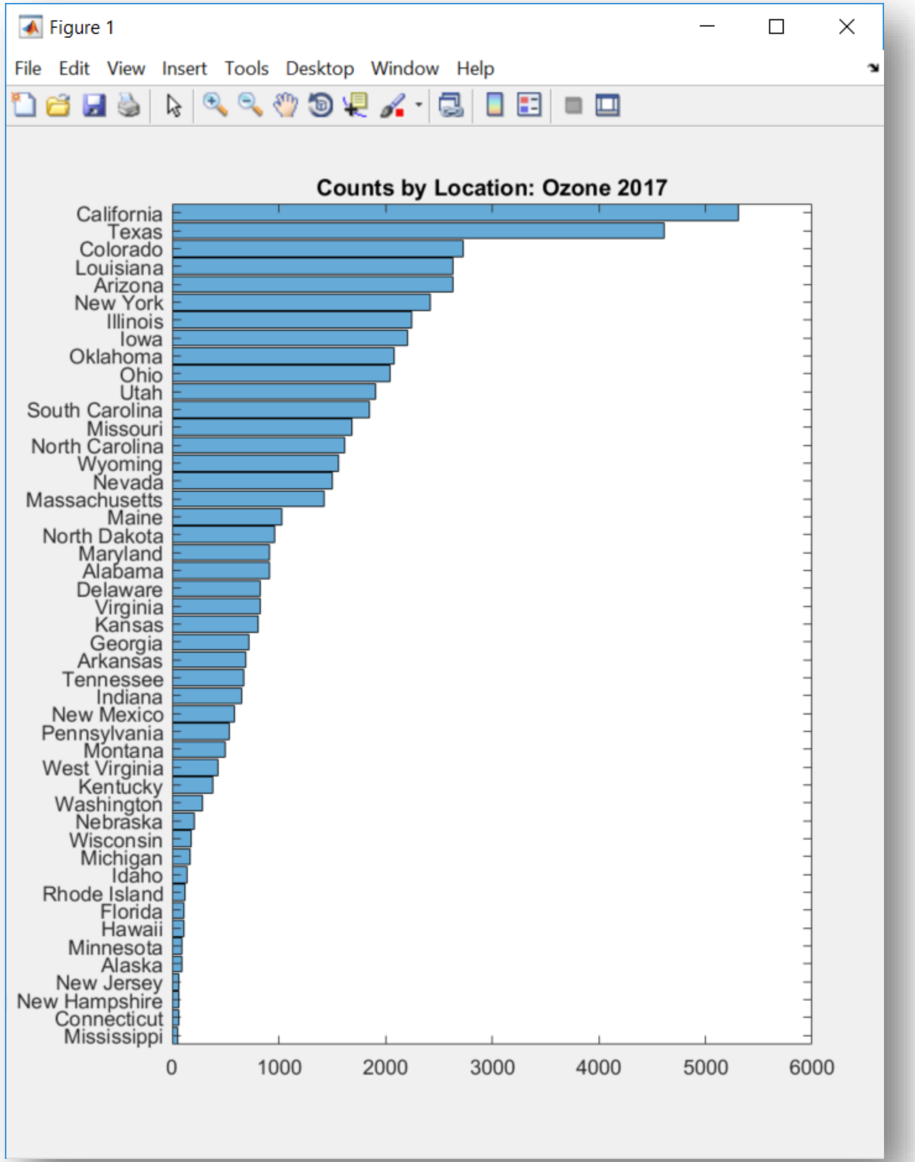


Get a summary of the data

tt – tall table

```
s = summary(ozone)
```

```
Evaluating tall expression using the Spark Cluster:
- Pass 1 of 1: Completed in 49 sec
Evaluation completed in 50 sec
s = struct with fields:

       DateLocal: [1×1 struct]
  ArithmeticMean: [1×1 struct]
             AQI: [1×1 struct]
       StateName: [1×1 struct]
```



Use data types to best represent the data

```
ozone.DateLocal = datetime(ozone.DateLocal,'InputFormat','uuuu-MM-dd');
ozone = table2timetable(ozone);
ozone.AQI = double(ozone.AQI)
```

```
ozone =
  M×3 tall timetable
     DateLocal       ArithmeticMean    AQI    StateName
    ___________      ______________    ___    _________
    04-Apr-1980        0.0475          67     Alabama
    05-Apr-1980      0.036588          67     Alabama
    06-Apr-1980      0.055824          84     Alabama
    07-Apr-1980      0.043941          61     Alabama
    08-Apr-1980      0.044235          49     Alabama
    09-Apr-1980      0.042765          58     Alabama
    10-Apr-1980         0.034          67     Alabama
    11-Apr-1980      0.041647          49     Alabama
         :               :              :        :
         :               :              :        :
```

Managing Big and Messy Time-stamped Data

Functions

`timetable`	Timetable array with time-stamped rows and variables of different types
`retime`	Resample or aggregate data in timetable, and resolve duplicate or irregular times
`synchronize`	Synchronize timetables to common time vector, and resample or aggregate data from input timetables
`lag`	Time-shift data in timetable
`table2timetable`	Convert table to timetable
`array2timetable`	Convert homogeneous array to timetable
`timetable2table`	Convert timetable to table
`istimetable`	Determine if input is timetable
`isregular`	Determine whether times in timetable are regular
`timerange`	Time range for timetable row subscripting
`withtol`	Time tolerance for timetable row subscripting
`vartype`	Subscript into table or timetable by variable type
`rmmissing`	Remove missing entries
`issorted`	Determine if array is sorted
`sortrows`	Sort rows of matrix or table
`unique`	Unique values in array

Use the results of explorations to help make decisions

```
ozone =
  M×3 tall timetable
```

DateLocal	ArithmeticMean	AQI	StateName
04-Apr-1980	0.0475	67	Alabama
05-Apr-1980	0.036588	67	Alabama
06-Apr-1980	0.055824	84	Alabama
07-Apr-1980	0.043941	61	Alabama
08-Apr-1980	0.044235	49	Alabama
09-Apr-1980	0.042765	58	Alabama
10-Apr-1980	0.034	67	Alabama
11-Apr-1980	0.041647	49	Alabama
:	:	:	:
:	:	:	:

```
pressure =
  M×4 tall timetable
```

DateLocal	SampleMeasurement	ParameterName	StateName
01-May-1980 00:00:00	908	Barometric pressure	Montana
01-May-1980 01:00:00	908	Barometric pressure	Montana
01-May-1980 02:00:00	908	Barometric pressure	Montana
01-May-1980 03:00:00	908	Barometric pressure	Montana
01-May-1980 04:00:00	908	Barometric pressure	Montana
01-May-1980 05:00:00	908	Barometric pressure	Montana
01-May-1980 06:00:00	908	Barometric pressure	Montana
01-May-1980 07:00:00	908	Barometric pressure	Montana
:	:	:	:
:	:	:	:

- Synchronize to daily data
- By location

DateLocal	StateName	AQI	O3	CO	SO2	NO2	T	P	WindDir	WindSpd	DP	RH
01-Jan-1980	New York	7	0.004235	1.4833	48.292	30.125	44.596	970.26	157.94	5.7067	28	64.995
02-Jan-1980	New York	14	0.006118	1	42.333	23.083	44.052	960.81	221.61	6.0492	26	81.171
03-Jan-1980	New York	17	0.014706	0.95	31.917	21.917	40.094	971.5	249.59	7.7008	11	79.395
04-Jan-1980	New York	15	0.008353	1.0833	37.75	24.375	40.07	982.47	251.96	5.2913	28	70.364
05-Jan-1980	New York	24	0.017176	0.7375	33.917	25.042	40.054	987.97	248.6	4.2533	25	66.574
06-Jan-1980	New York	21	0.015176	1.0292	48.125	26.375	46.059	990.06	195.86	3.3733	16	55.074
07-Jan-1980	New York	19	0.017353	1.5458	65.542	36.042	49.698	984.93	186.6	3.0873	22	78.042
08-Jan-1980	New York	15	0.009412	0.95652	40.957	25.957	52.472	979.23	141.23	2.2872	17	93.658
:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:

Synchronize all data to daily times

```
dailyMeteorologicalData = synchronize(T,P,WindDir,WindSpd,DP,RH,'daily','mean');
```

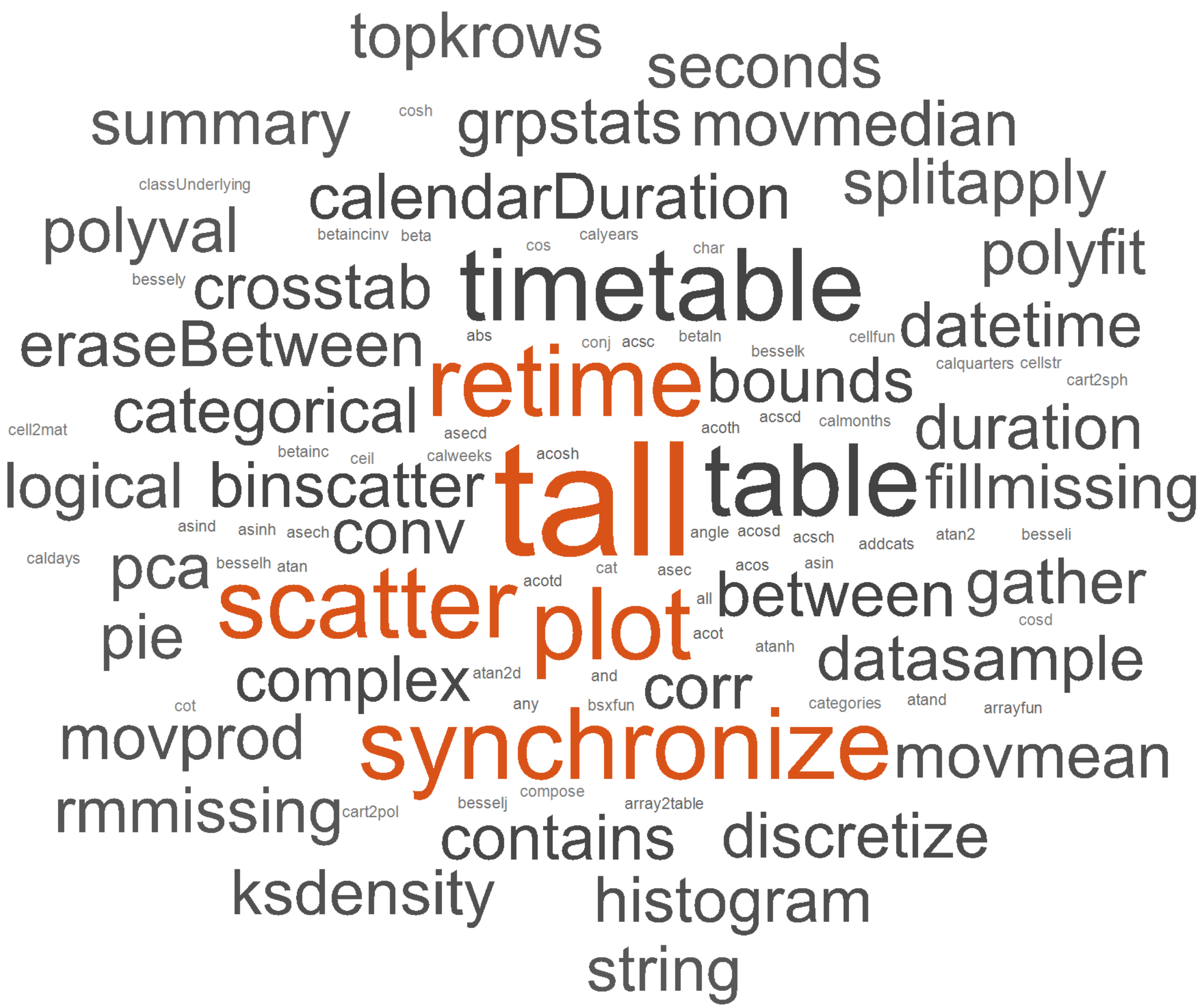
```
dailyData = synchronize(O3,CO,SO2,NO2,dailyMeteorologicalData);
```

Clean messy data using common preprocessing functions

```
ozone = sortrows(ozone);
ozone = rmmissing(ozone,'MinNumMissing',6);
ozone.eightHr = smoothdata(ozone.SampleMeasurement,'movmean',8);
daily8hrmax = retime(ozone(:,'eightHr'),'daily','max')
```

```
daily8hrmax =
  M×1 tall timetable
    DateLocal     eightHr
    _________     _______
        ?            ?
        ?            ?
        ?            ?
        :            :
        :            :
Preview deferred. Learn more.
```

Use familiar MATLAB functions on `tall` arrays



[Functions Supported with Tall Arrays]()

You don’t need to leave MATLAB to monitor large jobs

```
Evaluating tall expression using the Spark Cluster:
- Pass 1 of 13: Completed in 4.0333 min
- Pass 2 of 13: Completed in 2.3 min
- Pass 3 of 13: Completed in 1.8667 min
- Pass 4 of 13: Completed in 4.2167 min
- Pass 5 of 13: Completed in 4.2167 min
- Pass 6 of 13: Completed in 4.3 min
- Pass 7 of 13: Completed in 1.2 min
- Pass 8 of 13: Completed in 3.75 min
- Pass 9 of 13: Completed in 2.5167 min
- Pass 10 of 13: Completed in 38.7 min
- Pass 11 of 13: Completed in 51 sec
- Pass 12 of 13: Completed in 26.833 min
- Pass 13 of 13: 72% complete
```

Spark 2.0.0 | Jobs | Stages | Storage | Environment | Executors | **MATLAB Spark Job** application UI

Spark Jobs (?)

User: hgorr
Total Uptime: 51 min
Scheduling Mode: FIFO
Active Jobs: 1
Completed Jobs: 8
Failed Jobs: 2

▸ Event Timeline

Active Jobs (1)

Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
10 (MATLAB_Pass_10)	runJob at SparkIntegContext.java:662	2017/09/17 15:11:22	31 s	0/1	21/382

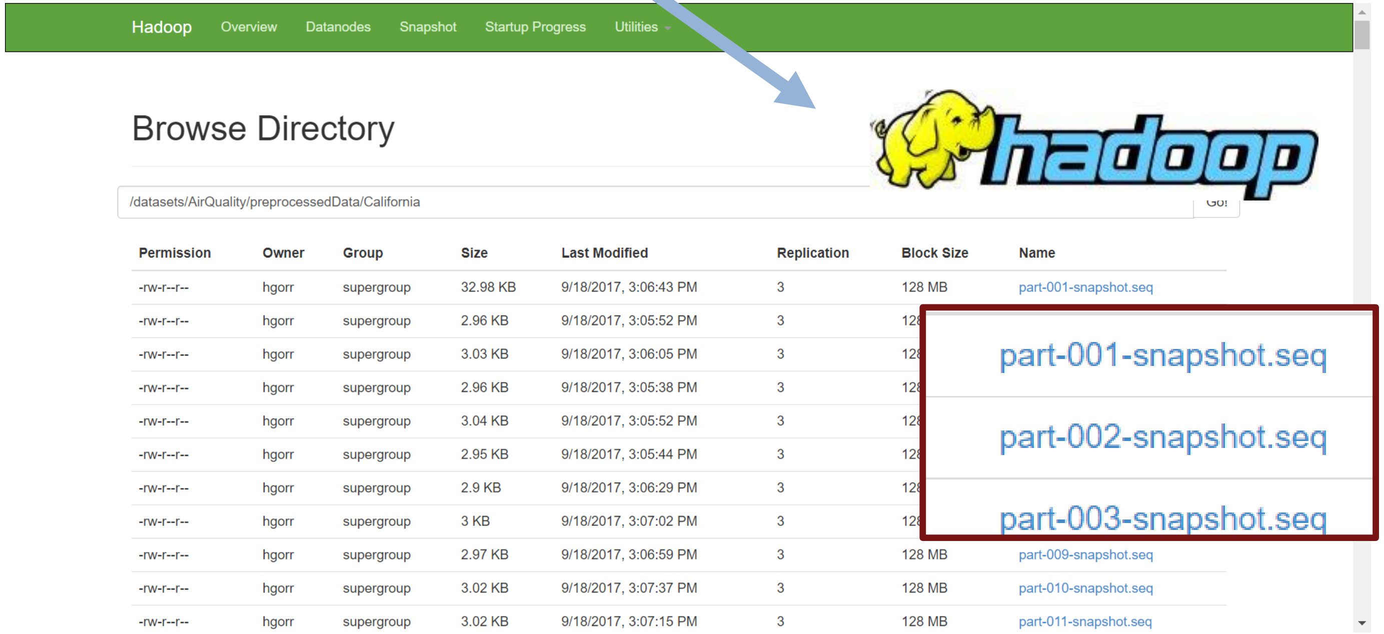
Completed Jobs (8)

Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
9 (MATLAB_Pass_9)	runJob at SparkIntegContext.java:662	2017/09/17 15:09:30	1.9 min	1/1	131/131
8 (MATLAB_Pass_8)	runJob at SparkIntegContext.java:662	2017/09/17 15:05:17	4.2 min	1/1	276/276
7 (MATLAB_Pass_7)	runJob at SparkIntegContext.java:662	2017/09/17 14:59:11	6.1 min	1/1	382/382
6 (MATLAB_Pass_6)	runJob at SparkIntegContext.java:662	2017/09/17 14:57:55	1.3 min	1/1	89/89
5 (MATLAB_Pass_5)	runJob at SparkIntegContext.java:662	2017/09/17 14:52:18	1.9 min	1/1	131/131

Save preprocessed data

```
newfiledir = 'hdfs://hadoop01glnxa64:54310/datasets/AirQuality/preprocessedData/';
write(newfiledir,dailyData)
```

```
Writing tall data to folder hdfs://hadoop01glnxa64:54310/datasets/AirQuality/preprocessedData/
Evaluating tall expression using the Spark Cluster:
- Pass 1 of 13: Completed in 4.0333 min
- Pass 2 of 13: Completed in 2.3 min
- Pass 3 of 13: Completed in 1.8667 min
- Pass 4 of 13: Completed in 4.2167 min
- Pass 5 of 13: Completed in 4.2167 min
- Pass 6 of 13: Completed in 4.3 min
- Pass 7 of 13: Completed in 1.2 min
- Pass 8 of 13: Completed in 3.75 min
- Pass 9 of 13: Completed in 2.5167 min
- Pass 10 of 13: Completed in 38.7 min
- Pass 11 of 13: Completed in 51 sec
- Pass 12 of 13: Completed in 26.833 min
- Pass 13 of 13: 72% complete
Evaluation 98% complete
```



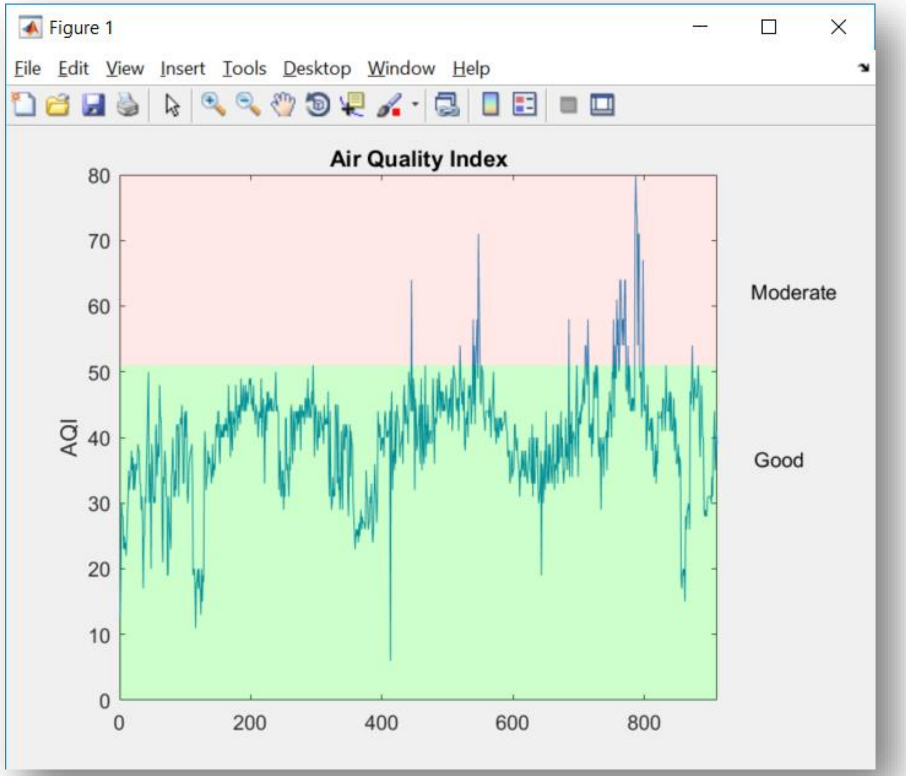
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hgorr	supergroup	32.98 KB	9/18/2017, 3:06:43 PM	3	128 MB	part-001-snapshot.seq
-rw-r--r--	hgorr	supergroup	2.96 KB	9/18/2017, 3:05:52 PM	3	128	
-rw-r--r--	hgorr	supergroup	3.03 KB	9/18/2017, 3:06:05 PM	3	128	
-rw-r--r--	hgorr	supergroup	2.96 KB	9/18/2017, 3:05:38 PM	3	128	
-rw-r--r--	hgorr	supergroup	3.04 KB	9/18/2017, 3:05:52 PM	3	128	
-rw-r--r--	hgorr	supergroup	2.95 KB	9/18/2017, 3:05:44 PM	3	128	
-rw-r--r--	hgorr	supergroup	2.9 KB	9/18/2017, 3:06:29 PM	3	128	
-rw-r--r--	hgorr	supergroup	3 KB	9/18/2017, 3:07:02 PM	3	128	
-rw-r--r--	hgorr	supergroup	2.97 KB	9/18/2017, 3:06:59 PM	3	128 MB	part-009-snapshot.seq
-rw-r--r--	hgorr	supergroup	3.02 KB	9/18/2017, 3:07:37 PM	3	128 MB	part-010-snapshot.seq
-rw-r--r--	hgorr	supergroup	3.02 KB	9/18/2017, 3:07:15 PM	3	128 MB	part-011-snapshot.seq

part-001-snapshot.seq
part-002-snapshot.seq
part-003-snapshot.seq

- ✓ Introduction to Tall Arrays
- ✓ Tall Arrays for Big Data Visualization and Preprocessing
- ▪ Machine Learning for Big Data Using Tall Arrays

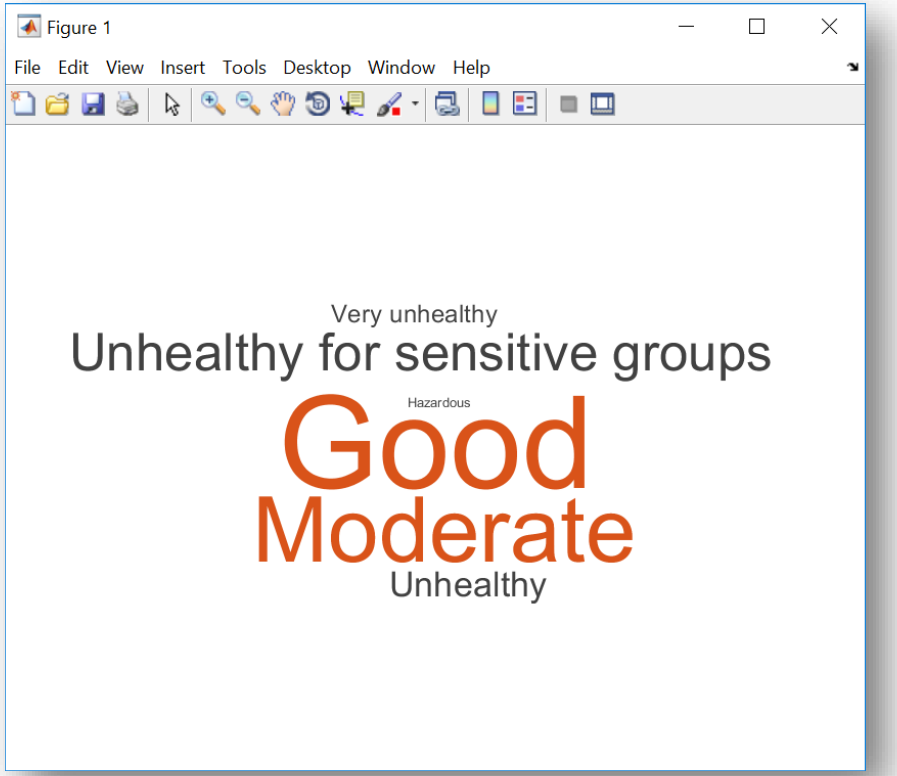
Predict air quality

Air Quality Index



Regression

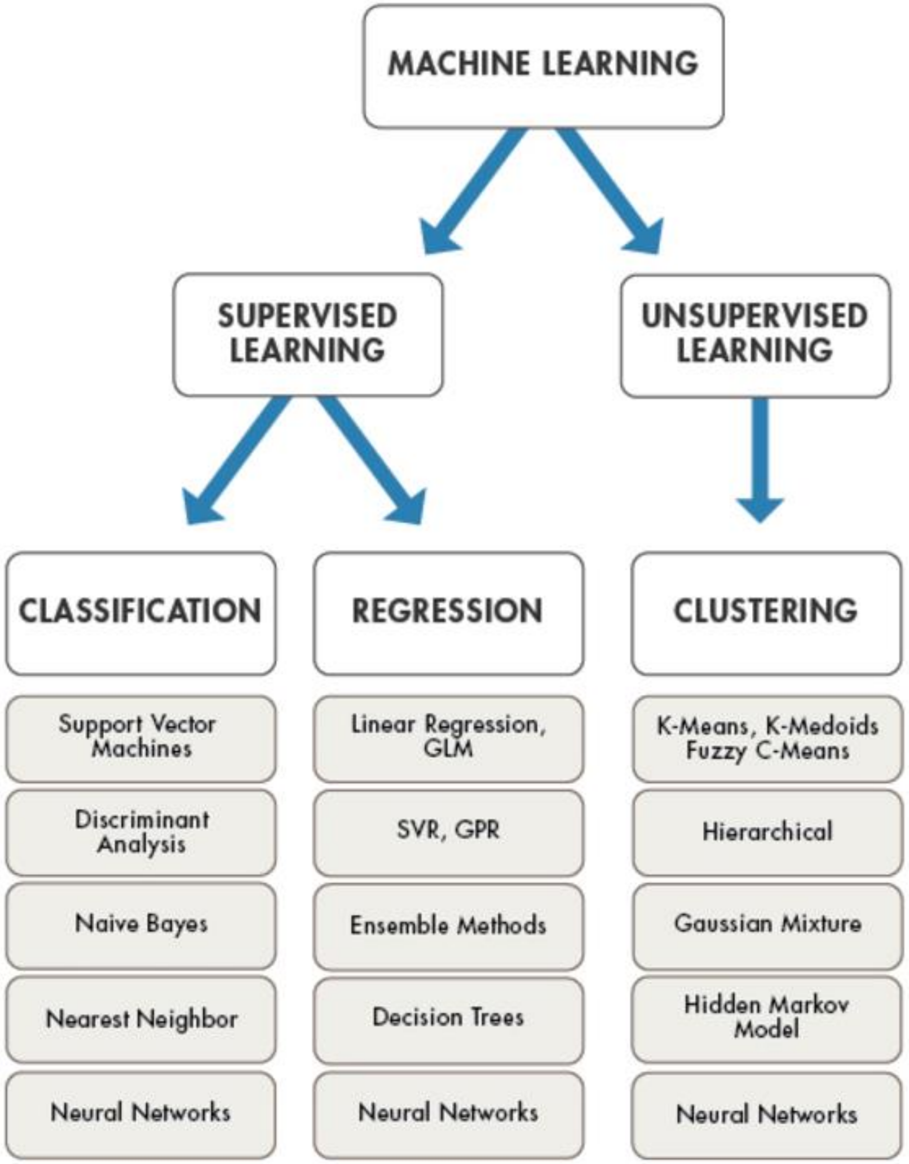
Air Quality Label



Classification

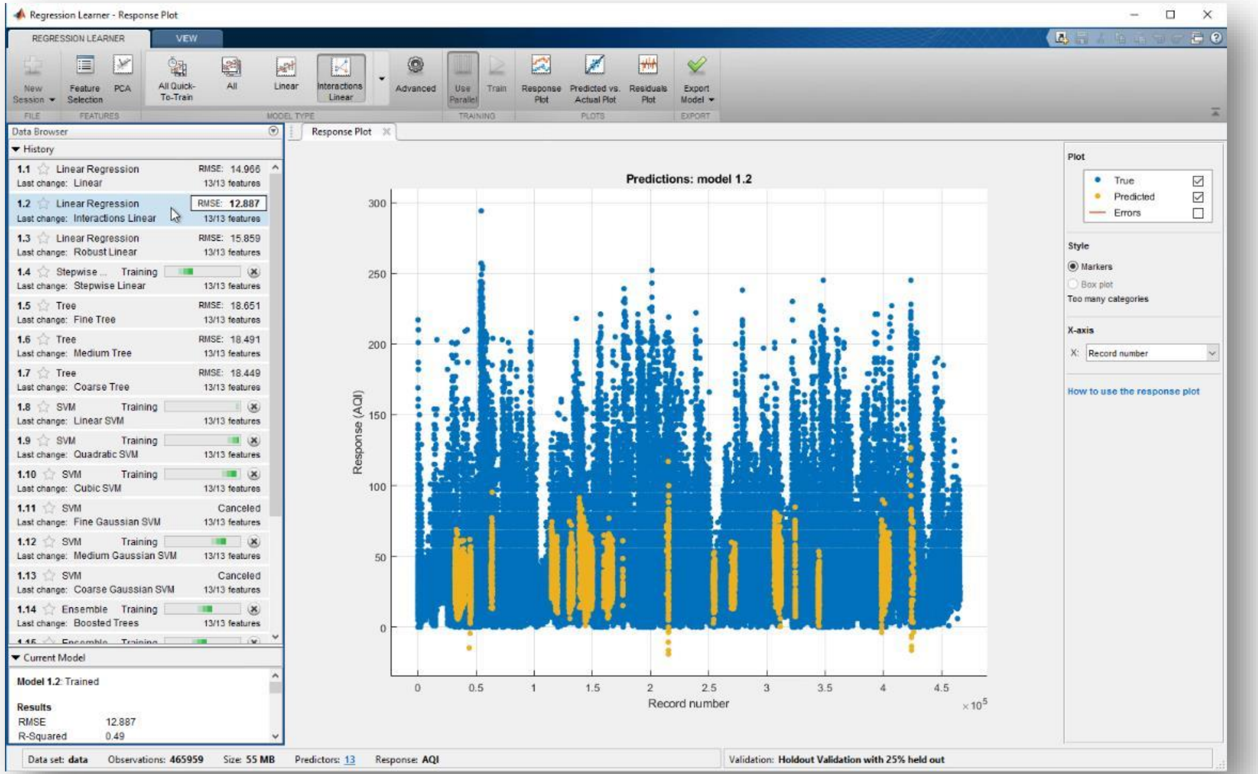
How do you know which model to use?

- Try them all ☺



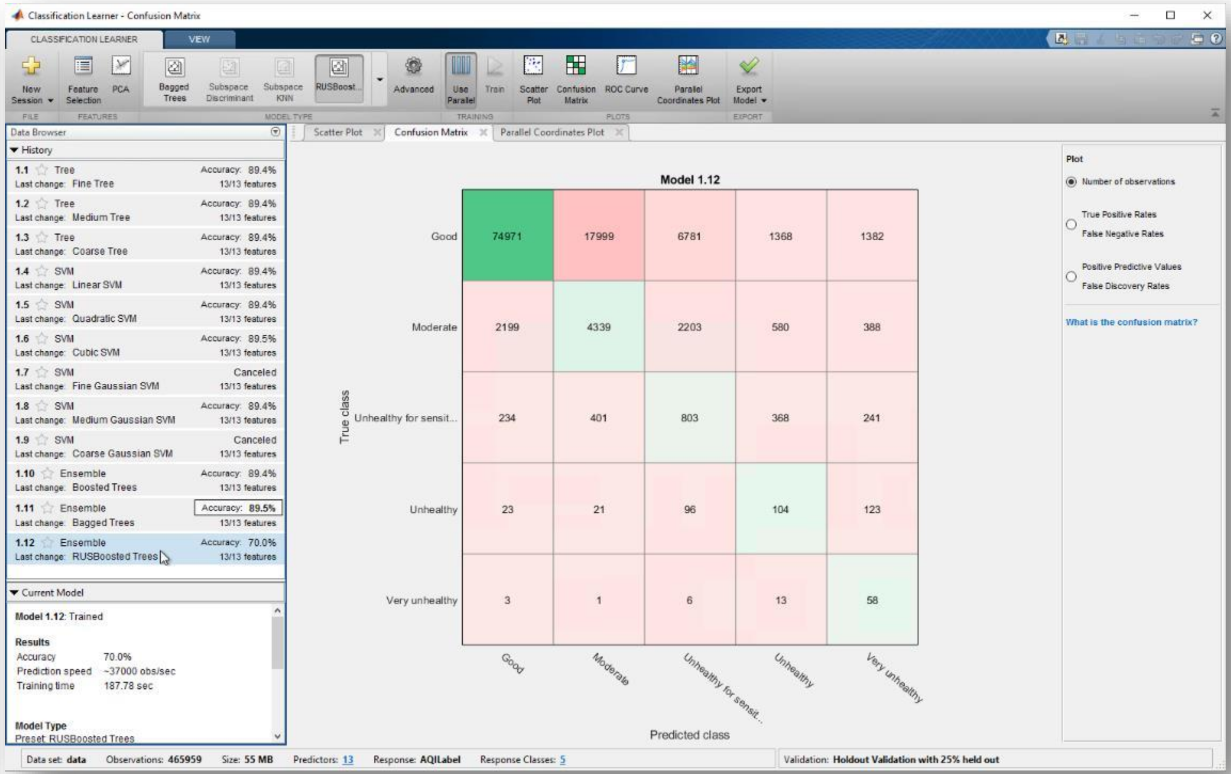
Use apps for model exploration on a subset of data

Air Quality Index



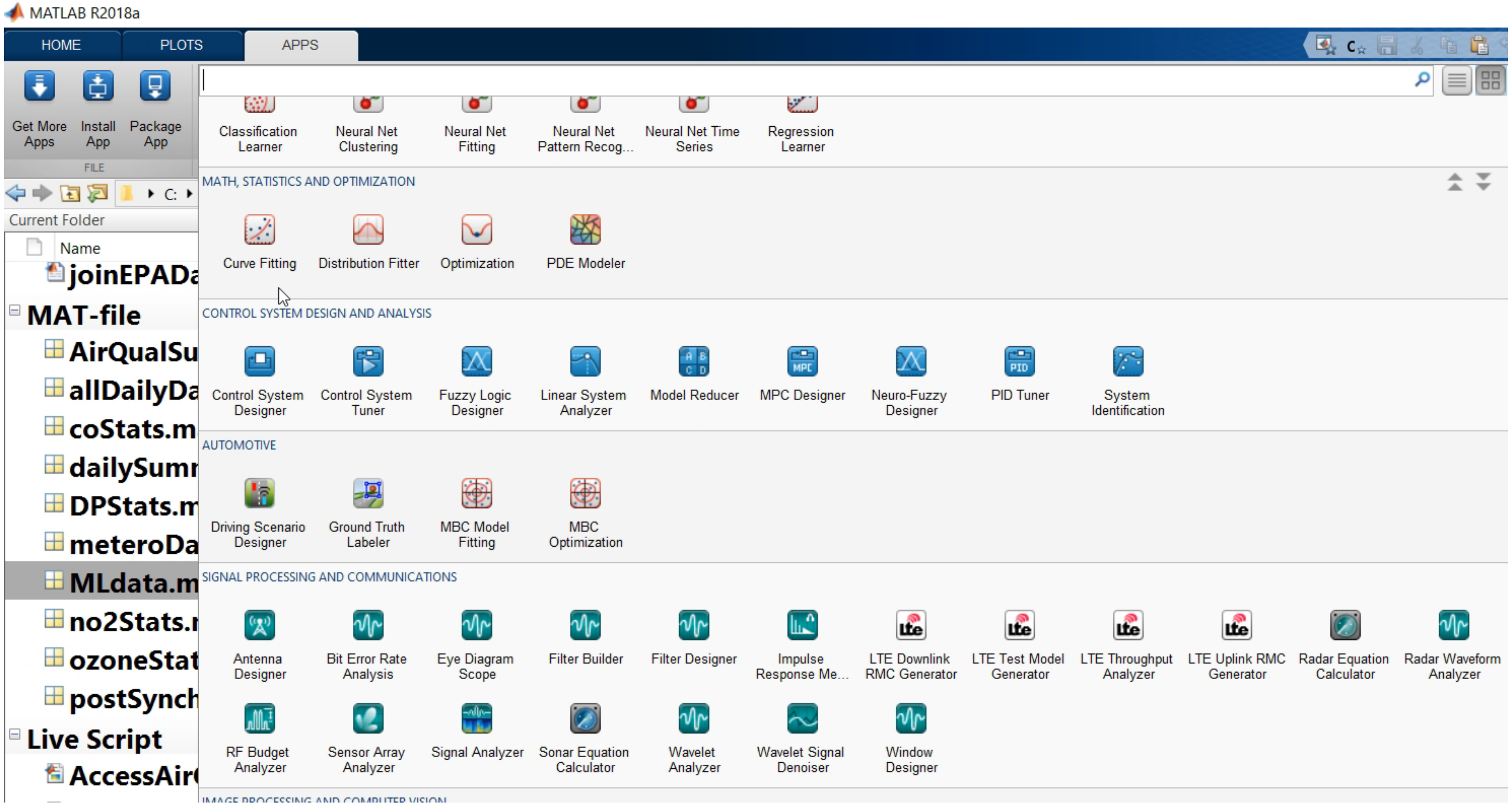
Regression Learner

Air Quality Label

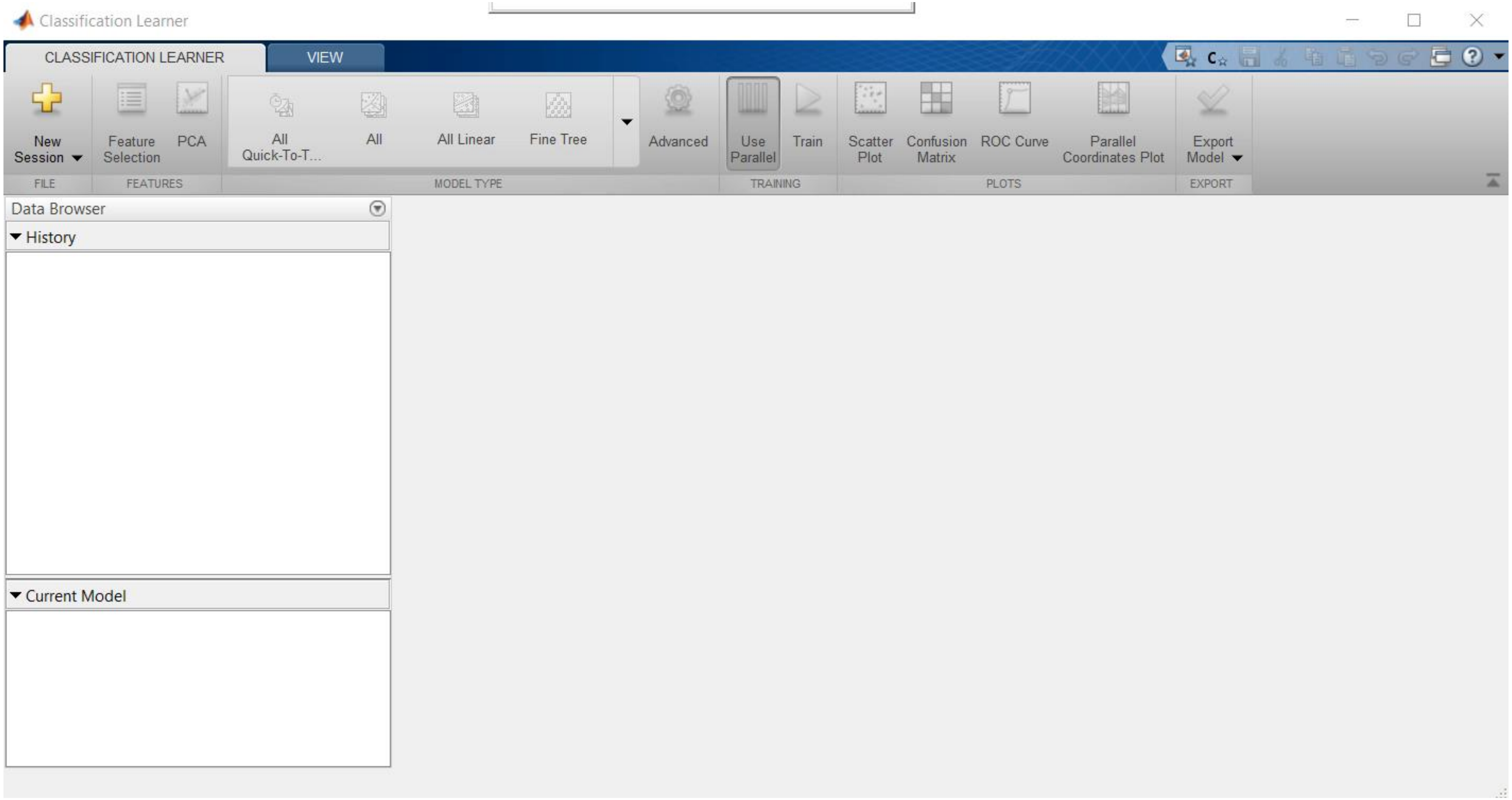


Classification Learner

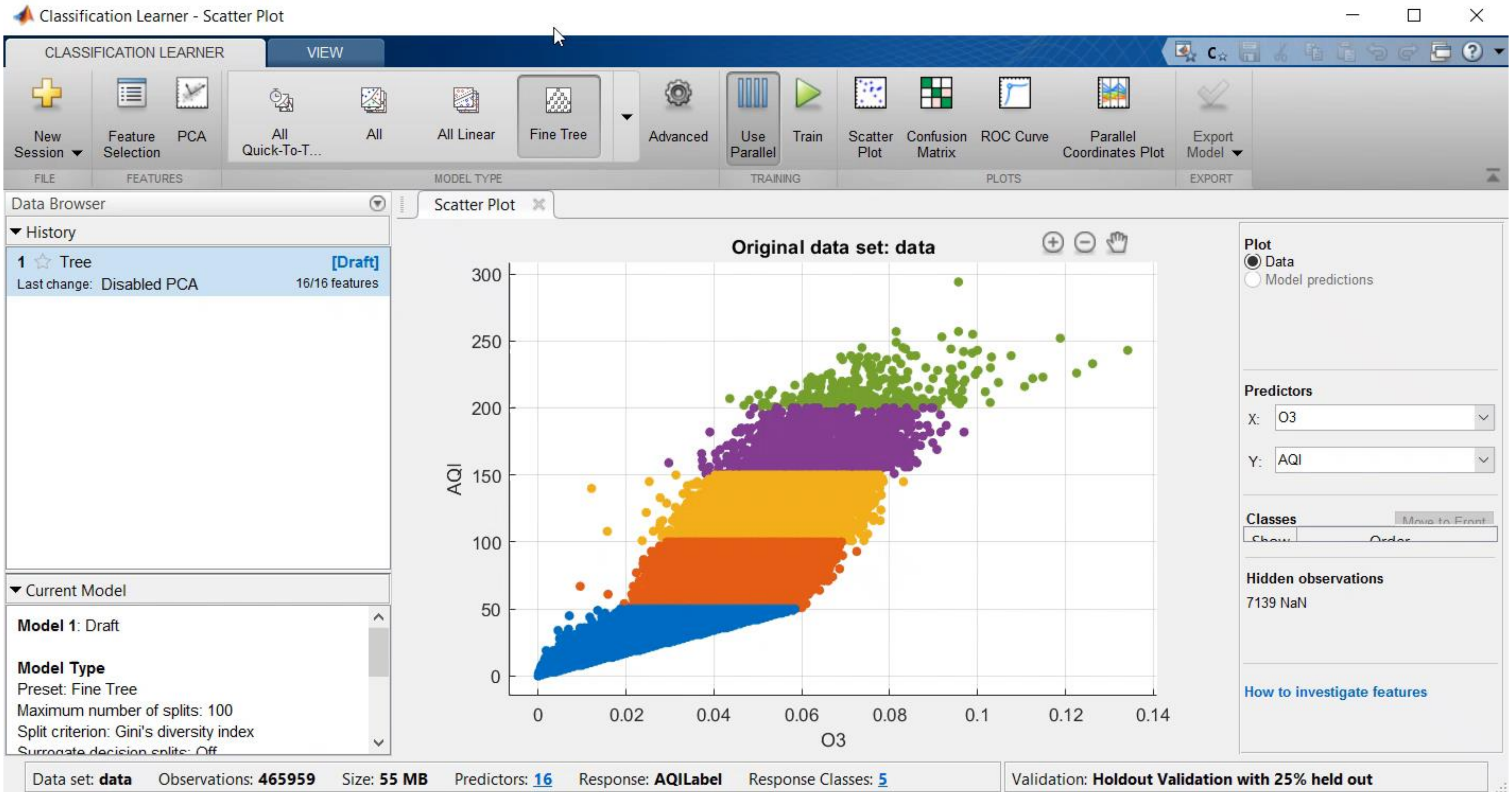
Validate and Compare Machine Learning Models



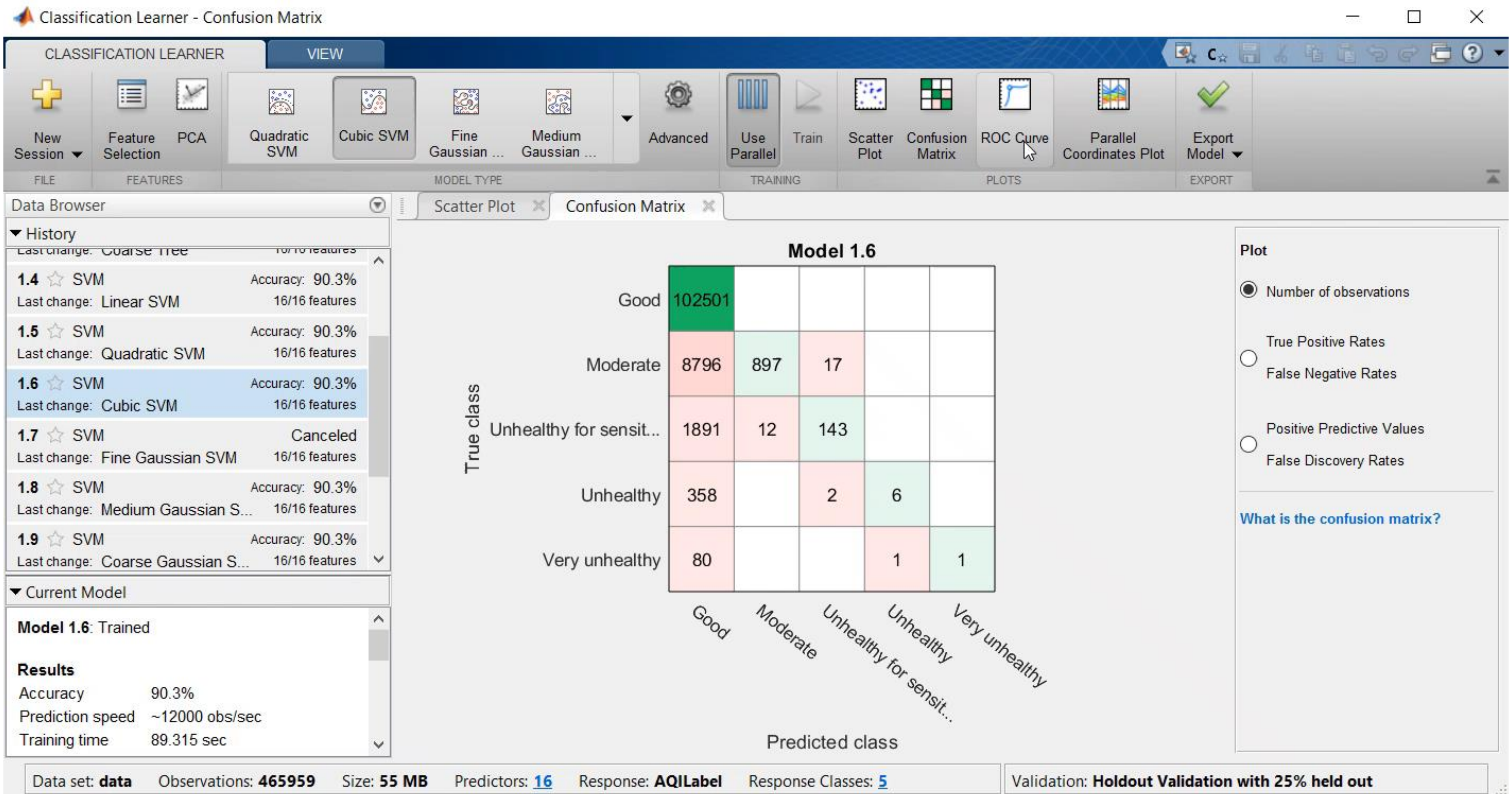
Validate and Compare Machine Learning Models



Validate and Compare Machine Learning Models



Validate and Compare Machine Learning Models



Scale up with `tall` machine learning models

- Linear Regression (`fitlm`)
- Logistic & Generalized Linear Regression (`fitglm`)
- Discriminant Analysis Classification (`fitcdiscr`)
- K-means Clustering (`kmeans`)
- Principal Component Analysis (`pca`)
- Partition for Cross Validation (`cvpartition`)

R2016b

- Linear Support Vector Machine (SVM) Classification (`fitclinear`)
- Naïve Bayes Classification (`fitcnb`)
- Random Forest Ensemble Classification (`TreeBagger`)
- Lasso Linear Regression (`lasso`)

R2017a

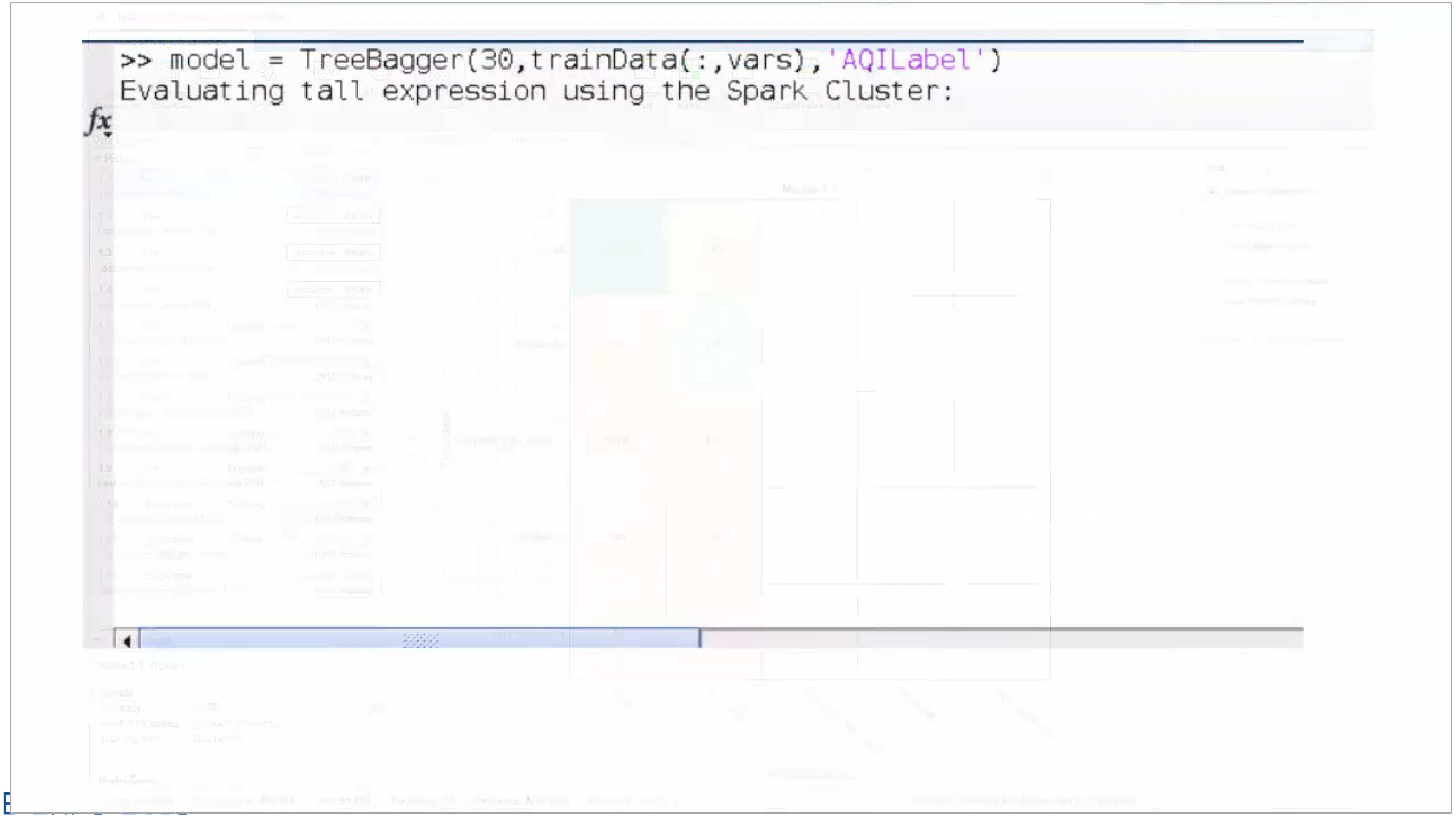
- Linear Support Vector Machine (SVM) Regression (`fitrlinear`)
- Single Classification Decision Tree (`fitctree`)
- Linear SVM Classification with Random Kernel Expansion (`fitckernel`)

R2017b

- Gaussian Kernel Regression (`fitrkernel`)

R2018a

Training Machine Learning Model against Spark for Air Quality Classification

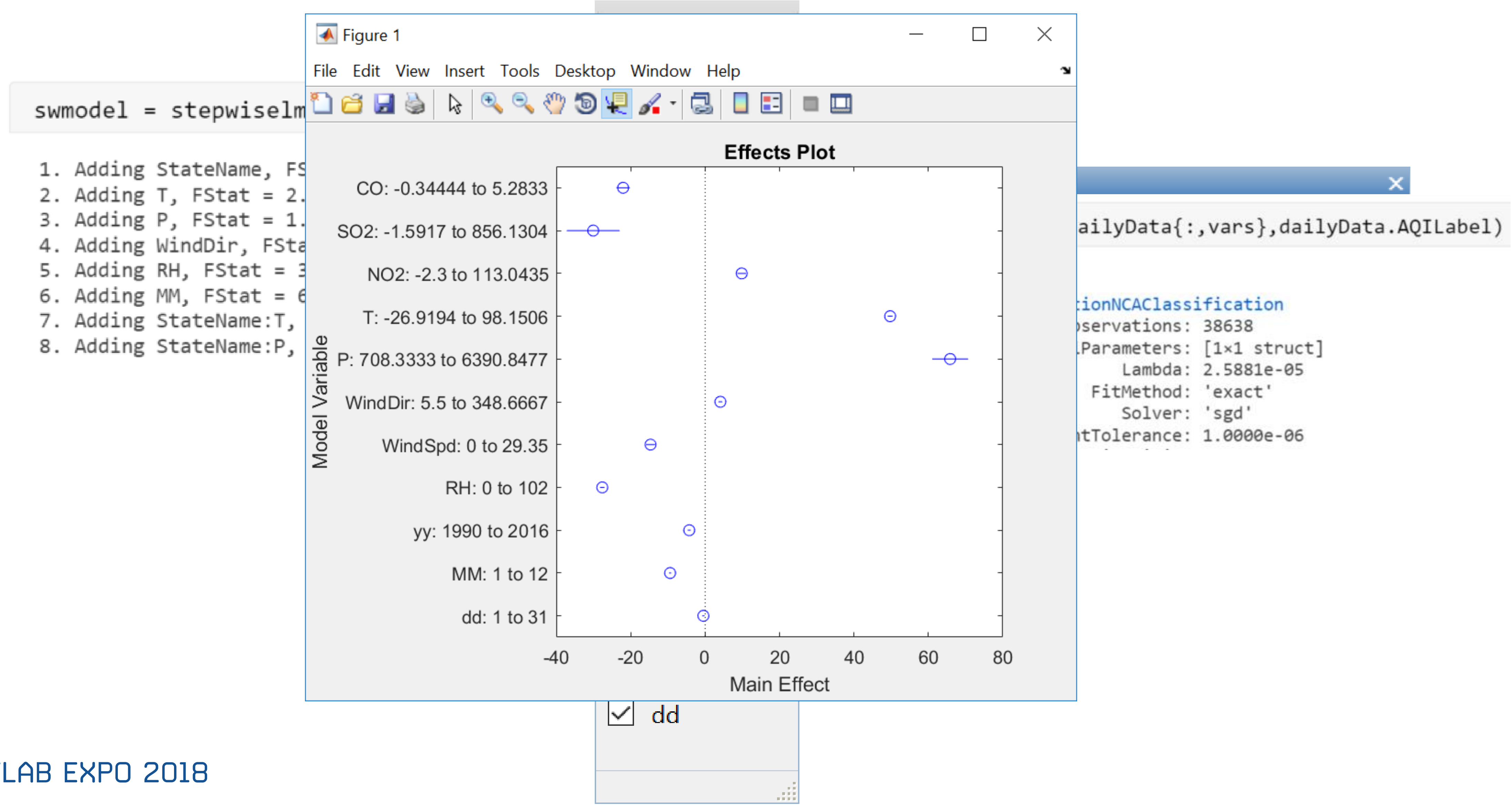


Train and validate with `tall` data for Air Quality Index Prediction

```
model = fitlm(dailyData(:,[5:11,13:16,3]))
```

```
Evaluating tall expression using the Parallel Pool 'local':
Evaluation completed in 0 sec
model =
Compact linear regression model:
    AQI ~ 1 + CO + SO2 + NO2 + T + P + WindDir + WindSpd + RH + YY + MM + DD

Estimated Coefficients:
                    Estimate        SE          tStat       pValue
                   _________    __________    ________    ___________

    (Intercept)       389.51        13.969      27.884    9.9792e-171
    CO                -3.049       0.12769     -23.878    8.2102e-126
    SO2            -0.023073     0.0042052     -5.4868     4.0985e-08
    NO2             0.057154     0.0044742      12.774     2.3766e-37
    T                0.36578     0.0022326      163.84              0
    P              0.0017117     0.0002197      7.7913     6.6682e-15
    WindDir         0.019722    0.00068229      28.906    2.6997e-183
    WindSpd         -0.34815      0.016799     -20.725     2.6815e-95
    RH              -0.24597      0.002423     -101.52              0
    YY              -0.17682     0.0069258      -25.53    1.6608e-143
    MM              -0.77294      0.011332     -68.209              0
    DD             -0.013008     0.0042385     -3.0691      0.0021477


Number of observations: 175927, Error degrees of freedom: 175915
Root Mean Squared Error: 15.6
R-squared: 0.219,  Adjusted R-Squared 0.219
F-statistic vs. constant model: 4.48e+03, p-value = 0
```

Select the most important features



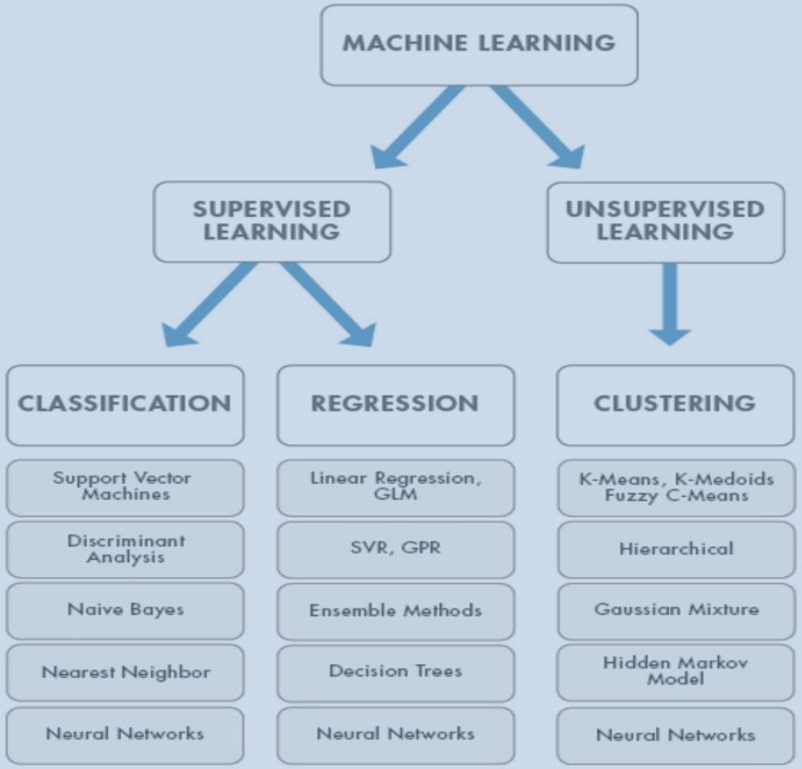
- ✓ **Introduction to Tall Arrays**
- ✓ **Tall Arrays for Big Data Visualization and Preprocessing**
- ✓ **Machine Learning for Big Data Using Tall Arrays**

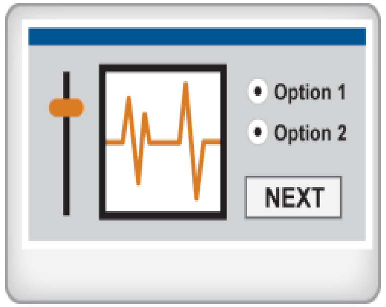
Building machine learning models with big data

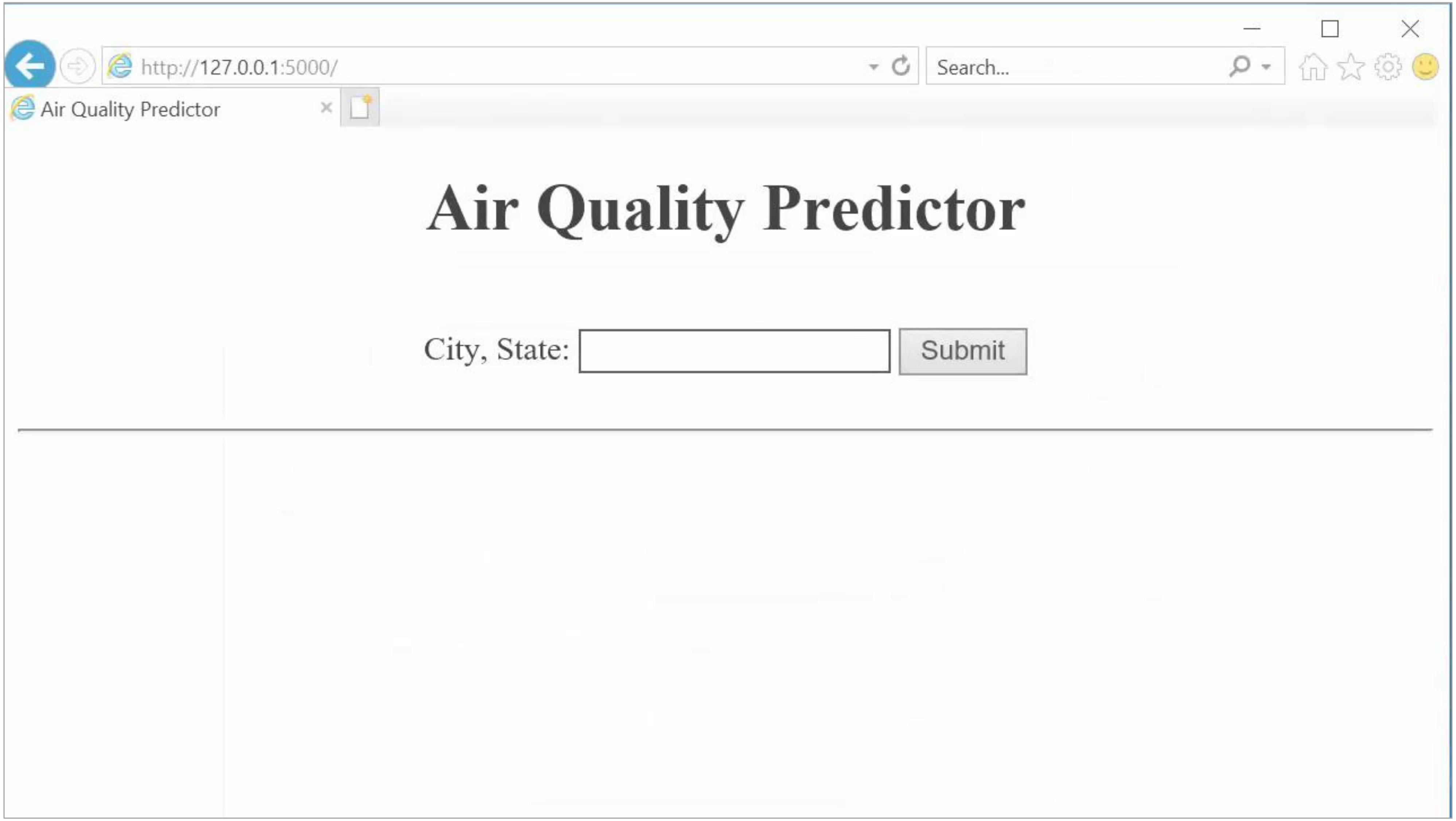
Access

Preprocess, Exploration & Model Development

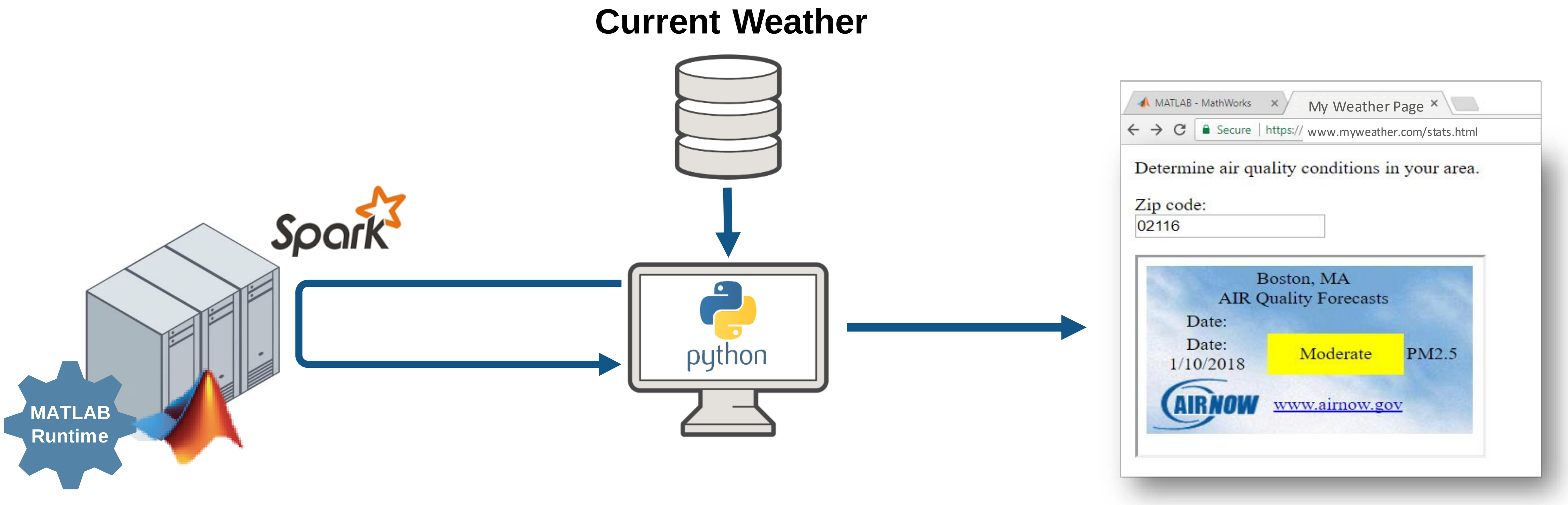
Scale up & Integrate with Production Systems





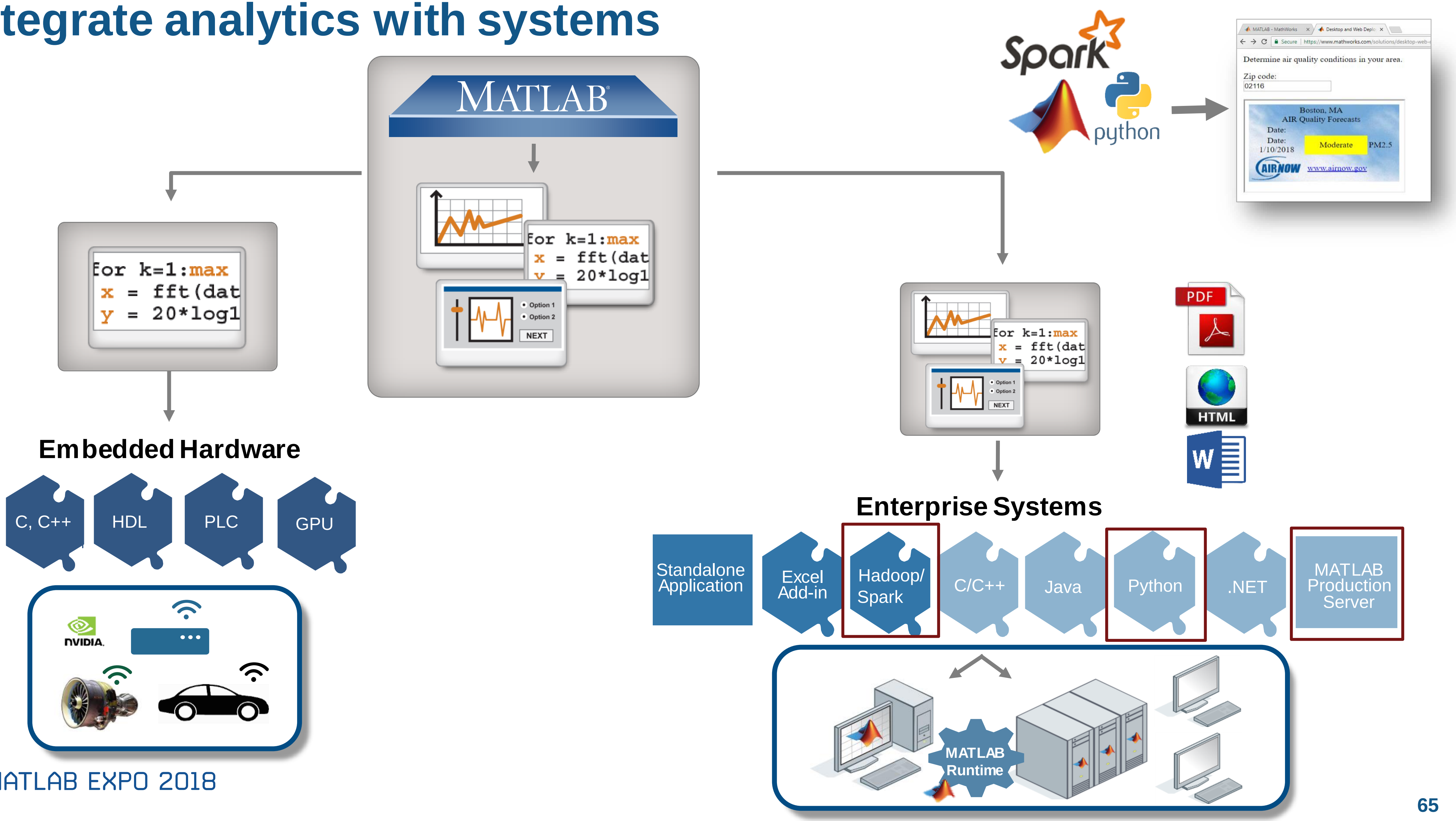

http://127.0.0.1:5000/
Search...
Air Quality Predictor
Air Quality Predictor
City, State:
Submit

Predict air quality for given location



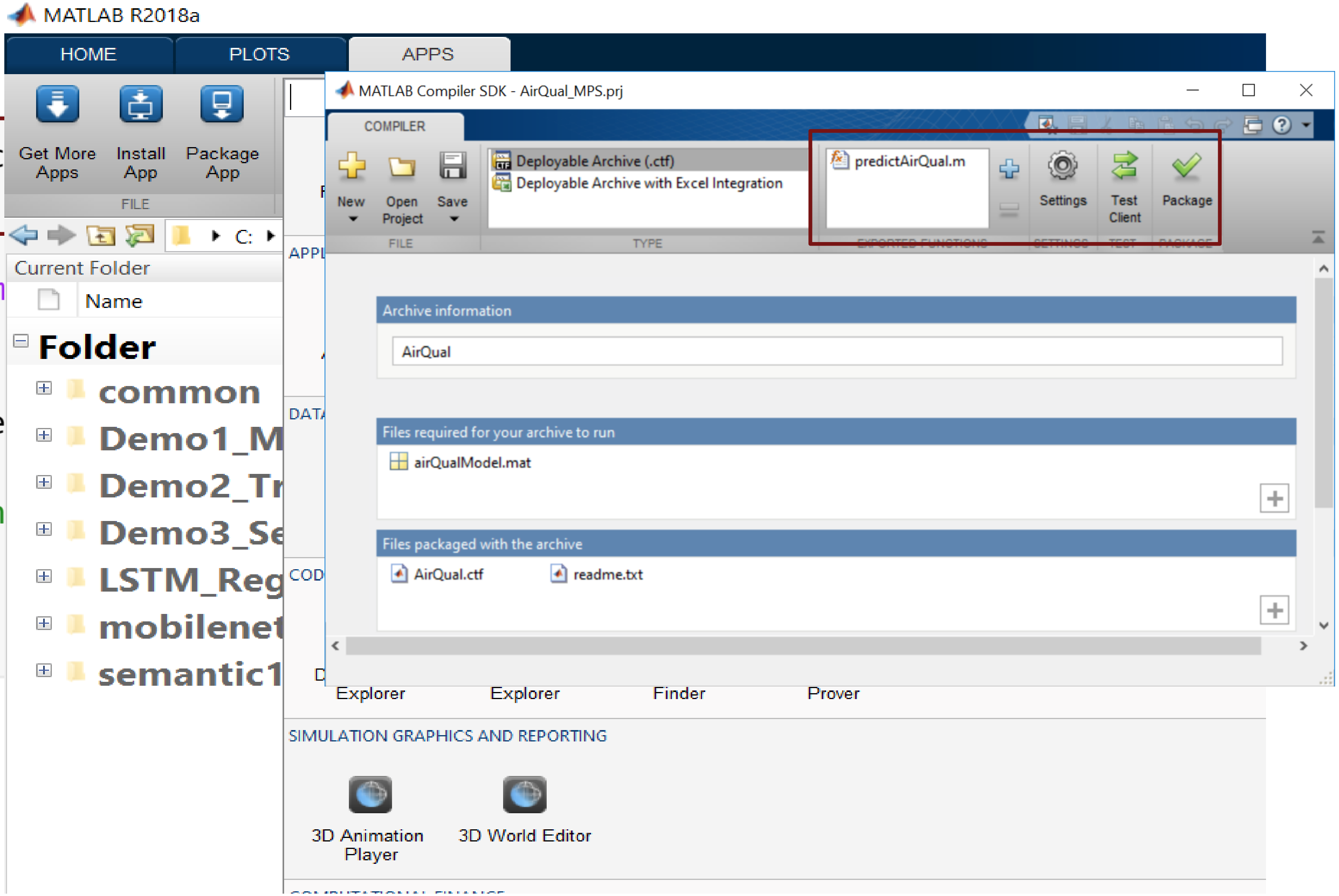
Use MATLAB model running on Spark in Python web framework

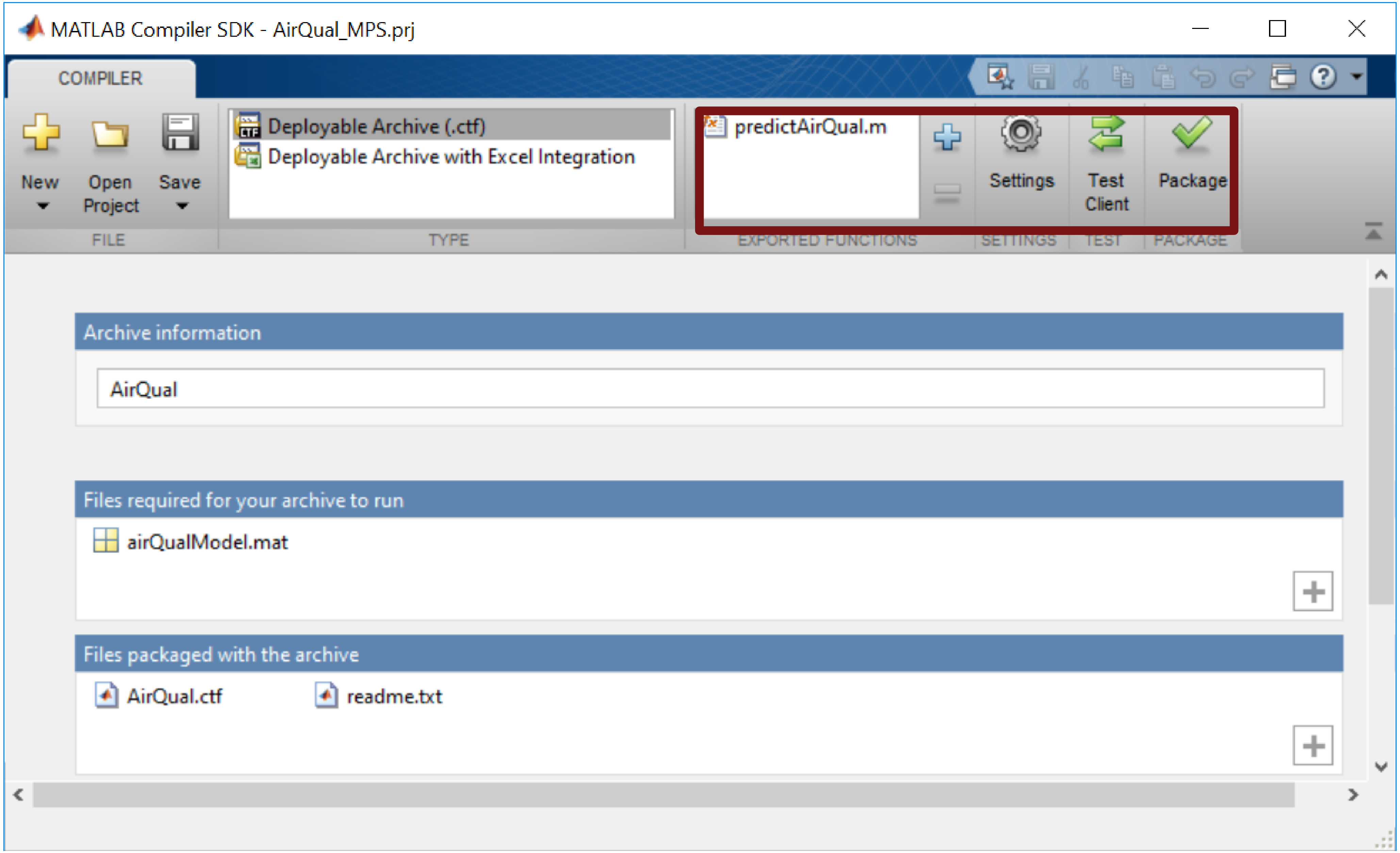
Integrate analytics with systems



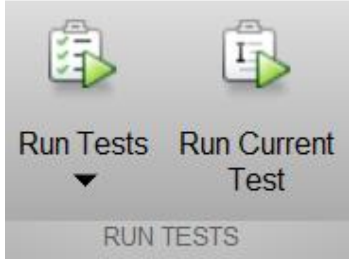
Package and test MATLAB code

```
function airQual = predic

% Load model
aq = load('airQualModel.m

% Determine air quality
airQual = predict(aq.mode

% Convert for use in pyth
airQual = char(airQual);

end
```

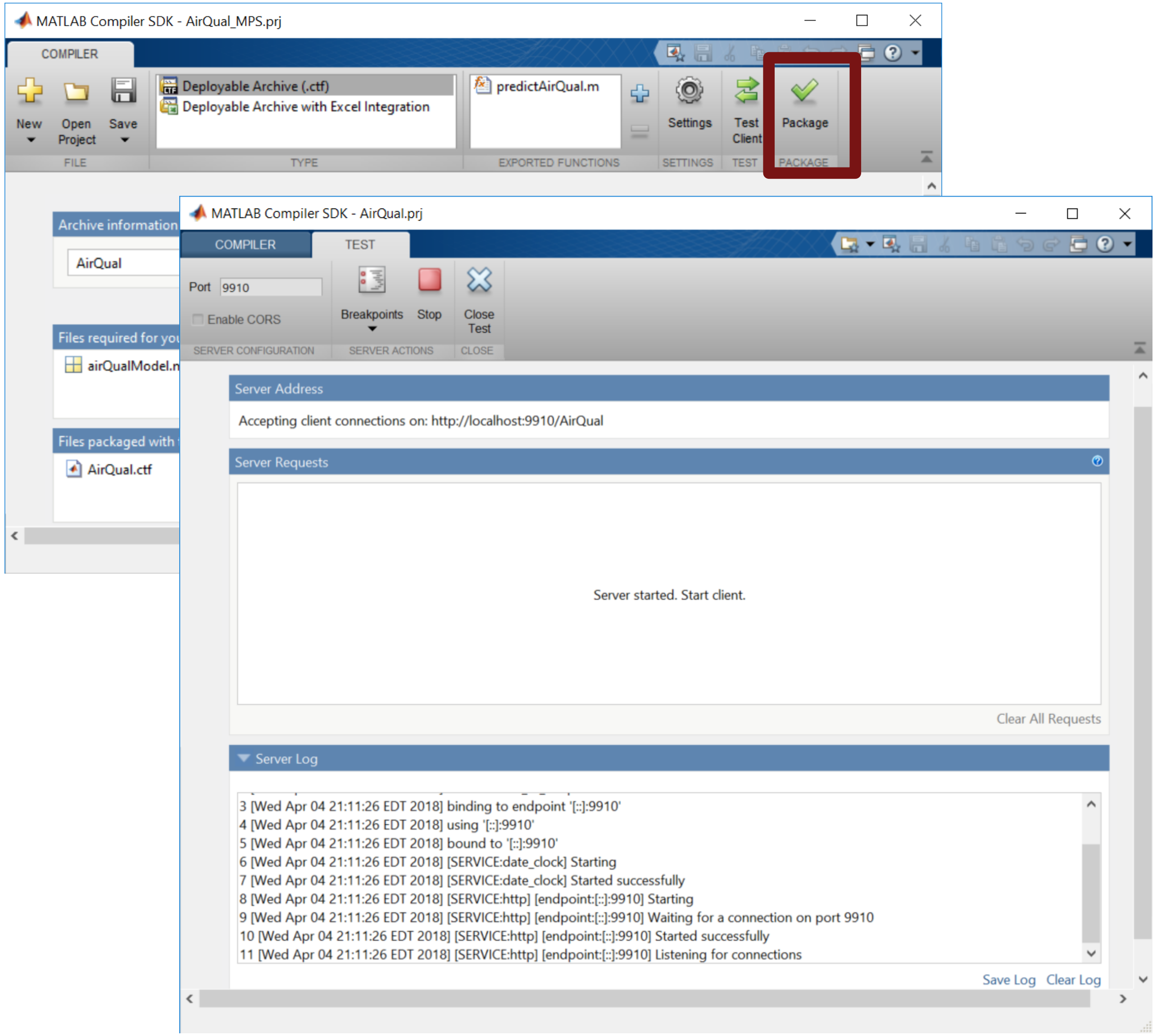



MATLAB Compiler SDK - AirQual_MPS.prj
COMPILER
New
Open Project
Save
FILE
Deployable Archive (.ctf)
Deployable Archive with Excel Integration
TYPE
predictAirQual.m
EXPORTED FUNCTIONS
Settings
SETTINGS
Test Client
TEST
Package
PACKAGE
Archive information
AirQual
Files required for your archive to run
airQualModel.mat
Files packaged with the archive
AirQual.ctf
readme.txt

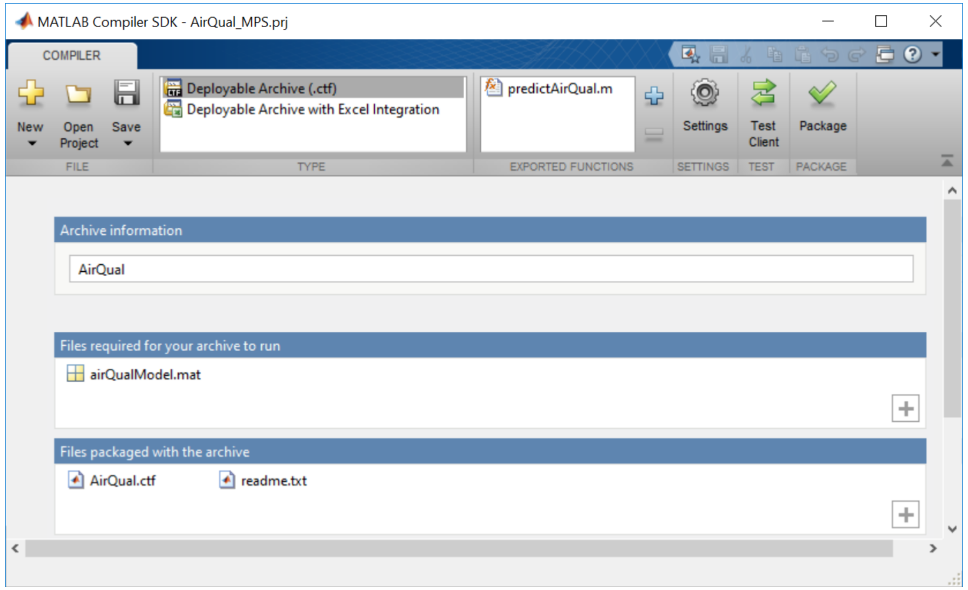
Package and test MATLAB code




```
runtests('testAirQual')

Running testAirQual
....
Done testAirQual
__________

ans =
  1×4 TestResult array with properties:

    Name
    Passed
    Failed
    Incomplete
    Duration
    Details
Totals:
   4 Passed, 0 Failed, 0 Incomplete.
   0.0043759 seconds testing time.
```



Call MATLAB in production environment

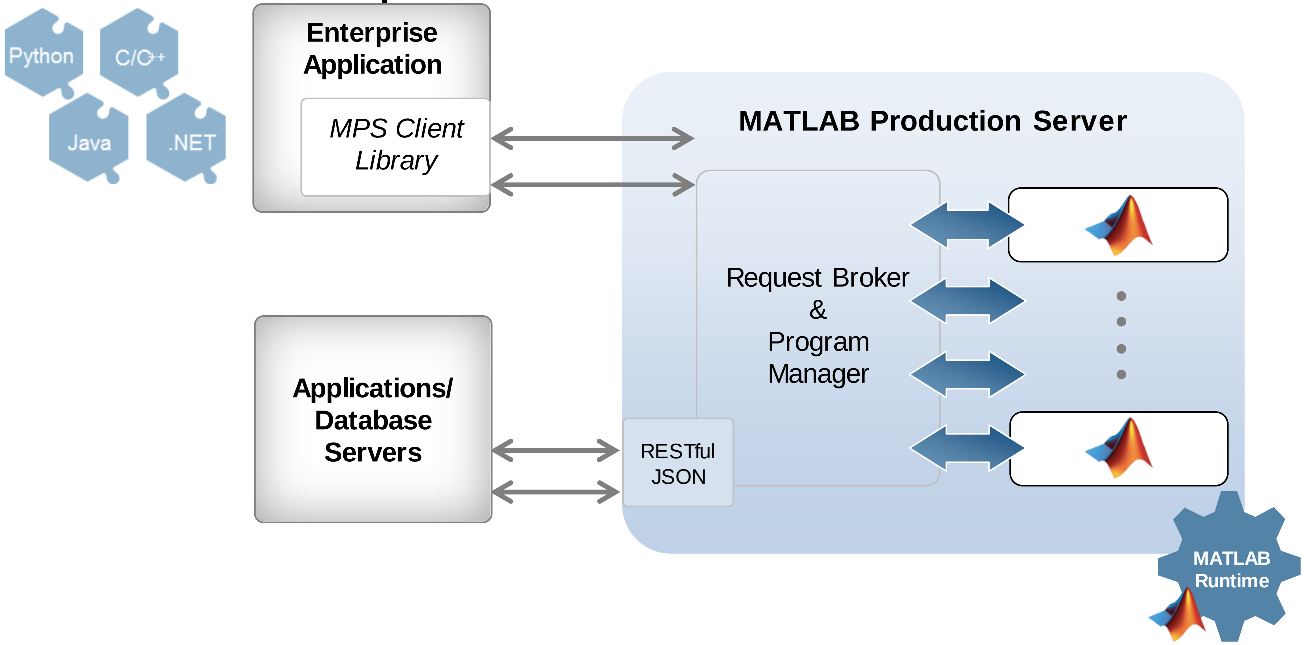


AirQual.ctf

```python
import matlab
from matlab.production_server import client
client_object = client.MWHttpClient('http://<HOST>:<PORT>')
air_qual = client_object.AirQual.predictAirQual(json_data)
```

MATLAB Production Server

- Server software
  - Manages packaged MATLAB programs and worker pool
- MATLAB Runtime libraries
  - Single server can use runtimes from different releases
- RESTful JSON interface
- Lightweight client libraries
  - C/C++, .NET, Python, and Java



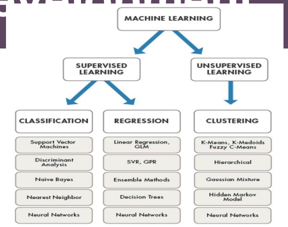
MATLAB for Modeling and Deploying Big Data Applications

Access

- Distributed Data Storage
- Different Data Sources & Types

Easily Access Data however/wherever it is stored using **Datastore**

Preprocess, Exploration & Model Development



- Preprocessing and Visualizing Big Data
- Parallelizing Jobs and Scaling up Computations to Cluster

Prototype and easily scale up algorithms to Big Data platforms using the familiar MATLAB Syntax with **Tall Arrays**

Scale up & Integrate with Production Systems

Spark

MATLAB Excel .NET .exe C/C++ Java .dll

- Enterprise level deployment

Seamless integration with Enterprise level systems using **MATLAB Production Server**

How do you get started?

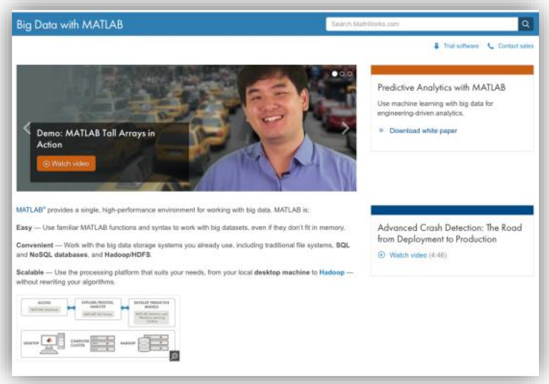
- Try Tall Array Based Processing on Your Own Set of Big Data

- Refer to the example mentioned below to get started:

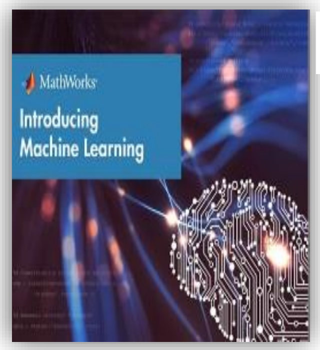
https://in.mathworks.com/help/matlab/examples/analyze-big-data-in-matlab-using-tall-arrays.html

Other Resources

mathworks.com/big-data



mathworks.com/machine-learning



eBook

MathWorks Training Offerings

Machine Learning with MATLAB

INTERMEDIATE

This two-day course focuses on data analytics and machine learning techniques in MATLAB using functionality within Statistics and Machine Learning Toolbox™ and Neural Network Toolbox™. The course demonstrates the use of unsupervised learning to discover features in large data sets and supervised learning to build predictive models. Examples and exercises highlight techniques for visualization and evaluation of results. Topics include:

- Importing and organizing data
- Finding natural patterns in data
- Building predictive models
- Evaluating and improving the model

Prerequisites: *MATLAB Fundamentals*

Parallel Computing with MATLAB

INTERMEDIATE

This two-day course shows how to use Parallel Computing Toolbox™ to speed up existing code and scale up across multiple computers using MATLAB Distributed Computing Server™ (MDCS). Attendees who are working with long-running simulations, or large data sets, will benefit from the hands-on demonstrations and exercises in the course. Topics include:

- Parallel for-loops
- Offloading execution
- Working with clusters
- Distributing and processing large data sets
- GPU computing

Prerequisites: *MATLAB Fundamentals*

http://www.mathworks.com/services/training/

Speaker Details

Email: Alka.Nair@mathworks.in

LinkedIn: https://www.linkedin.com/in/alka-nair-1820501a/

Contact MathWorks India

Products/Training Enquiry Booth

Call: 080-6632-6000

Email: info@mathworks.in

- **Share your experience with MATLAB & Simulink on Social Media**
  - Use #MATLABEXPO

- **Share your session feedback:**

Please fill in your feedback for this session in the feedback form