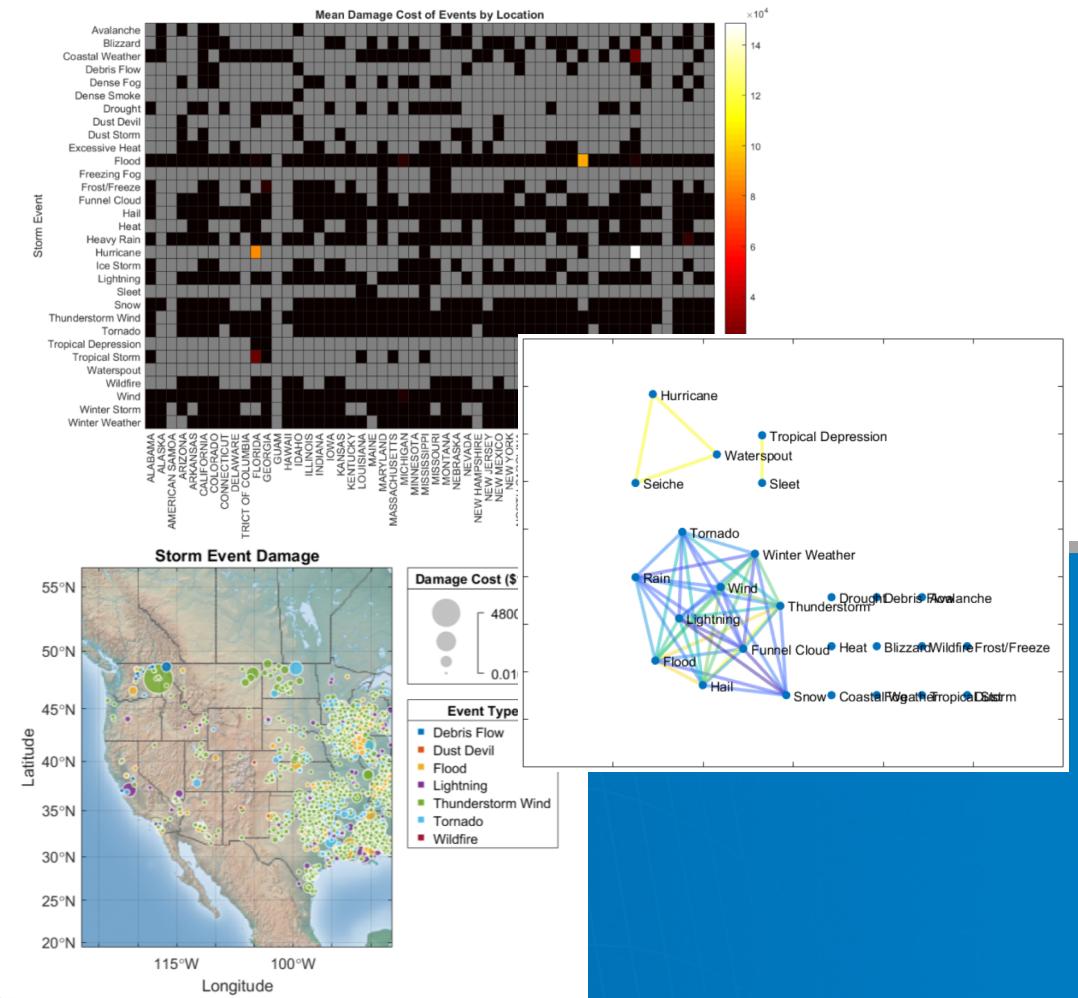


# Data Analytics and Machine Learning With MATLAB

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



## MATLAB Onramp

Get started quickly with the basics of MATLAB.

[Launch](#)

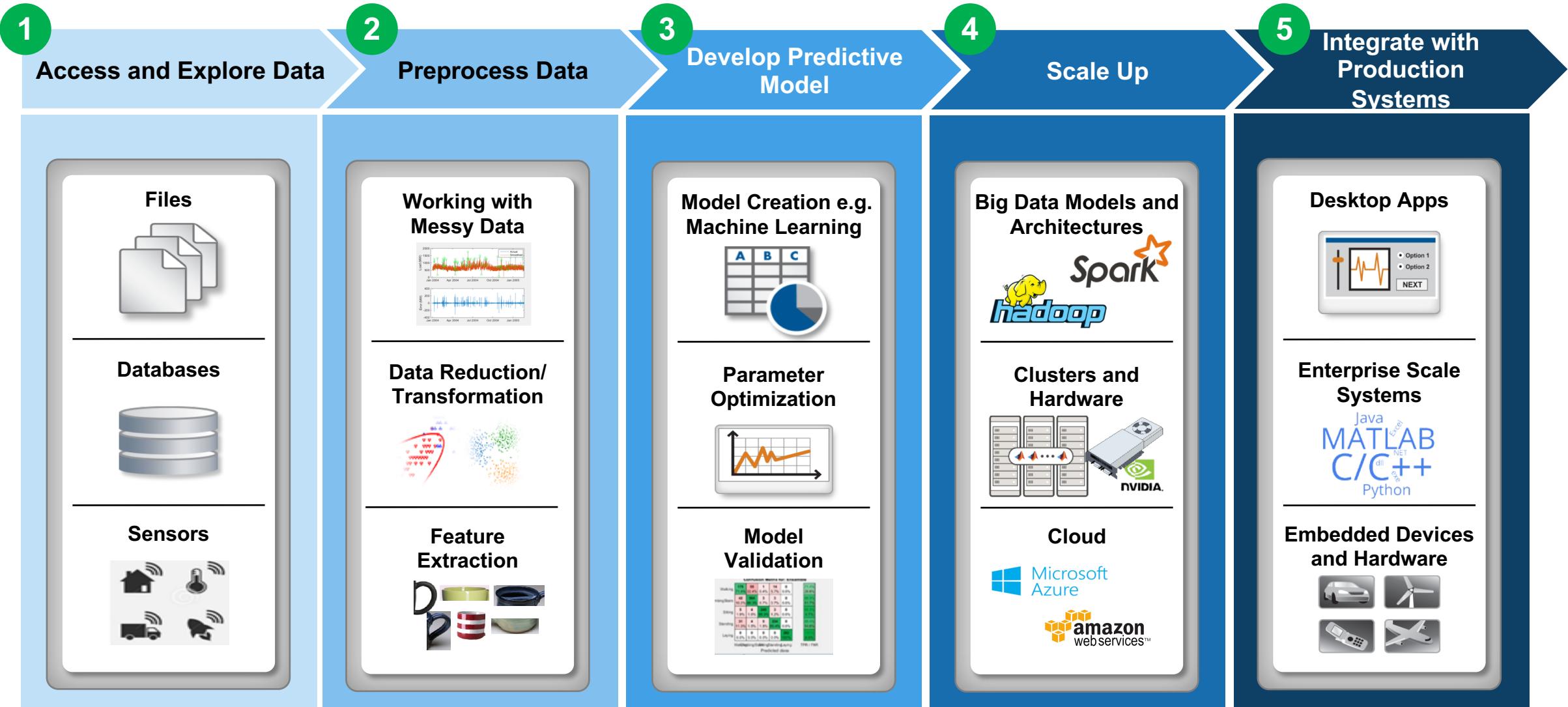


## Deep Learning Onramp

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

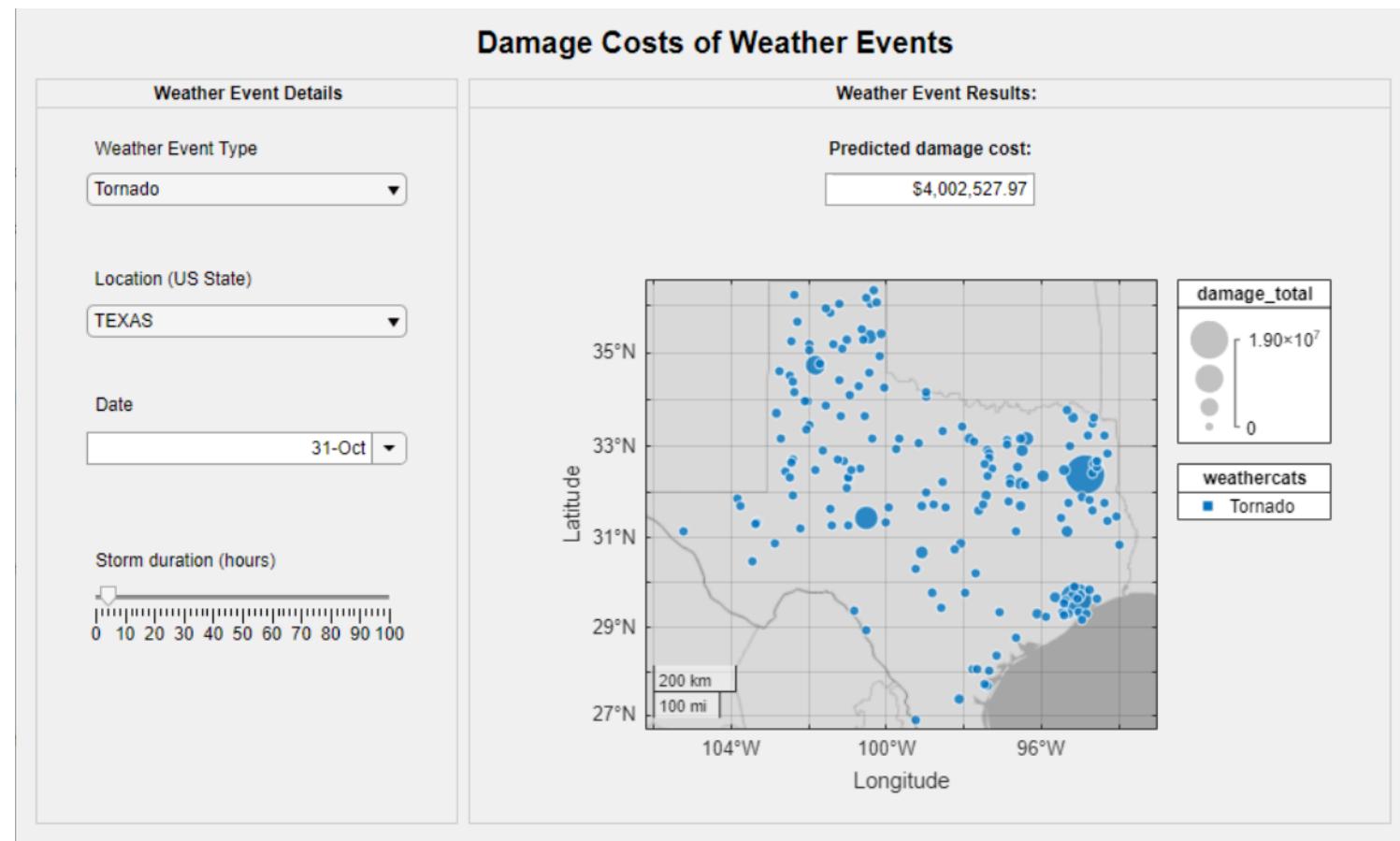
[Launch](#)

# 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



1

Access and Explore Data

# Access data from many sources

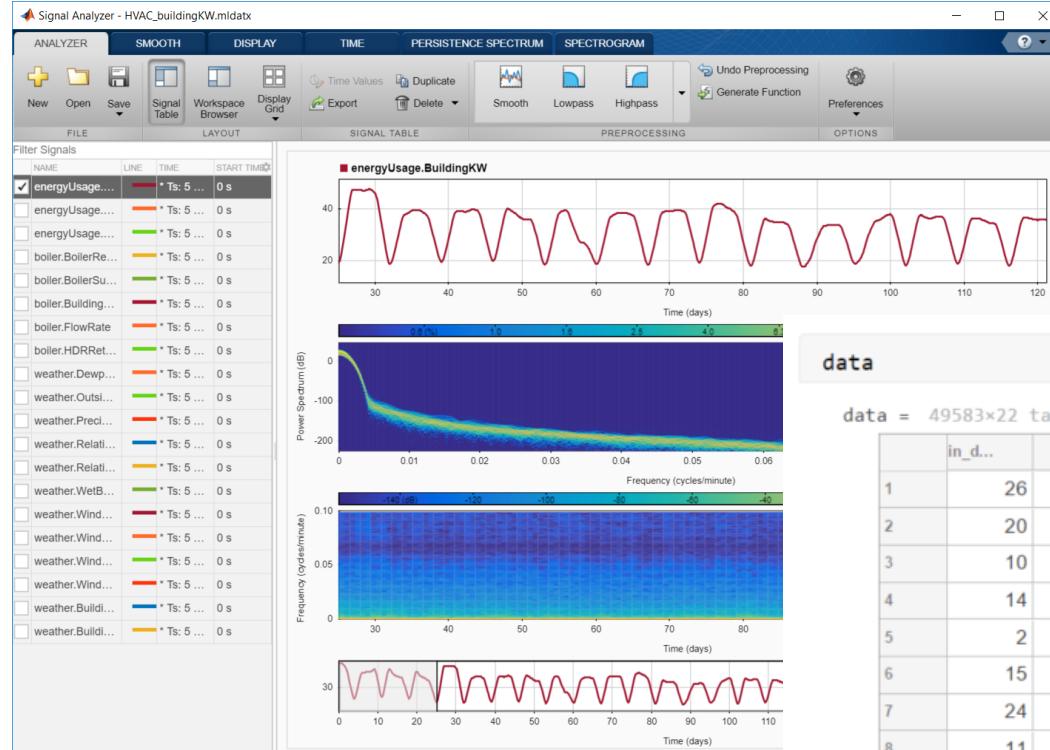


The screenshot shows the MATLAB Import Data tool window titled "Import - C:\MATLAB\StormEvents\_Mass2015.csv". The "IMPORT" tab is selected. In the "DELIMITERS" section, "Comma" is chosen as the column delimiter. The "IMPORTED DATA" section shows a preview of the "StormEvents\_Mass2015" table with columns: month\_name, event\_type, cz\_type, cz\_fips, cz\_name, wfo, end\_date\_time, cz\_timezone, injuries\_dir..., injuries\_inde..., deaths\_dire..., deaths\_inde..., damage\_pr..., and damage. The "VIEW" tab is also visible at the top.

2

## Preprocess Data

## Spend less time cleaning data



data

data = 49583×22 table | Reduced from 49962 rows

	in_d...	end_day	event_id	state	T	year	event_t...	end_time...	damage...	T	damage...	begin_lat	begin_le...
1	26	26	324445	MISSOURI	20				Sort Smallest to Largest	0	0	39.9400	-92
2	20	20	181522	KANSAS	200				Sort Largest to Smallest	0	0	39.3500	-101
3	10	10	333022	ARIZONA	20				Min: 1993	0	0	33.9583	-109
4	14	14	254237	WISCONSIN	20				Max: 2017	0	0	Nan	
5	2	2	592975	GEORGIA	20					0	0	32.6600	-81
6	15	15	707180	GEORGIA	20					NaN	0	32.6328	-83
7	24	24	447971	NEBRASKA	20					NaN	0	41.2100	-96
8	11	11	5621848	COLORADO	19					NaN	0	38.1000	-103
9	2	2	181446	<undefined>	20					NaN	0	38.5641	-77

Code ^

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

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

 Include Missing (NaN) - 0 rows

Selecting 49583 out of 49962 rows

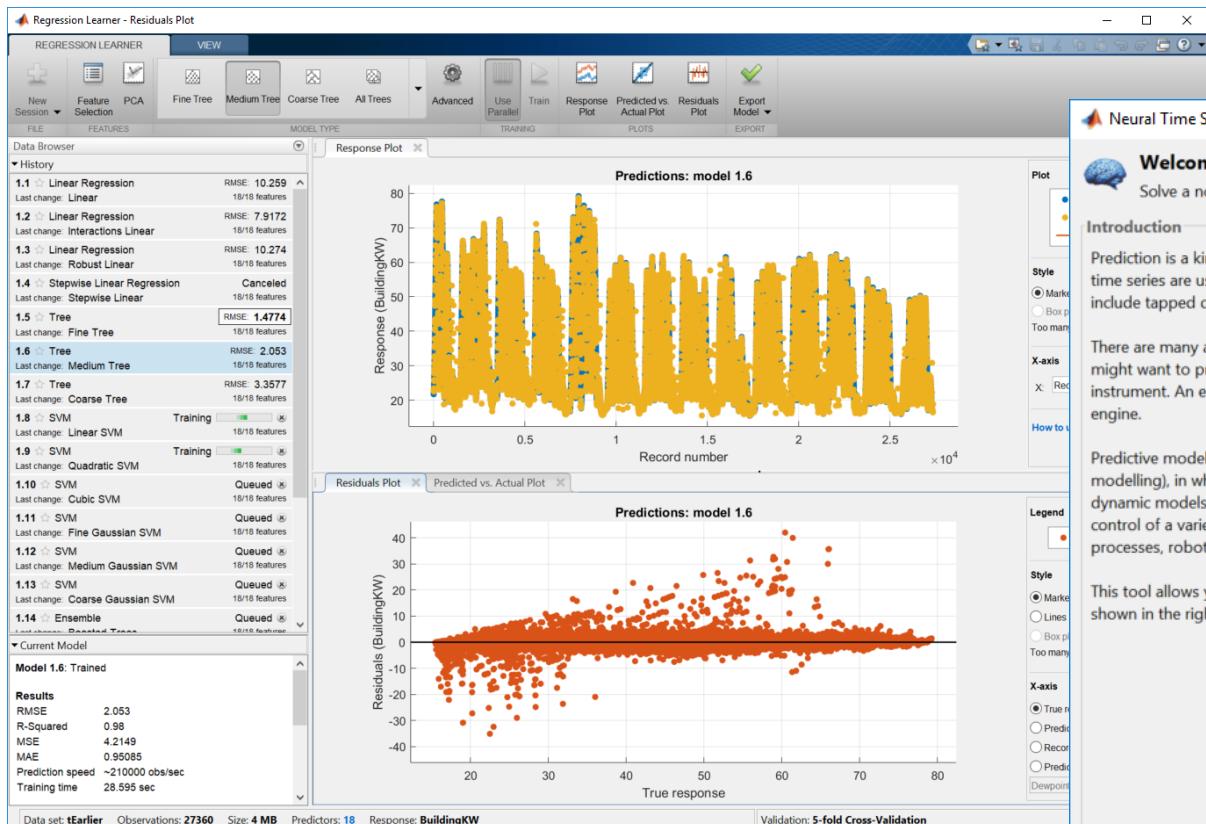
double

3

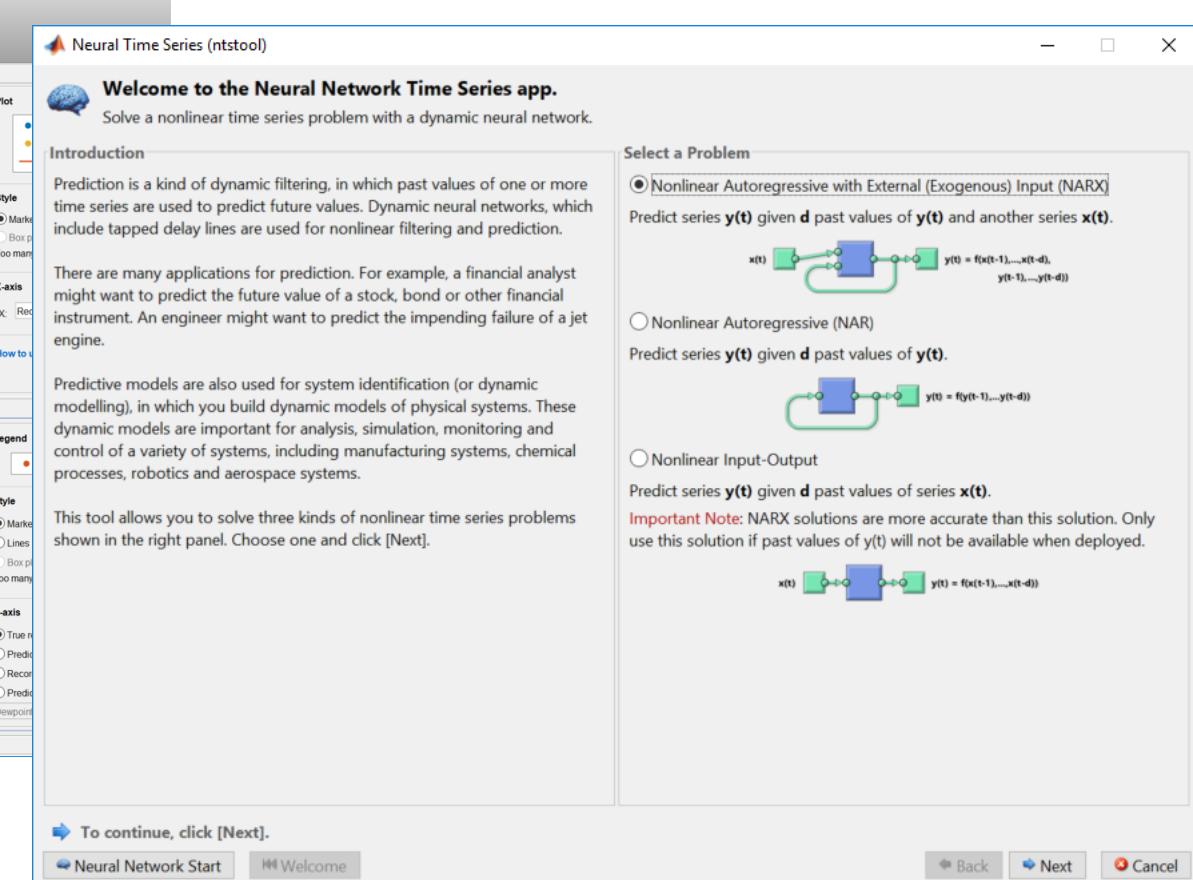
Develop Predictive  
Models

# Get started easily with advanced techniques

## Classification Regression

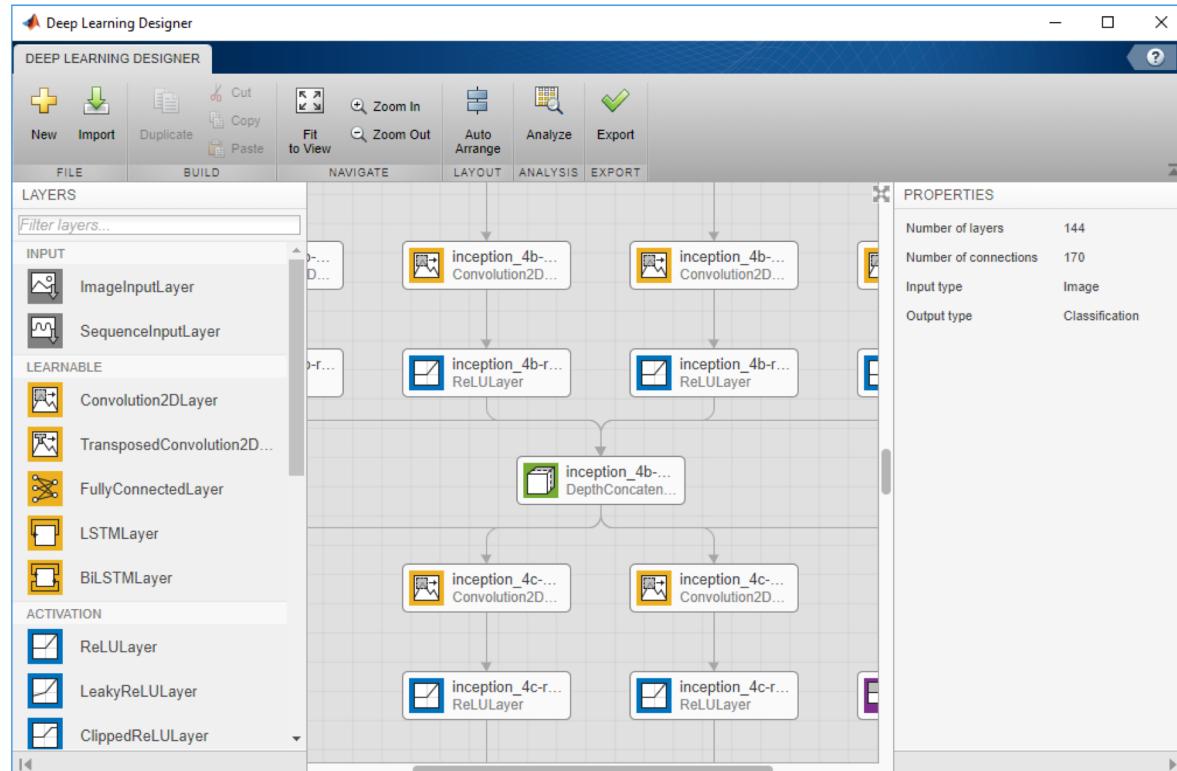


## Deep Learning

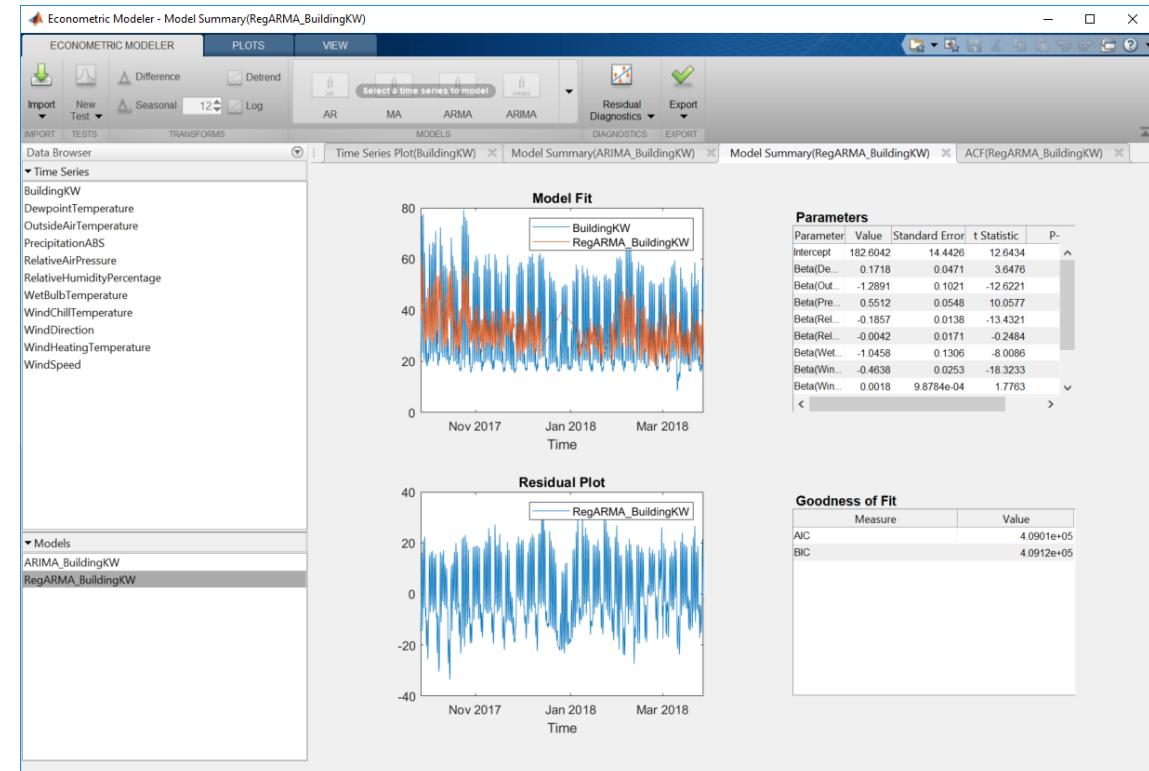


# Explore different types of models

## Neural Networks



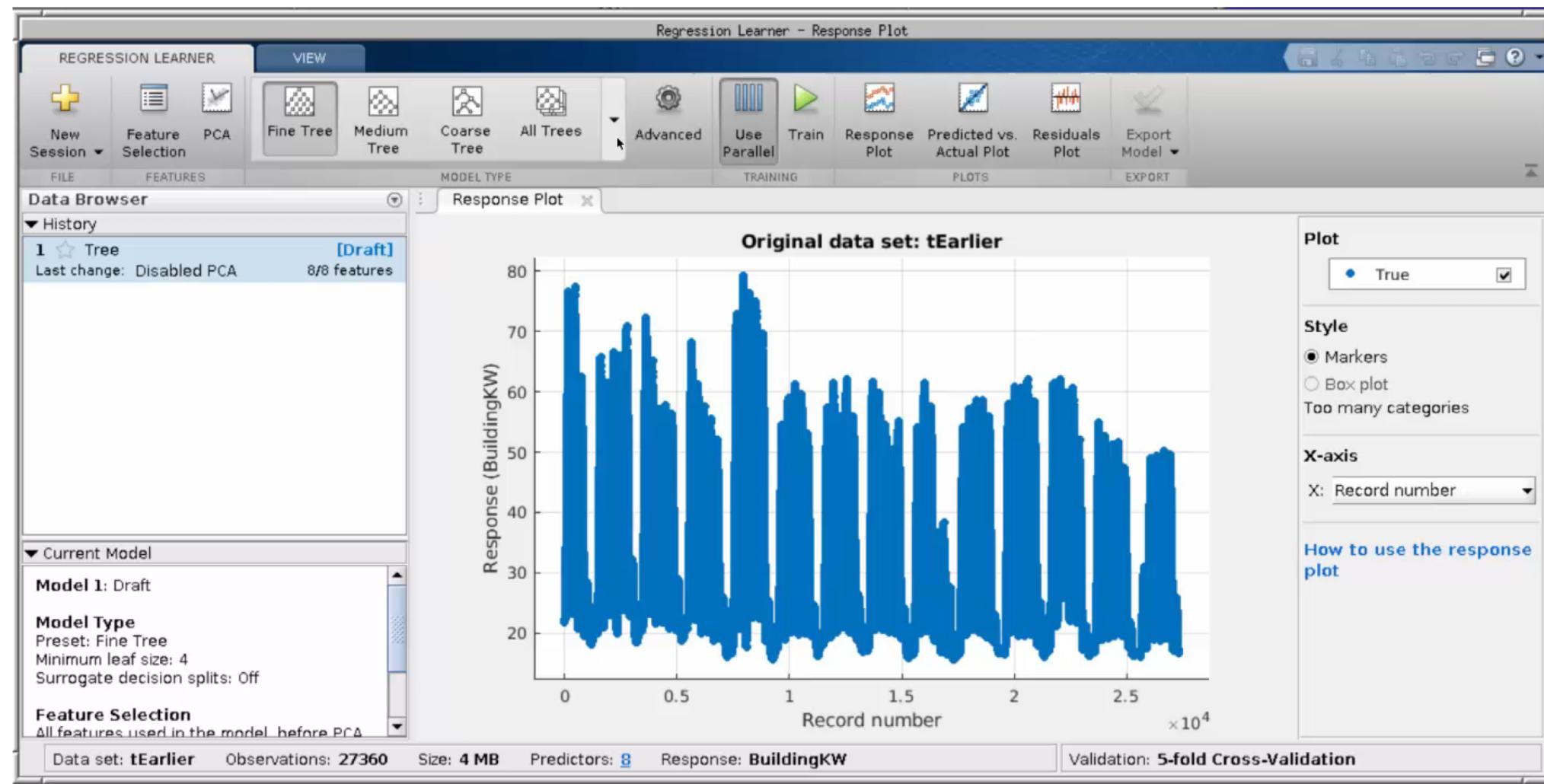
## Time Series Models



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 = readable('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')
% Set AWS REGION to use S3 US West (Oregon)
fileLoc = 'datasets/FoodImages';
```



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

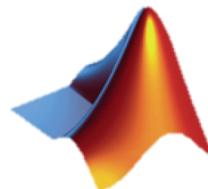
4

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

MATLAB Spark Job application UI

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

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

# Parallel and Distributed Computing



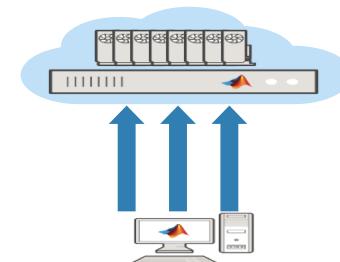
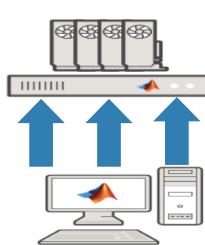
Single  
CPU



Single CPU  
Single GPU



Single CPU, Multiple GPUs



## Parallel Computing Toolbox

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

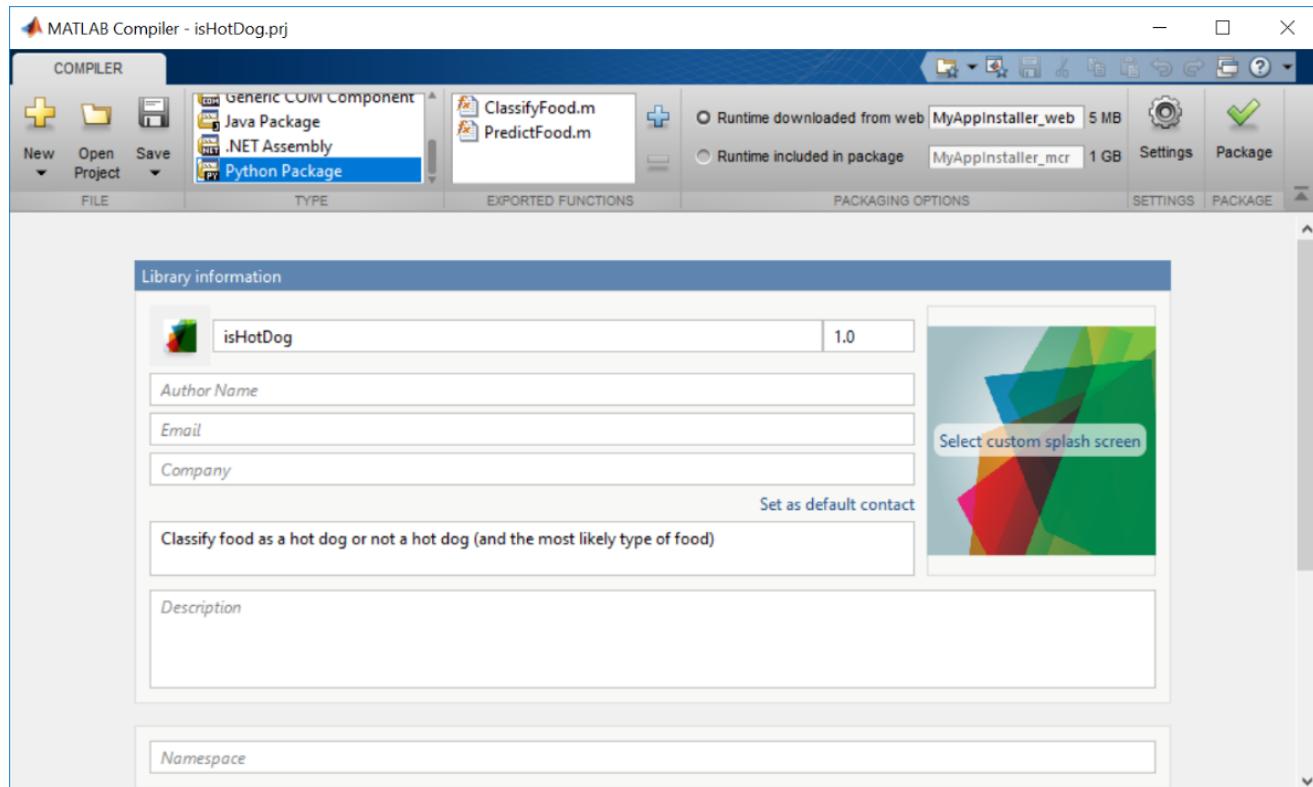
## MATLAB Distributed Computing Server

- 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

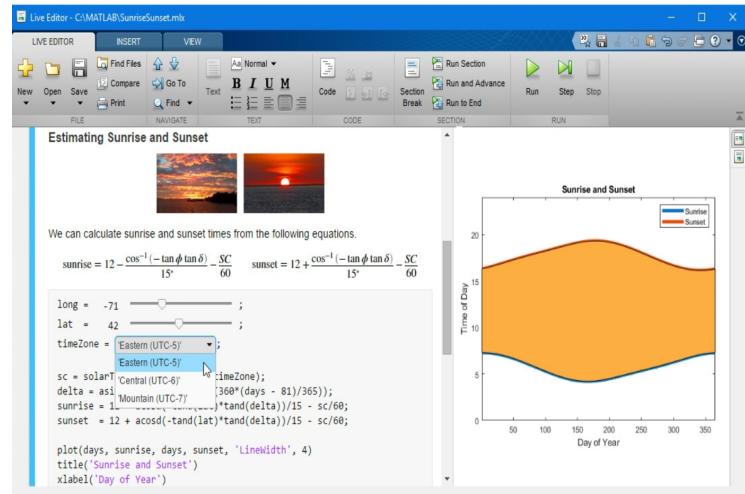
The screenshot shows the MATLAB Web App Compiler interface with the project 'EventDamageCosts.prj\*' open. The left pane displays the 'Archive information' section where the archive name is set to 'EventDamageCosts'. The 'Web app information' section shows a preview of the web application titled 'EventDamageCosts' with a map visualization. The 'Files required for your app to run' section lists 'MLModel.mat' and 'preprocessedDat...'. The main workspace on the right is titled 'Damage Costs of Weather Events' and contains two panels: 'Weather Event Details' and 'Weather Event Results'. The 'Weather Event Details' panel includes dropdowns for 'Weather Event Type' (set to 'Tornado'), 'Location (US State)' (set to 'TEXAS'), and 'Date' (set to '31-Oct'). It also features a slider for 'Storm duration (hours)' ranging from 0 to 100. The 'Weather Event Results' panel displays a predicted damage cost of '\$4,002,527.97'. Below this, a scatter plot shows damage costs across a geographic area from 27°N to 35°N latitude and 96°W to 104°W longitude. The plot includes a legend for 'weathercats' with a single entry 'Tornado' represented by a blue square. A color scale for 'damage\_total' ranges from 0 to  $1.90 \times 10^7$ .

5

Integrate with  
Production Systems

# Share your discoveries

## Document and publish results

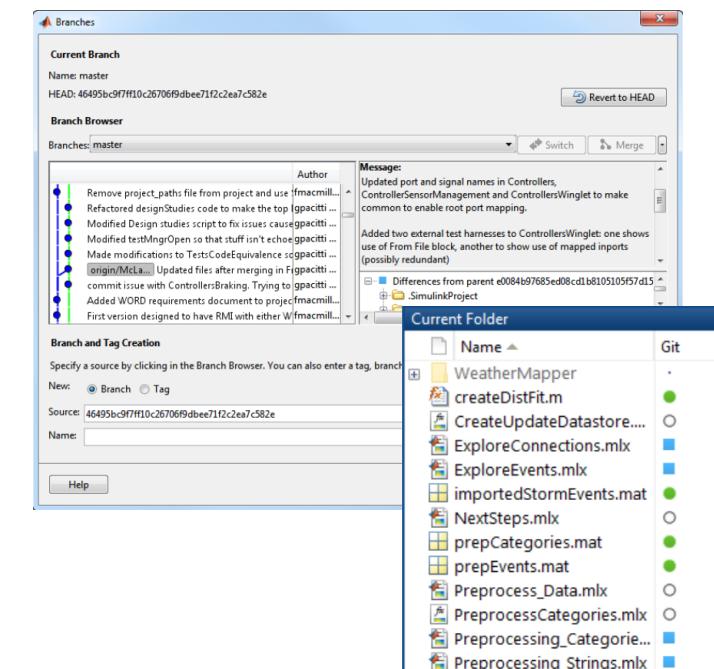


.pdf, html, LaTeX



## 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