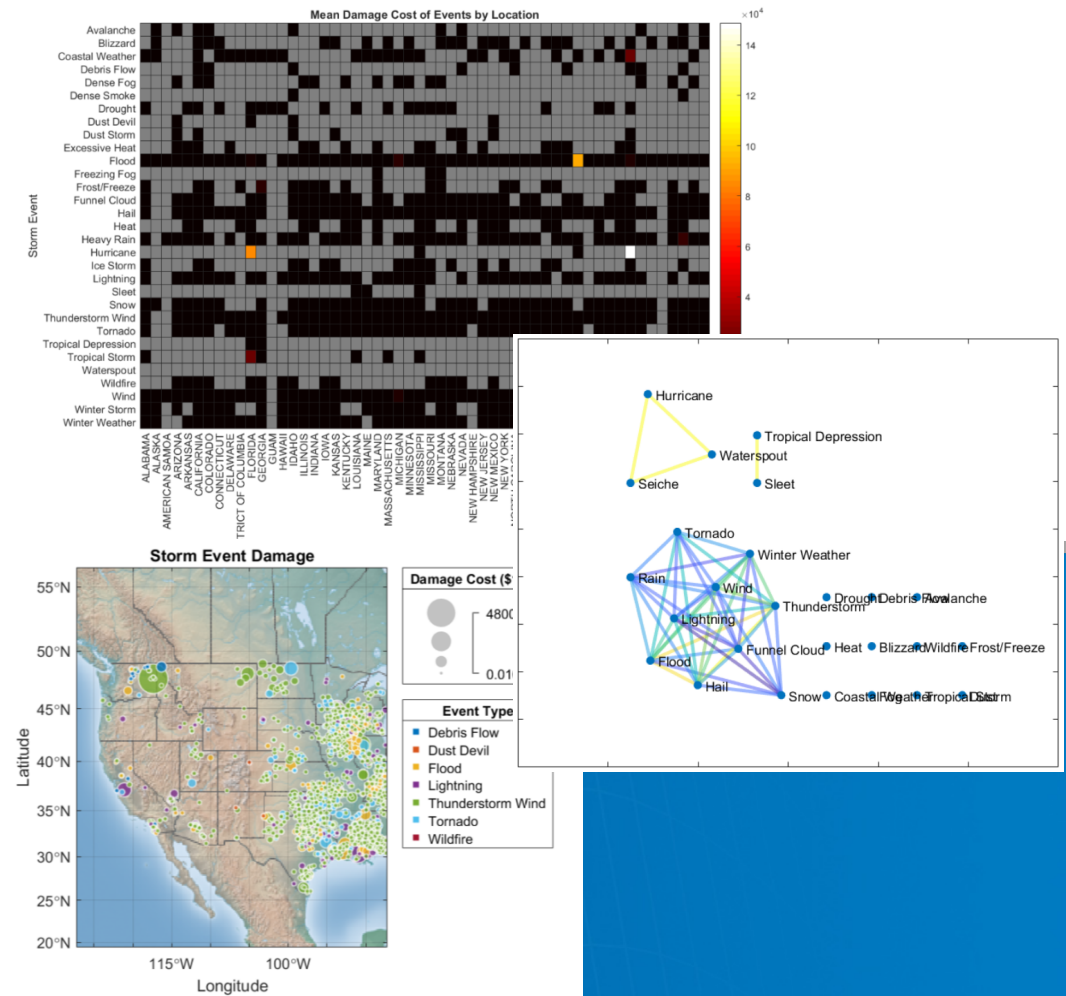


# Data Analytics and Machine Learning With MATLAB

Sumit Tandon  
 Senior Customer Success Manager, MathWorks, Inc.  
[Sumit.tandon@mathworks.com](mailto:Sumit.tandon@mathworks.com)



# Why MATLAB?

## Focus on solving your problems



**Productive environment**  
tuned for engineering and scientific work



**Ready to use**  
with toolboxes that work out of the box



**Ready to run** on  
production systems  
without rewriting code




**Reliable**  
entrusted to send a spacecraft  
to Pluto, create certified code  
for medical devices



**Execution speed**  
with optimized code that  
leverages GPUs, clusters, and  
clouds

# Complementary, Interactive, Self-paced MATLAB Tutorials

Ideal for new users or a refresher




FREE

## MATLAB Onramp

Get started quickly with the basics of MATLAB.

---

Launch



NEW

FREE

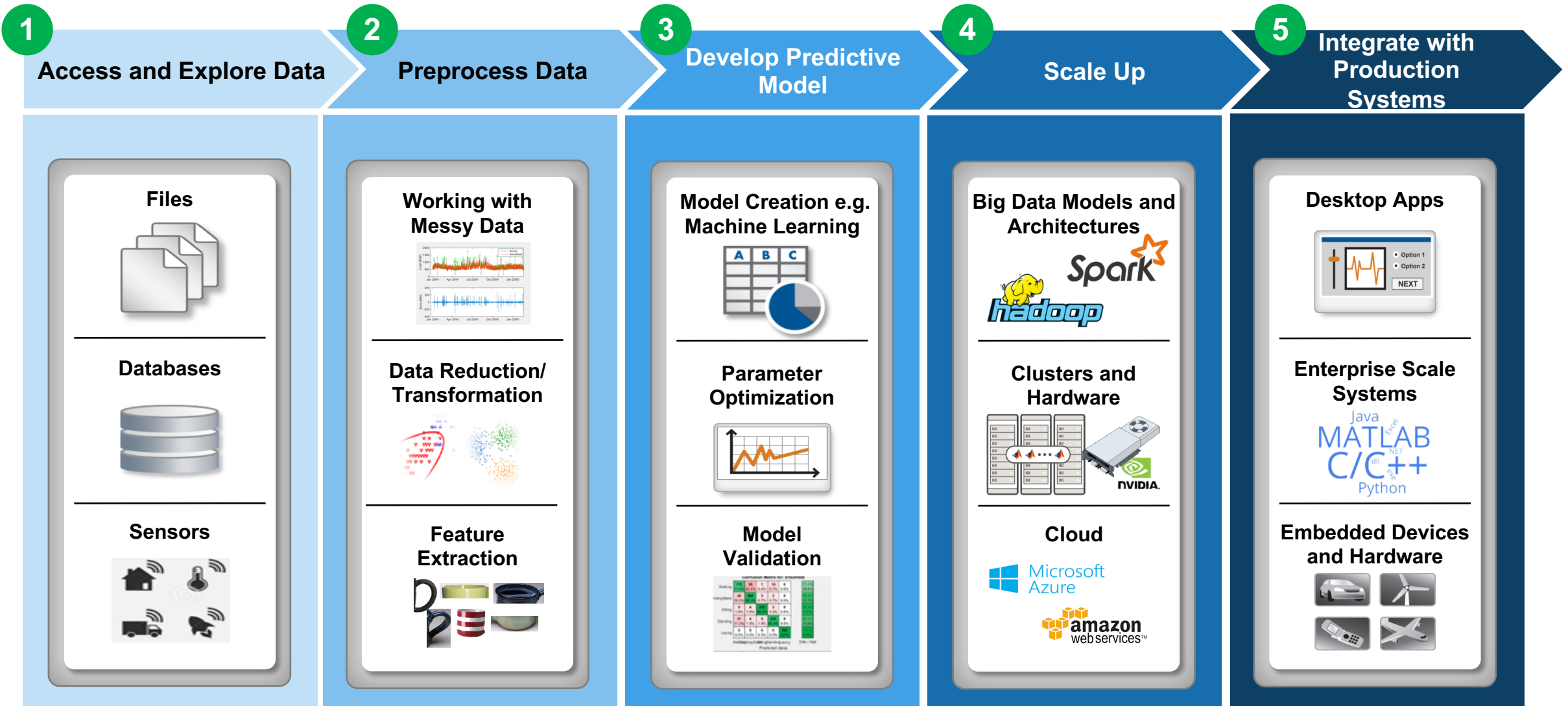
## Deep Learning Onramp

Get started quickly using deep learning methods to perform image recognition.

---

Launch

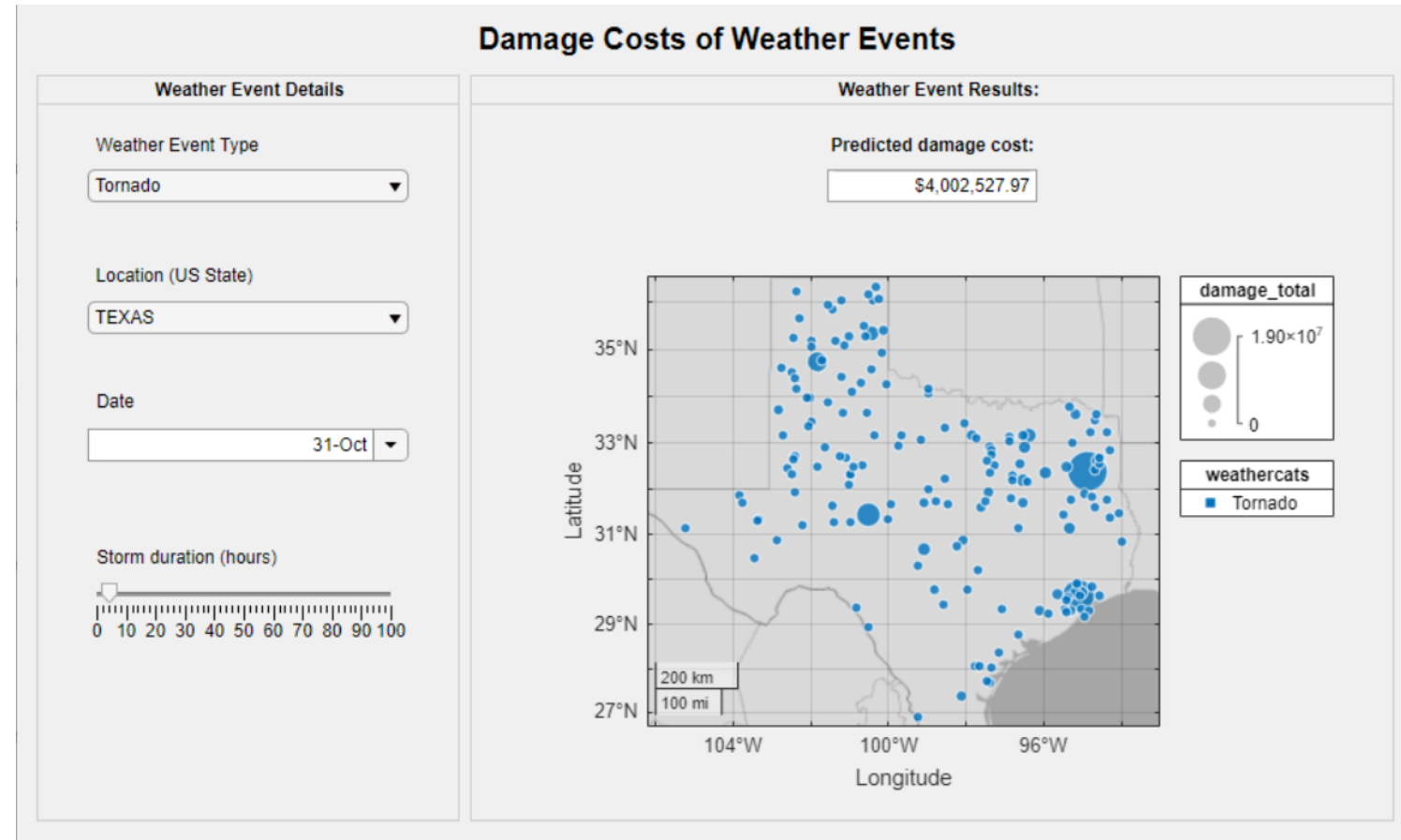
# Data Analytics Workflow





# Demo: Predict Damage Cost of Weather Events

- Use historical weather events data from 1980-2017
- Preprocess data
- Develop prediction model based on event type, location, time/month/year
- Predict damage value to prepare for future

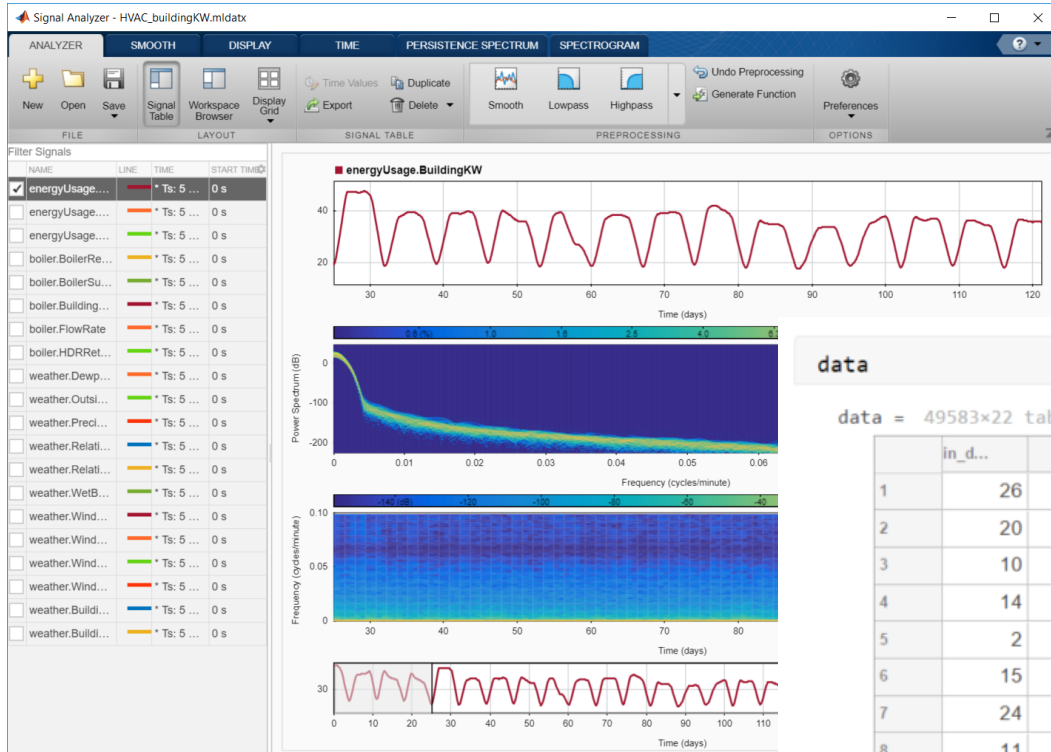




2

## Preprocess Data

## Spend less time cleaning data



```
data = sortrows(data);
data = fillmissing(data, 'linear');
data = smoothdata(data);
```

data

data = 49583x22 table | Reduced from 49962 rows

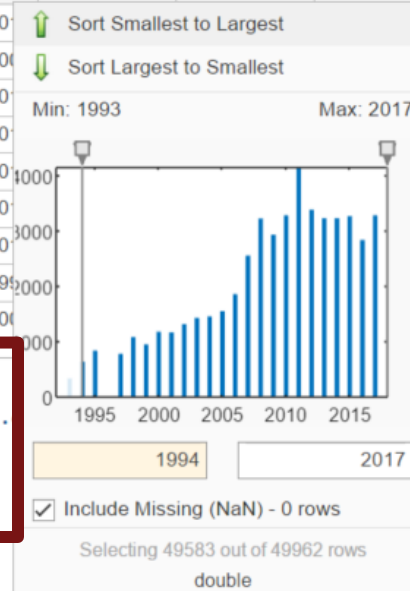
	in_d...	end_day	event_id	state	T	year	event_t...	end_tim...	damage...	T	damage...	begin_lat	begin_lo
1	26	26	324445	MISSOURI	20							39.9400	-92
2	20	20	181522	KANSAS	20							39.3500	-101
3	10	10	333022	ARIZONA	20							33.9583	-109
4	14	14	254237	WISCONSIN	20							NaN	
5	2	2	592975	GEORGIA	20							32.6600	-81
6	15	15	707180	GEORGIA	20							32.6328	-83
7	24	24	447971	NEBRASKA	20							41.2100	-96
8	11	11	5621848	COLORADO	19							38.1000	-103
9	2	2	181446	<undefined>	20							38.5641	-77

Code ^

```
data = data(data.damage_crops <= 4548.0398 | ismissing(data.
data = data(data.year >= 1994 | ismissing(data.year),:)
```

Update Code

Copy



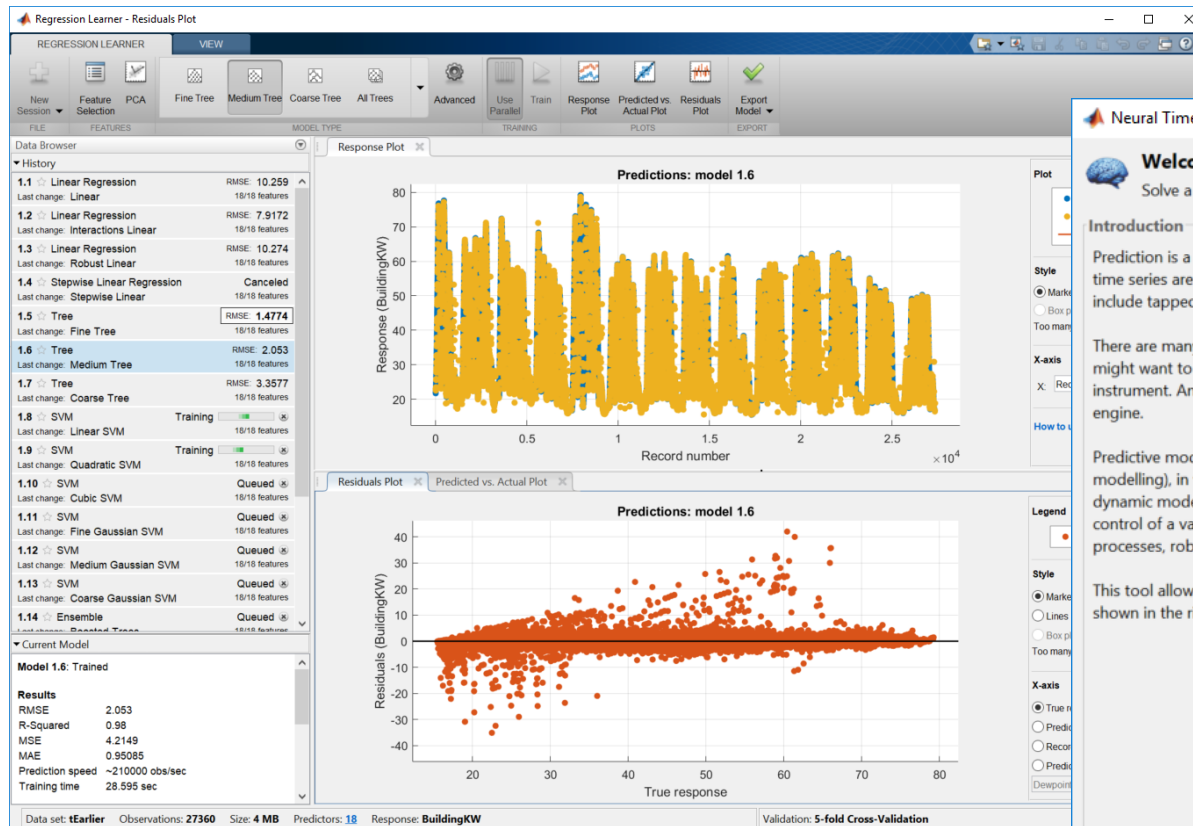
3

## Develop Predictive Models

## Get started easily with advanced techniques

Classification  
Regression

## Deep Learning



The screenshot shows the Neural Time Series (ntstool) app interface. The top toolbar includes options for New Session, Feature Selection, PCA, Fine Tree, Medium Tree, Coarse Tree, All Trees, Advanced, Use Parallel, Train, Response Plot, Predicted vs. Actual Plot, Residuals Plot, and Export Model. The main plot area displays two plots: a Response Plot (Predictions: model 1.6) and a Residuals Plot (Predictions: model 1.6). The bottom status bar indicates the data set is 'Earlier', observations are 27360, size is 4 MB, predictors are 18, response is Building kW, and validation is 5-fold Cross-Validation.

**Welcome to the Neural Network Time Series app.**  
Solve a nonlinear time series problem with a dynamic neural network.

**Introduction**

Prediction is a kind of dynamic filtering, in which past values of one or more time series are used to predict future values. Dynamic neural networks, which include tapped delay lines are used for nonlinear filtering and prediction.

There are many applications for prediction. For example, a financial analyst might want to predict the future value of a stock, bond or other financial instrument. An engineer might want to predict the impending failure of a jet engine.

Predictive models are also used for system identification (or dynamic modelling), in which you build dynamic models of physical systems. These dynamic models are important for analysis, simulation, monitoring and control of a variety of systems, including manufacturing systems, chemical processes, robotics and aerospace systems.

This tool allows you to solve three kinds of nonlinear time series problems shown in the right panel. Choose one and click [Next].

**Select a Problem**

☒ Nonlinear Autoregressive with External (Exogenous) Input (NARX)  
Predict series  $y(t)$  given  $d$  past values of  $y(t)$  and another series  $x(t)$ .

$x(t) \rightarrow y(t) = f(y(t-1), \dots, y(t-d), x(t-1), \dots, x(t-d))$

☐ Nonlinear Autoregressive (NAR)  
Predict series  $y(t)$  given  $d$  past values of  $y(t)$ .

$y(t) = f(y(t-1), \dots, y(t-d))$

☐ Nonlinear Input-Output  
Predict series  $y(t)$  given  $d$  past values of series  $x(t)$ .

**Important Note:** NARX solutions are more accurate than this solution. Only use this solution if past values of  $y(t)$  will not be available when deployed.

$x(t) \rightarrow y(t) = f(x(t-1), \dots, x(t-d))$

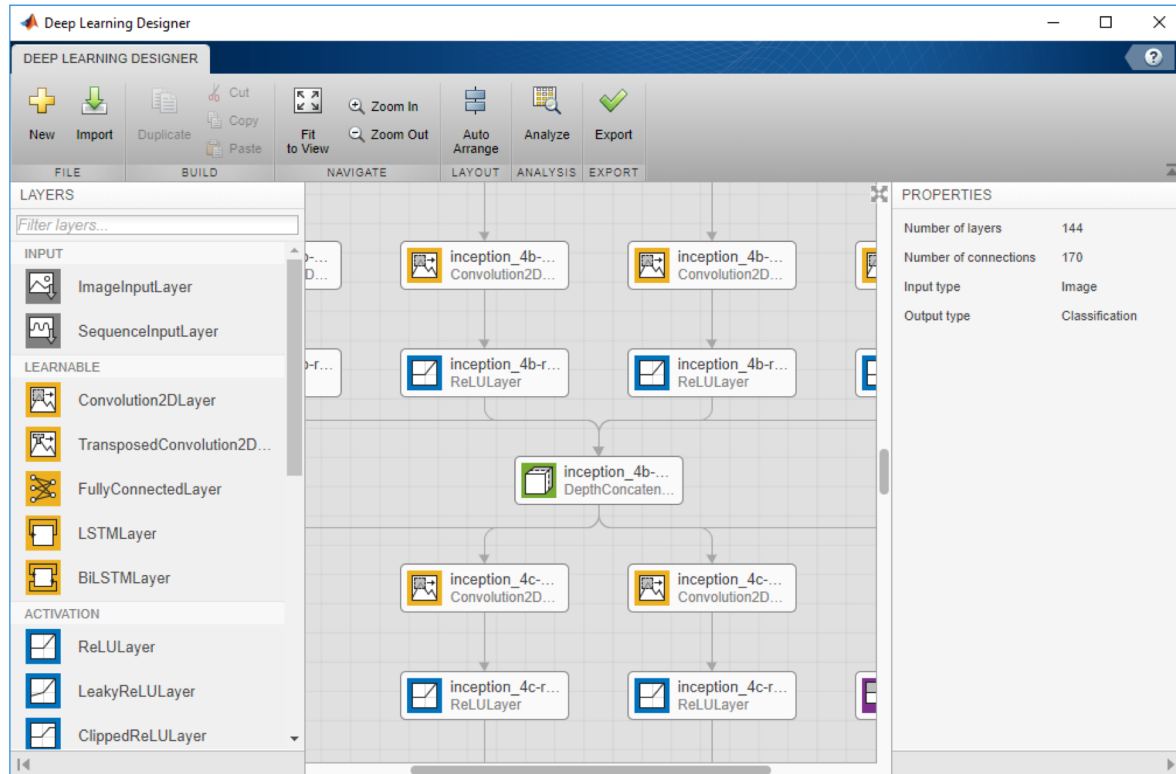
To continue, click [Next].

Neural Network Start Welcome Back Next Cancel

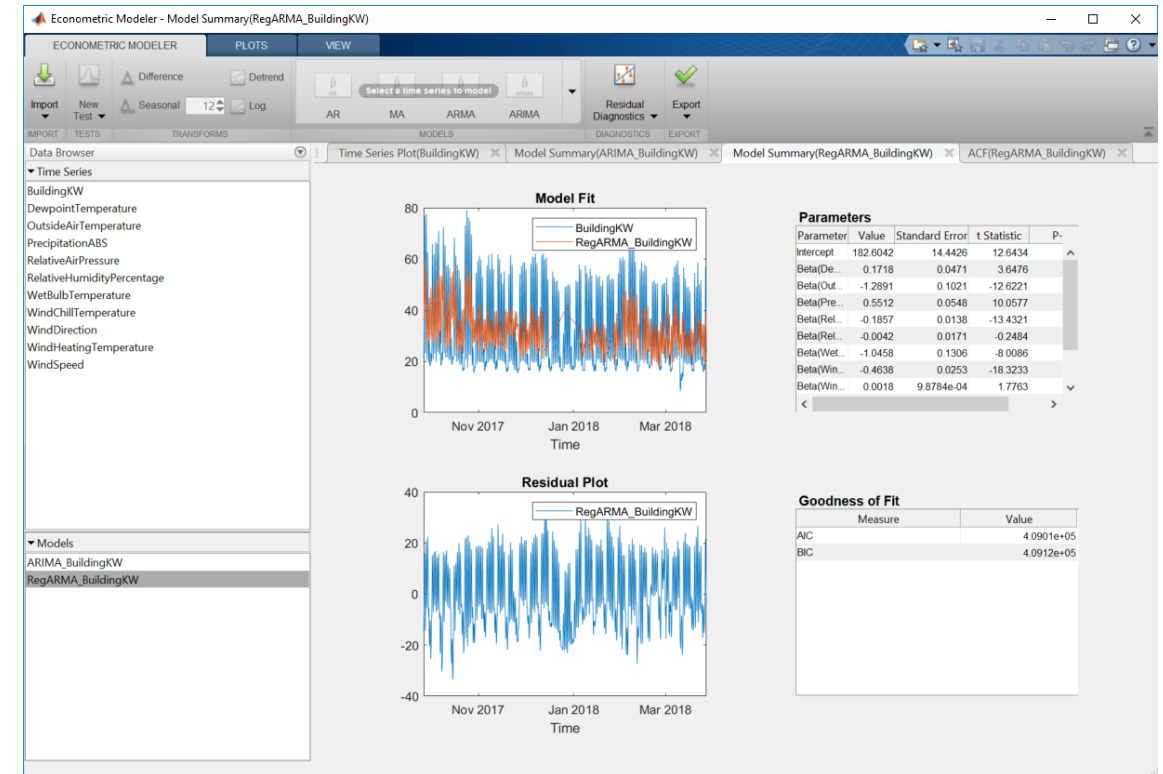


# Explore different types of models

## Neural Networks



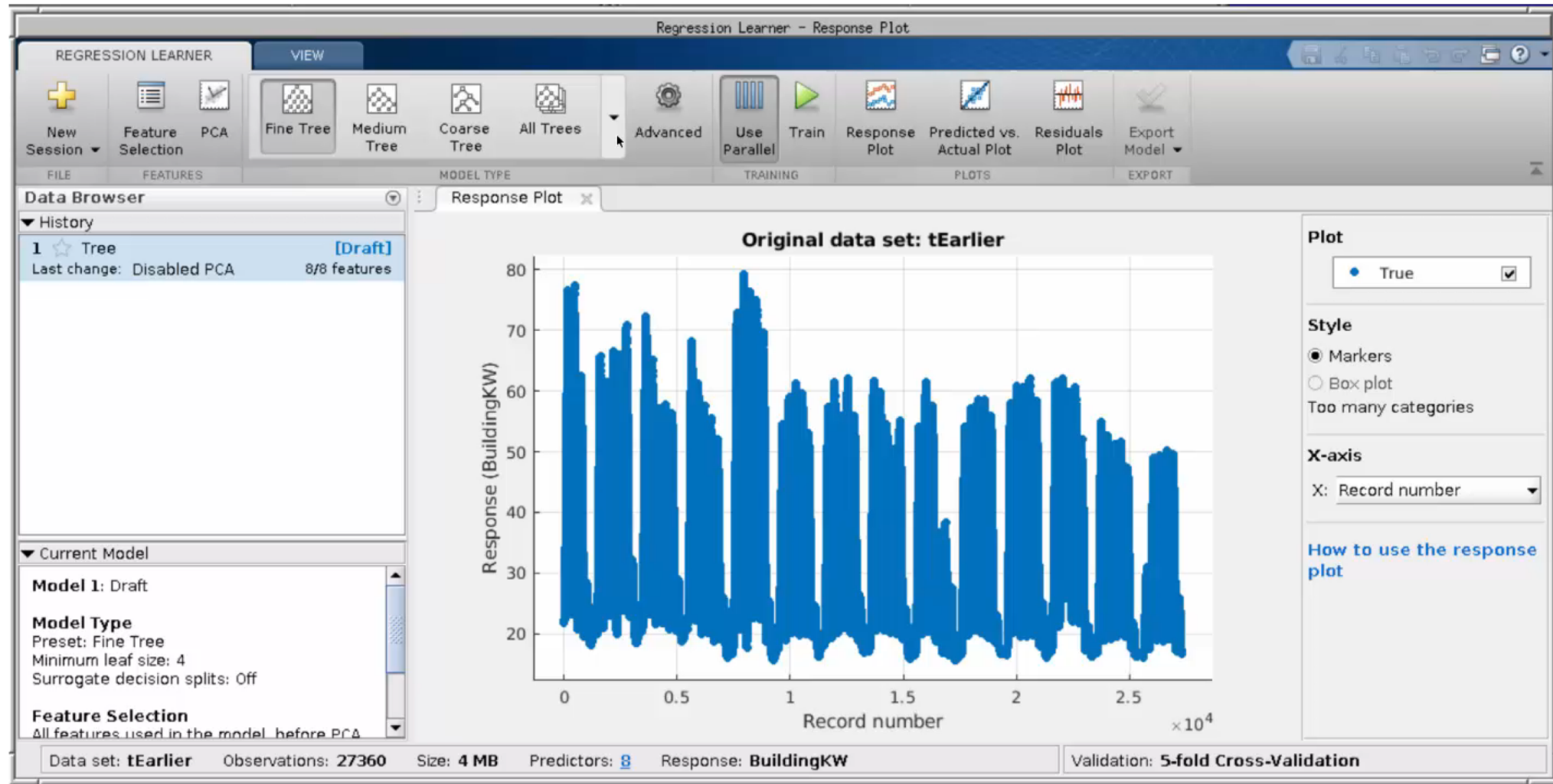
## Time Series Models



3

Develop Predictive  
Models

Try many algorithms in parallel and validate results



# Scale to big data using the same code

## One file

### Access Data

```
measured = readtable('PumpData.csv');  
measured = table2timetable(measured);
```

### Preprocess Data

#### Select data of interest

```
measured = measured(timerange(seconds(1),seconds(2)), 'Speed')
```

#### Work with missing data

```
measured = fillmissing(measured, 'linear');
```

#### Calculate statistics

```
m = mean(measured.Speed);  
s = std(measured.Speed);
```

## One hundred files

### Access Data

```
measured = datastore('PumpData*.csv');  
measured = tall(measured);  
measured = table2timetable(measured);
```

### Preprocess Data

#### Select data of interest

```
measured = measured(timerange(seconds(1),seconds(2)), 'Speed')
```

#### Work with missing data

```
measured = fillmissing(measured, 'linear');
```

#### Calculate statistics

```
m = mean(measured.Speed);  
s = std(measured.Speed);
```

```
[m,s] = gather(m,s);
```

4

Scale Up

## Access data from anywhere with minimal changes

```
setenv('AWS_ACCESS_KEY_ID', 'ACCESS_KEY_ID')  
setenv('AWS_SECRET_ACCESS_KEY', 'ACCESS_KEY')  
setenv('AWS_REGION', 'us-east-1')
```

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

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



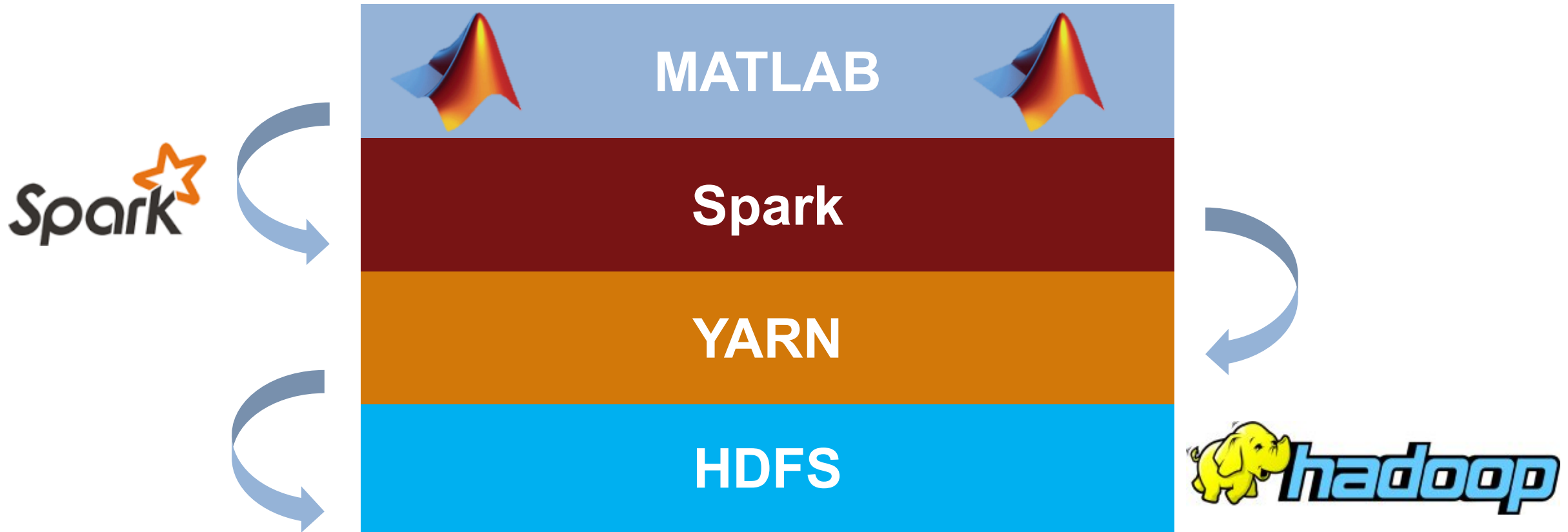
Local disk



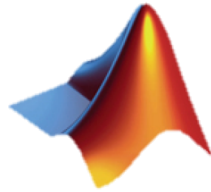
4

Scale Up

Use MATLAB on a Spark-enabled Hadoop cluster for machine learning at scale



# Monitor large jobs in MATLAB



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

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)	<a href="#">runJob at SparkIntegContext.java:662</a>	2017/09/17 15:11:22	31 s	0/1	<div><div></div></div> 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)	<a href="#">runJob at SparkIntegContext.java:662</a>	2017/09/17 15:09:30	1.9 min	1/1	<div><div></div></div> 131/131
8 (MATLAB_Pass_8)	<a href="#">runJob at SparkIntegContext.java:662</a>	2017/09/17 15:05:17	4.2 min	1/1	<div><div></div></div> 276/276
7 (MATLAB_Pass_7)	<a href="#">runJob at SparkIntegContext.java:662</a>	2017/09/17 14:59:11	6.1 min	1/1	<div><div></div></div> 382/382
6 (MATLAB_Pass_6)	<a href="#">runJob at SparkIntegContext.java:662</a>	2017/09/17 14:57:55	1.3 min	1/1	<div><div></div></div> 89/89
5 (MATLAB_Pass_5)	<a href="#">runJob at SparkIntegContext.java:662</a>	2017/09/17 14:52:18	1.9 min	1/1	<div><div></div></div> 131/131

# Parallel and Distributed Computing



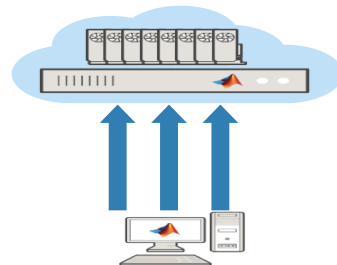
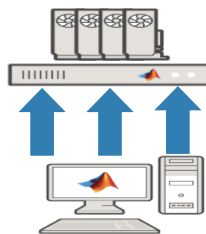
Single  
CPU



Single CPU  
Single GPU



Single CPU, Multiple GPUs



## Parallel Computing Toolbox

- Speed up parallel applications
- Take advantage of GPUs
- Prototype code for clusters

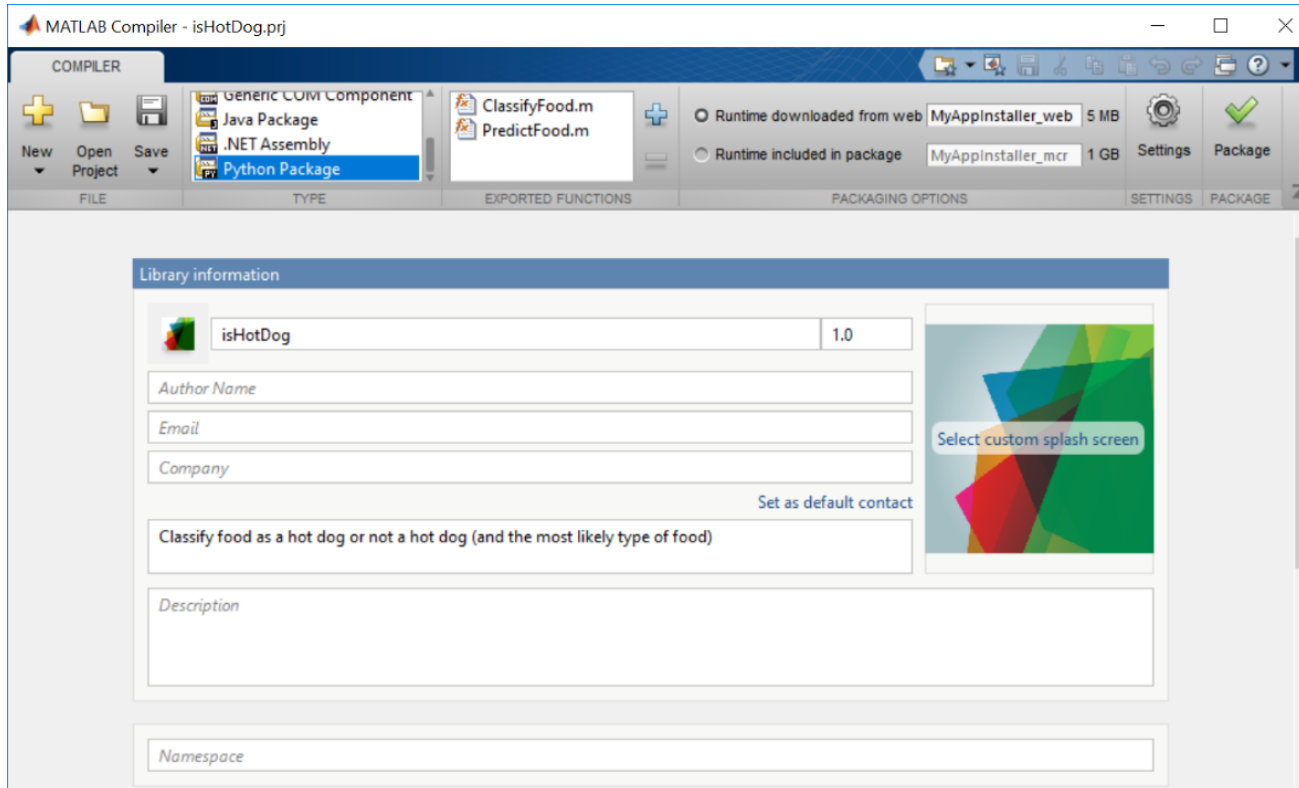
## MATLAB Distributed Computing Server

- Scale up computation

5

Integrate with  
Production Systems

# Package and deploy model to run anywhere

Enterprise  
Applications

Dashboards



Devices



Cloud, IOT

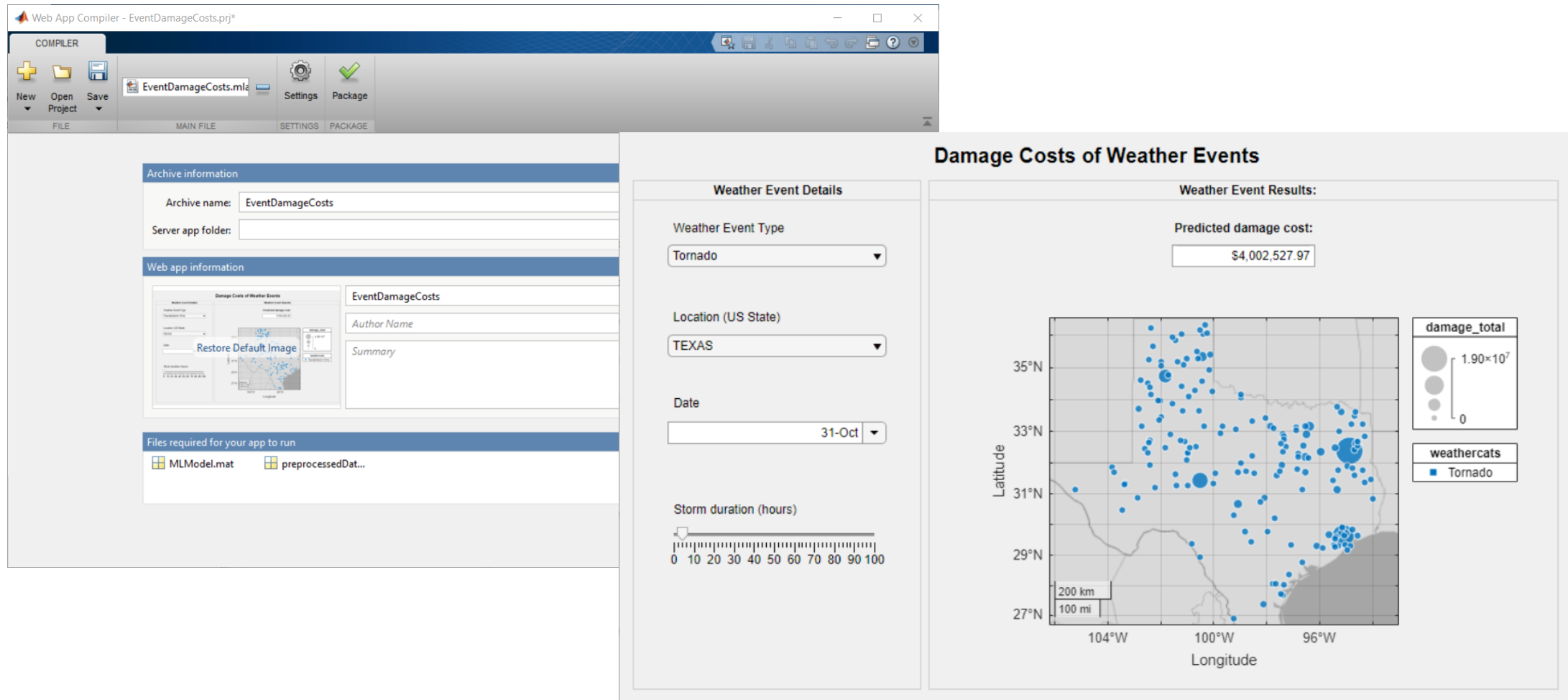




5

Integrate with  
Production Systems

## Create web application



Web App Compiler - EventDamageCosts.prj\*

COMPLER

New Open Save EventDamageCosts.mlx Settings Package

FILE MAIN FILE SETTINGS PACKAGE

Archive information

Archive name: EventDamageCosts

Server app folder:

Web app information

EventDamageCosts

Author Name

Summary

Files required for your app to run

MLModel.mat preprocessedDat...

Weather Event Details

Weather Event Type: Tornado

Location (US State): TEXAS

Date: 31-Oct

Storm duration (hours): 0 10 20 30 40 50 60 70 80 90 100

Weather Event Results:

Predicted damage cost: \$4,002,527.97

Latitude: 35°N 33°N 31°N 29°N 27°N

Longitude: 104°W 100°W 96°W

damage\_total: 1.90x10<sup>7</sup> 0

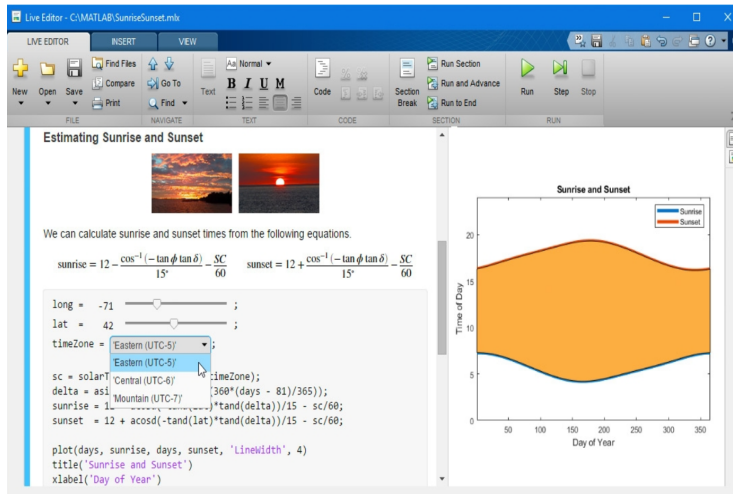
weathercats: Tornado

5

Integrate with  
Production Systems

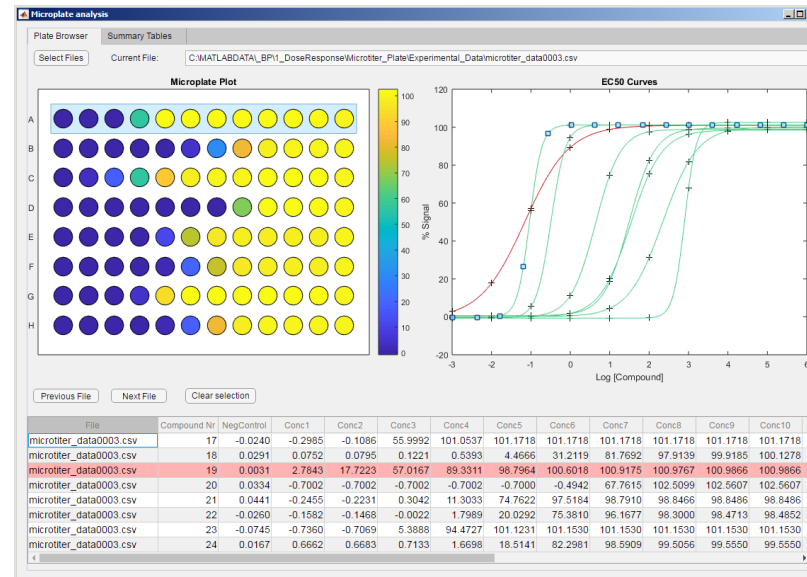
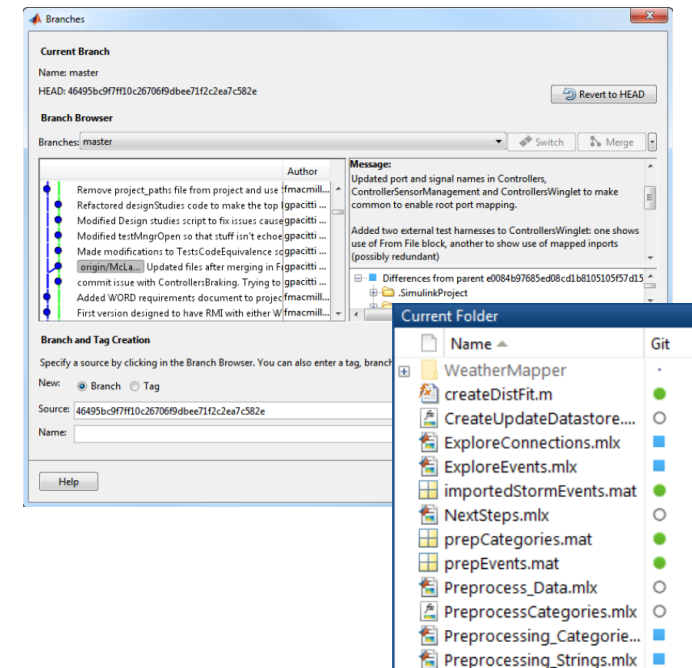
## Share your discoveries

## Document and publish results



.pdf, html, LaTeX

## Create apps

Use source control  
(GitHub, SVN)

# Running MATLAB on Cedar/Graham/Niagara

- <https://docs.computecanada.ca/wiki/MATLAB>
- <https://docs.computecanada.ca/wiki/Cedar>
- <https://docs.computecanada.ca/wiki/Graham>
- <https://docs.computecanada.ca/wiki/Niagara>

Q & A