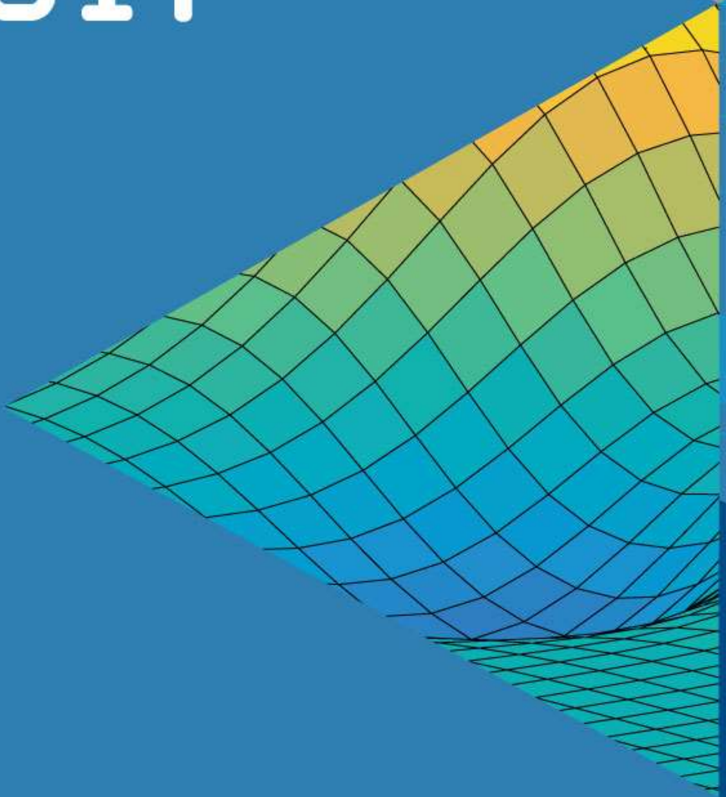
MATLAB EXPO 2017

KOREA

4월 27일, 서울

등록 하기 matlabexpo.co.kr

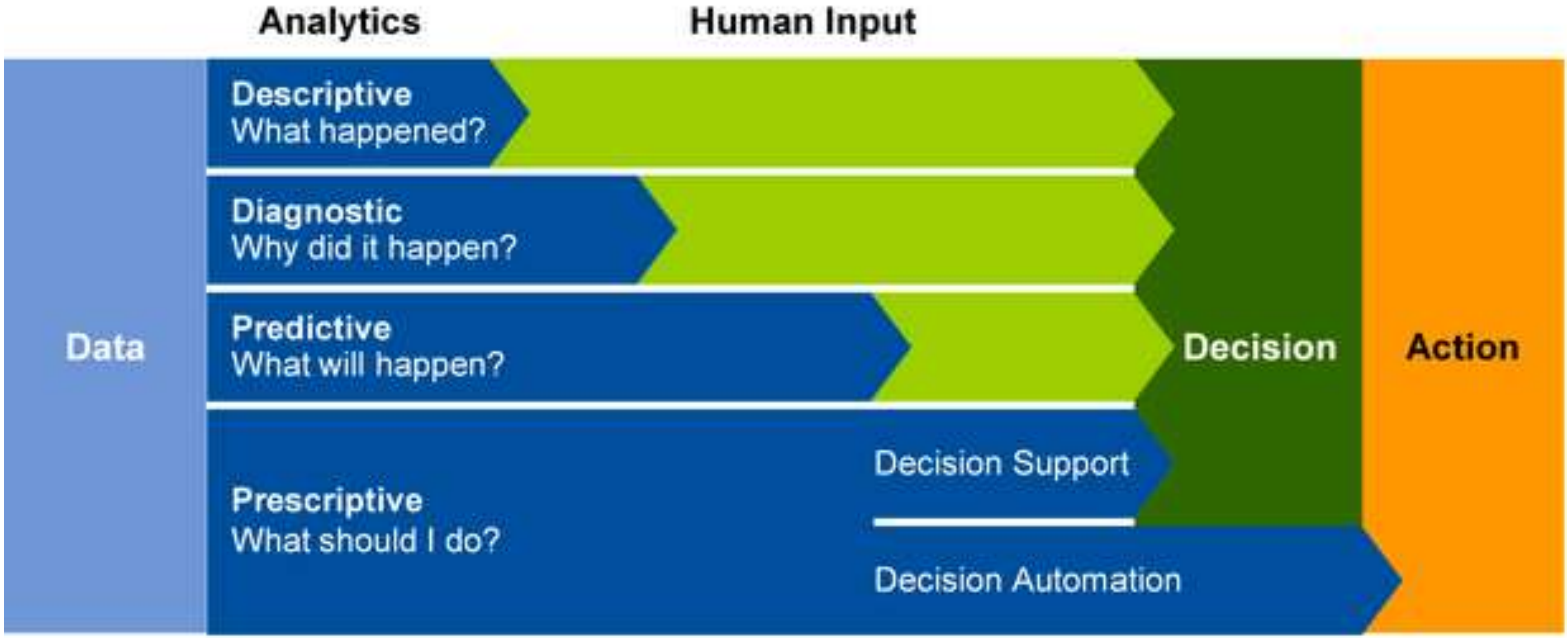


빅데이터 처리 및 머신 러닝 기법

Application Engineer

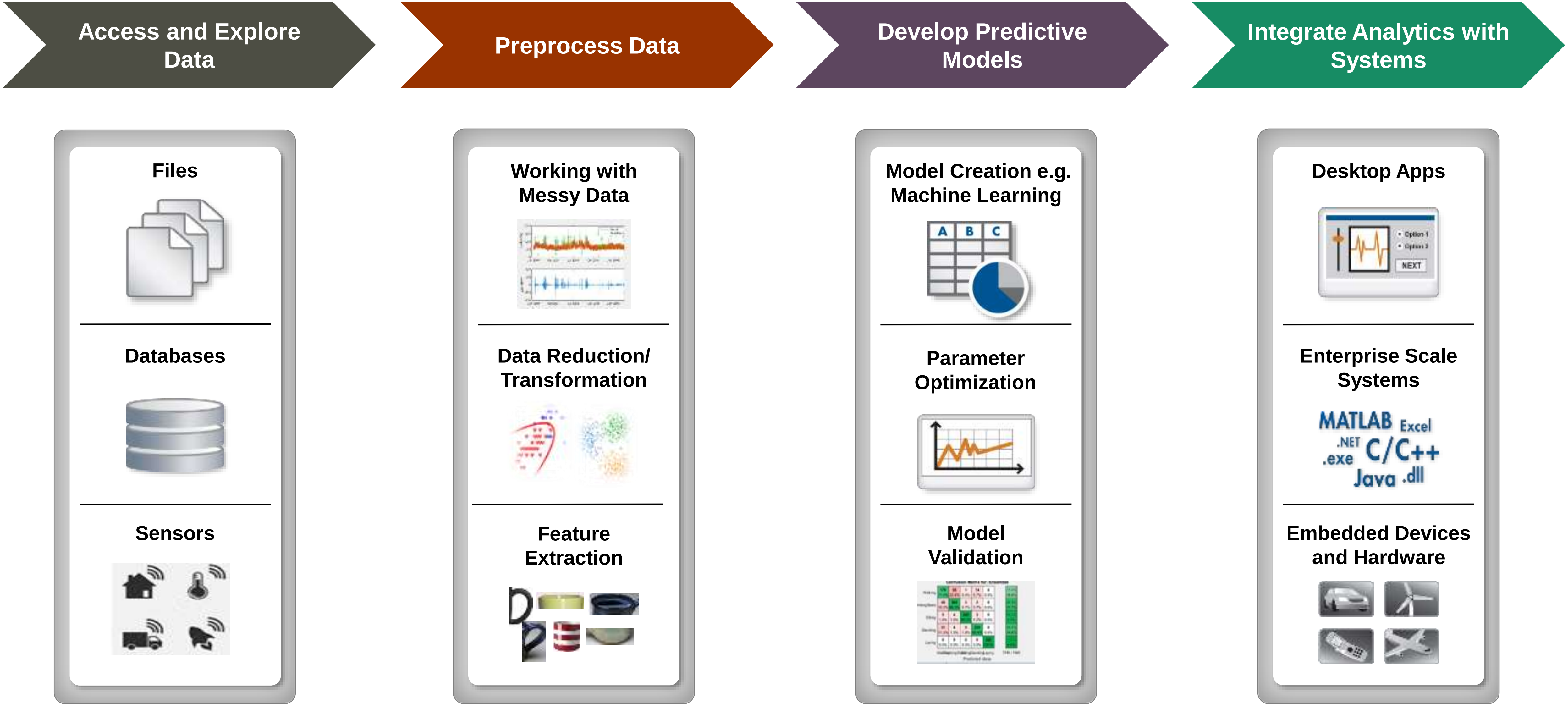
엄준상 과장

Data Analytics



Turn large volumes of complex data into actionable information
source: Gartner

Data Analytics Workflow



Access and Explore Data

- Files
- Databases
- Sensors

Preprocess Data

- Working with Messy Data
- Data Reduction/ Transformation
- Feature Extraction

Develop Predictive Models

- Model Creation e.g. Machine Learning
- Parameter Optimization
- Model Validation

Integrate Analytics with Systems

- Desktop Apps
- Enterprise Scale Systems
- Embedded Devices and Hardware

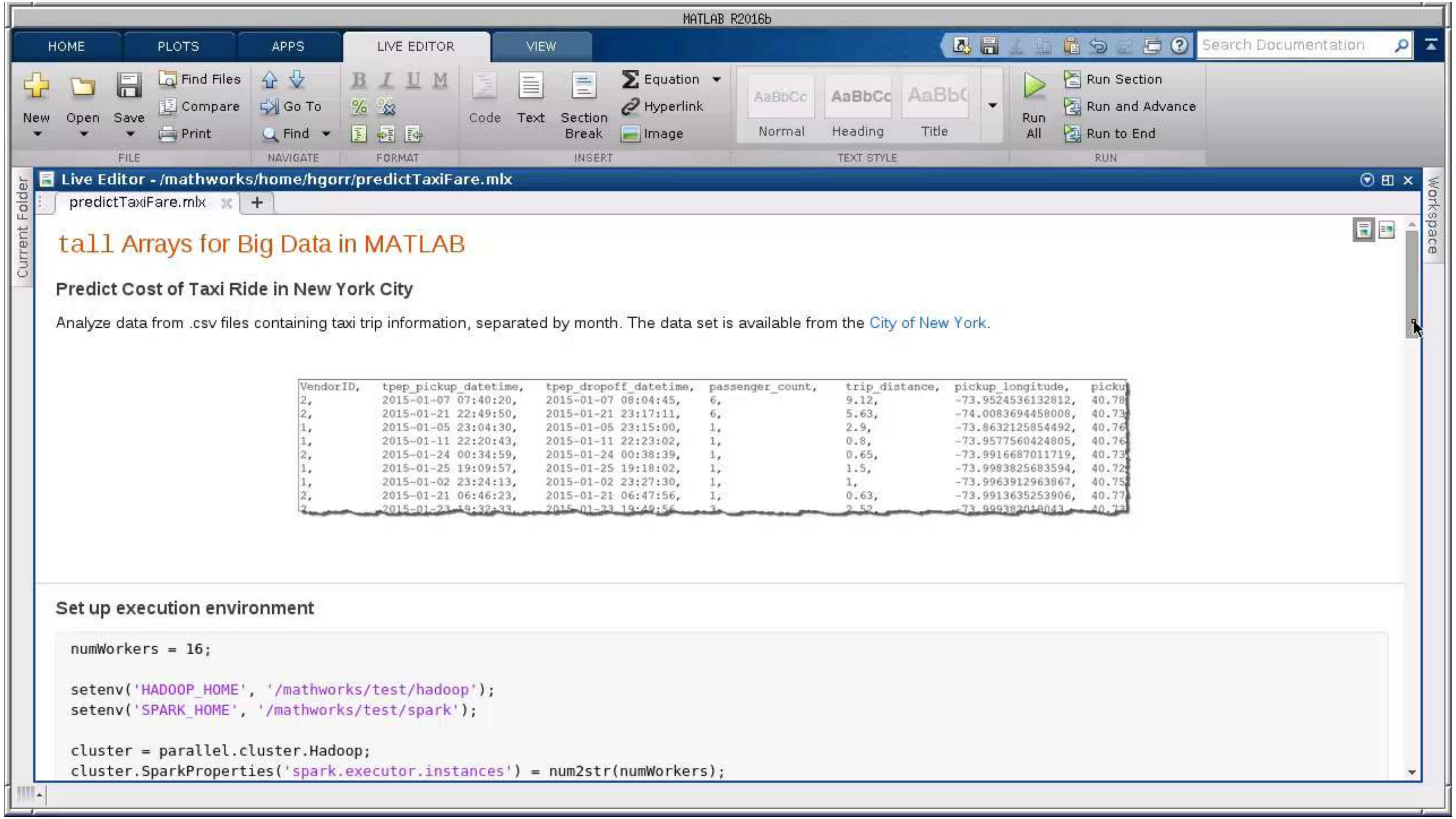
Example: Working with Big Data in MATLAB

- **Objective:** Create a model to predict the cost of a taxi ride in New York City

- **Inputs:**
  - Monthly taxi ride log files
  - The local data set is **small** (~20 MB)
  - The full data set is **big** (~25 GB)

- **Approach:**
  - Acecss Data
  - Preprocess and explore data
  - Develop and validate predictive model (linear fit)
    - Work with subset of data for prototyping
    - Scale to full data set on a cluster

Example: Working with Big Data in MATLAB



tall Arrays for Big Data in MATLAB

Predict Cost of Taxi Ride in New York City

Analyze data from .csv files containing taxi trip information, separated by month. The data set is available from the City of New York.

```
VendorID,   tpep_pickup_datetime,  tpep_dropoff_datetime,  passenger_count,   trip_distance,  pickup_longitude,    picku
2,          2015-01-07 07:40:20,   2015-01-07 08:04:45,    6,                 9.12,           -73.9524536132812,   40.78
2,          2015-01-21 22:49:50,   2015-01-21 23:17:11,    6,                 5.63,           -74.0083694458008,   40.73
1,          2015-01-05 23:04:30,   2015-01-05 23:15:00,    1,                 2.9,            -73.8632125854492,   40.76
1,          2015-01-11 22:20:43,   2015-01-11 22:23:02,    1,                 0.8,            -73.9577560424805,   40.76
2,          2015-01-24 00:34:59,   2015-01-24 00:38:39,    1,                 0.65,           -73.9916687011719,   40.73
1,          2015-01-25 19:09:57,   2015-01-25 19:18:02,    1,                 1.5,            -73.9983825683594,   40.72
1,          2015-01-02 23:24:13,   2015-01-02 23:27:30,    1,                 1,              -73.9963912963867,   40.75
2,          2015-01-21 06:46:23,   2015-01-21 06:47:56,    1,                 0.63,           -73.9913635253906,   40.77
2,          2015-01-23 19:32:33,   2015-01-23 19:49:56,    3,                 2.52,           -73.999382019043,    40.73
```

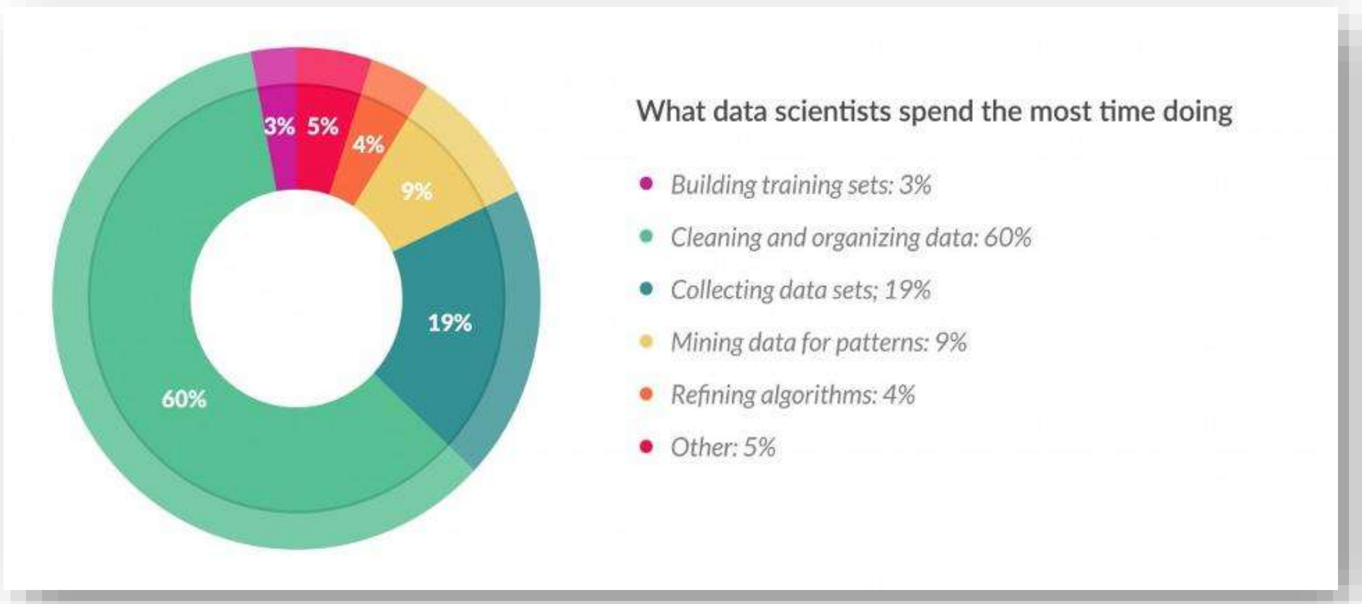
Set up execution environment

```
numWorkers = 16;

setenv('HADOOP_HOME', '/mathworks/test/hadoop');
setenv('SPARK_HOME', '/mathworks/test/spark');

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

Data Access and Pre-processing – Challenges

Challenges

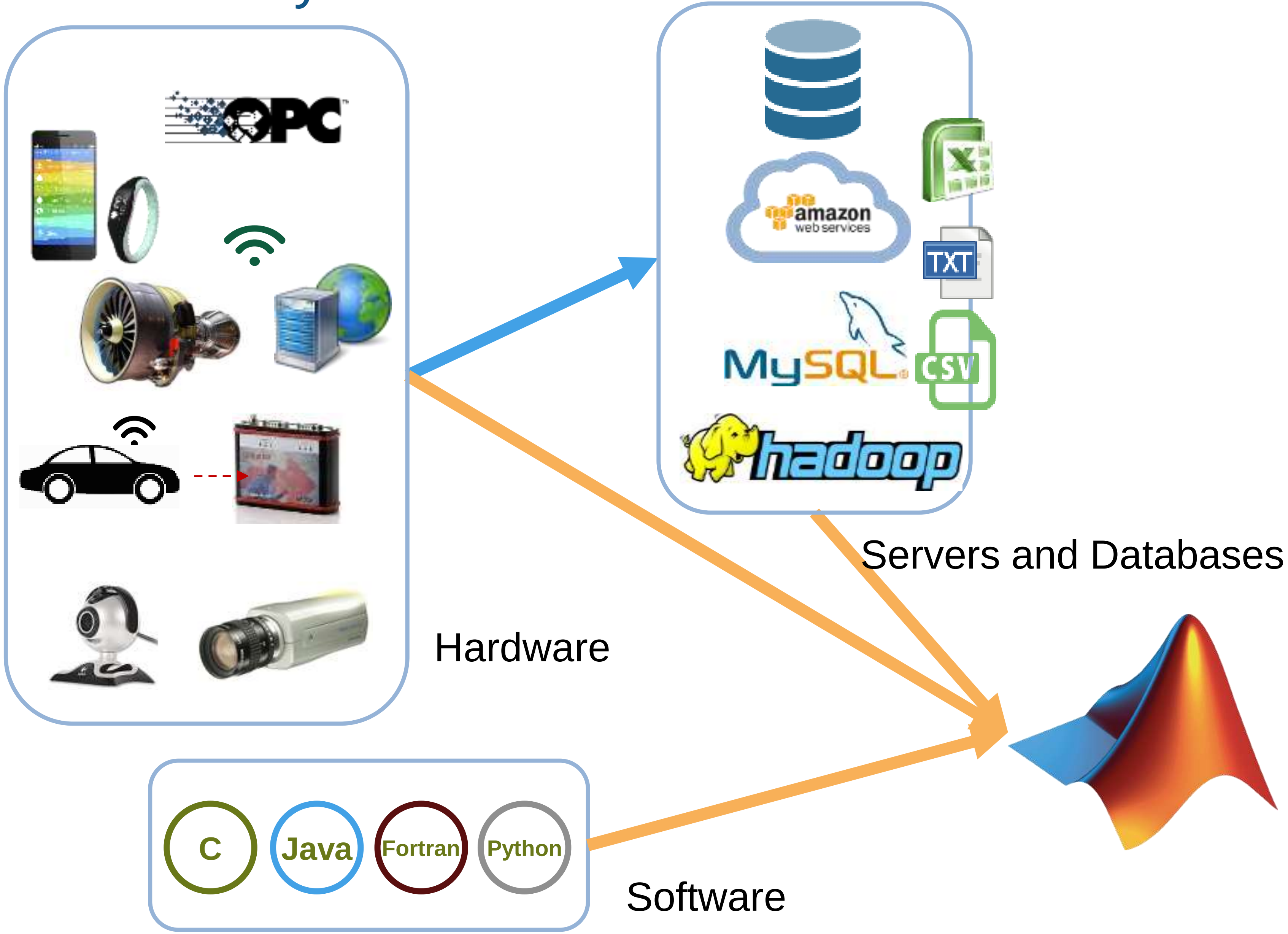
- Data aggregation
  - Different sources (files, web, etc.)
  - Different types (images, text, audio, etc.)
- Data clean up
  - Poorly formatted files
  - Irregularly sampled data
  - Redundant data, outliers, missing data etc.
- Data specific processing
  - Signals: Smoothing, resampling, denoising, Wavelet transforms, etc.
  - Images: Image registration, morphological filtering, deblurring, etc.
- Dealing with out of memory data (big data)



What data scientists spend the most time doing

Data preparation accounts for about 80% of the work of data scientists - Forbes

Data Analytics Workflow: Data Access



Servers and Databases

Hardware

Software

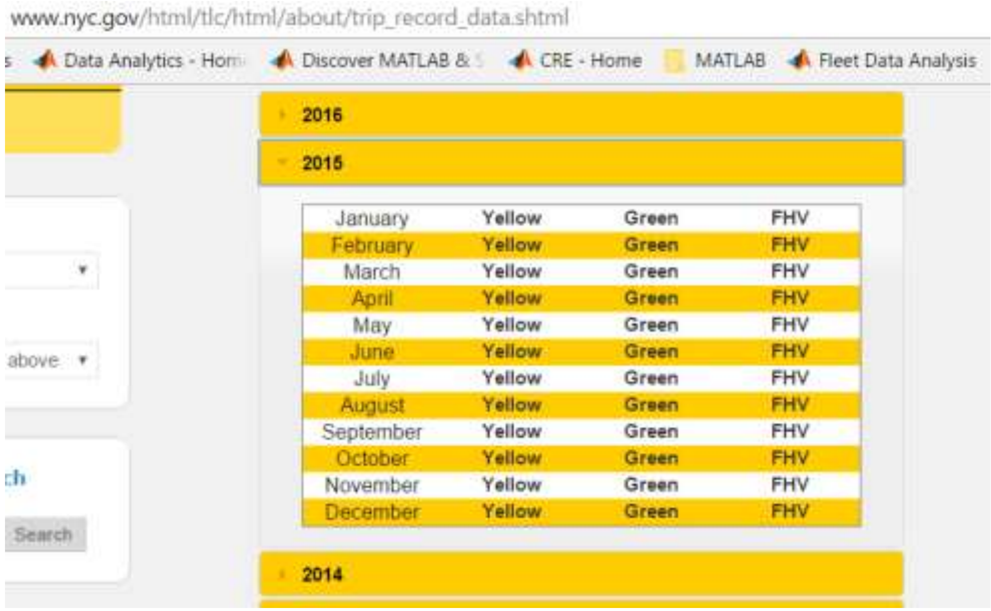
Business and Transactional Data

- Repositories – SQL, NoSQL, etc.
- File I/O – Text, Spreadsheet, etc.
- Web Sources – RESTful, JSON, etc.

Engineering, Scientific and Field Data

- Real-Time Sources – Sensors, GPS, etc.
- File I/O – Image, Audio, etc.
- Communication Protocols – OPC (OLE for Process Control), CAN (Controller Area Network), etc.

Data Analytics Workflow: Big Data Access and Pre-processing



www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

2016

2015

January	Yellow	Green	FHV
February	Yellow	Green	FHV
March	Yellow	Green	FHV
April	Yellow	Green	FHV
May	Yellow	Green	FHV
June	Yellow	Green	FHV
July	Yellow	Green	FHV
August	Yellow	Green	FHV
September	Yellow	Green	FHV
October	Yellow	Green	FHV
November	Yellow	Green	FHV
December	Yellow	Green	FHV

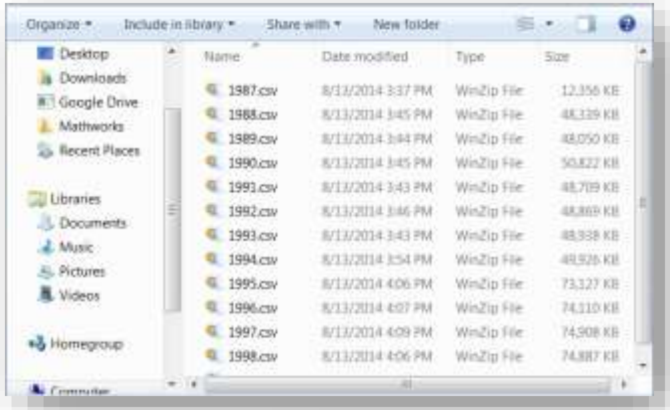
2014

Download 2015 Taxi Data from Web using *'websave' in parallel*

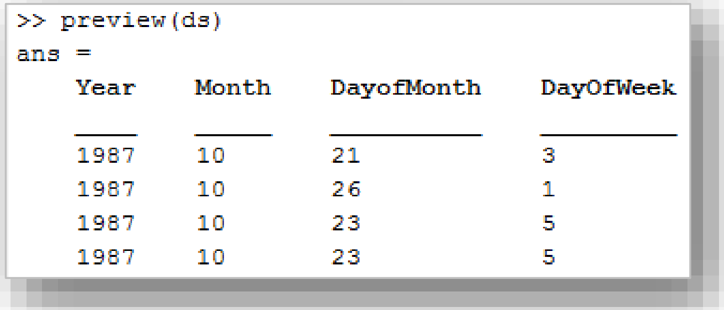
```
parfor i=1:12
   fileName = ['taxiData2015_', num2str(i)]
   url      = ['https://s3.amazonaws.com/nyc-tlc/trip+data/yellow_tripdata_2016-0',num2str(i), '.csv']
   websave(fileName, url)
end
```

Big Data in Recent Releases

- **`datastore`**
  - Tabular text files
  - Images
  - Excel spreadsheets
  - (SQL) Databases
  - HDFS (Hadoop)
  - S3 (Amazon Web Services)
- **MATLAB MapReduce**
  - Scales from Desktop to Hadoop



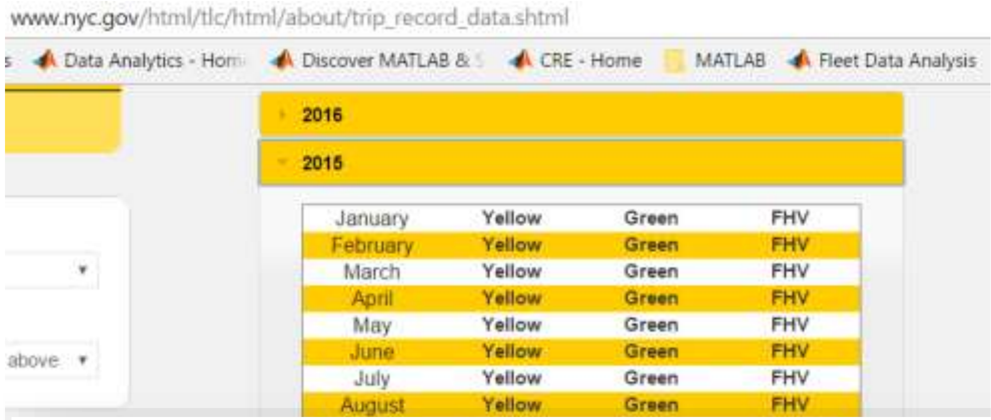
Name	Date modified	Type	Size
1987.csv	8/13/2014 3:37 PM	WinZip File	12,356 KB
1988.csv	8/13/2014 3:45 PM	WinZip File	48,339 KB
1989.csv	8/13/2014 3:44 PM	WinZip File	48,050 KB
1990.csv	8/13/2014 3:45 PM	WinZip File	50,822 KB
1991.csv	8/13/2014 3:43 PM	WinZip File	48,709 KB
1992.csv	8/13/2014 3:46 PM	WinZip File	48,869 KB
1993.csv	8/13/2014 3:43 PM	WinZip File	48,938 KB
1994.csv	8/13/2014 3:54 PM	WinZip File	49,926 KB
1995.csv	8/13/2014 4:06 PM	WinZip File	73,127 KB
1996.csv	8/13/2014 4:07 PM	WinZip File	74,110 KB
1997.csv	8/13/2014 4:09 PM	WinZip File	74,908 KB
1998.csv	8/13/2014 4:06 PM	WinZip File	74,887 KB



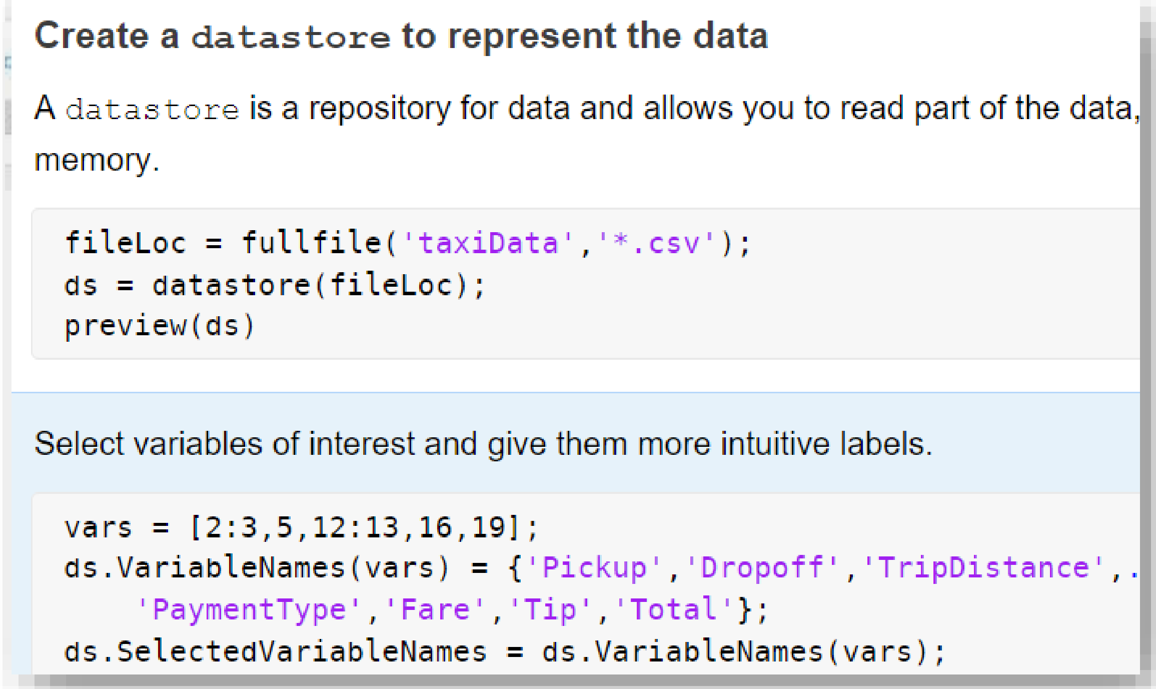
```
>> preview(ds)
ans =
      Year      Month     DayofMonth      DayOfWeek
      ____      _____     __________      _________
      1987      10        21              3
      1987      10        26              1
      1987      10        23              5
      1987      10        23              5
```

```
airdata = datastore('*.csv');
airdata.SelectedVariables = {'Distance', 'ArrDelay'};

data = read(airdata);
```

Data Analytics Workflow: Big Data Access and Pre-processing



January	Yellow	Green	FHV
February	Yellow	Green	FHV
March	Yellow	Green	FHV
April	Yellow	Green	FHV
May	Yellow	Green	FHV
June	Yellow	Green	FHV
July	Yellow	Green	FHV
August	Yellow	Green	FHV



Create a `datastore` to represent the data

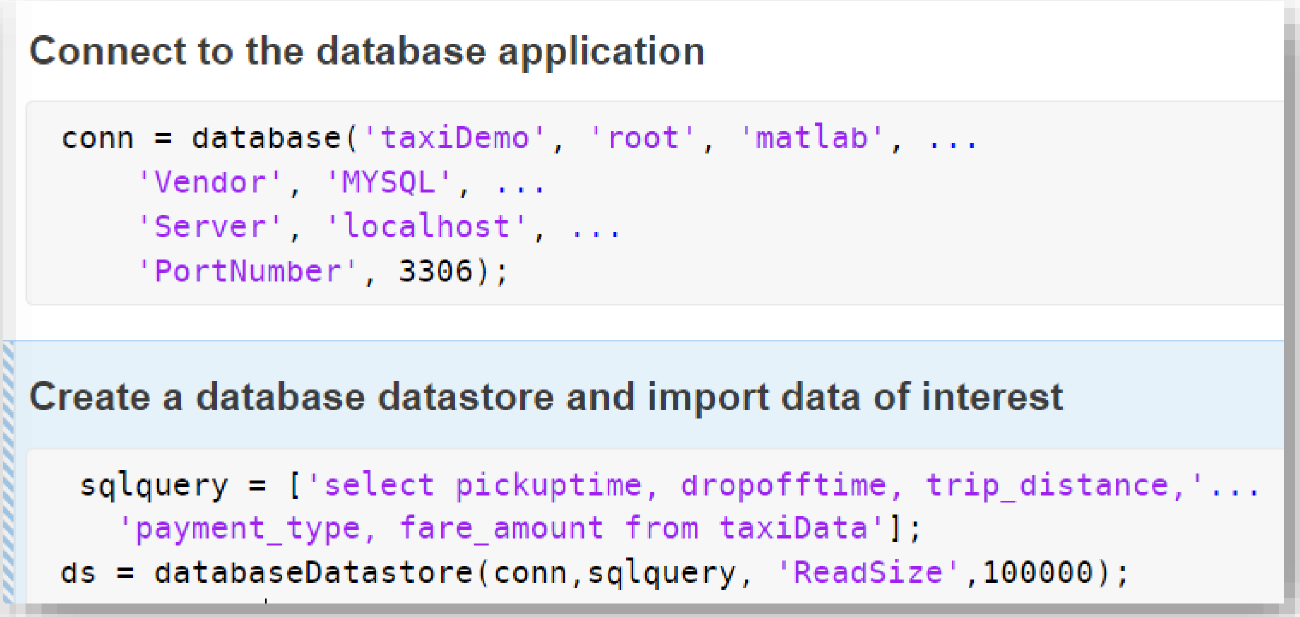
A `datastore` is a repository for data and allows you to read part of the data,

memory.

```
fileLoc = fullfile('taxiData','*.csv');
ds = datastore(fileLoc);
preview(ds)
```

Select variables of interest and give them more intuitive labels.

```
vars = [2:3,5,12:13,16,19];
ds.VariableNames(vars) = {'Pickup','Dropoff','TripDistance',.
    'PaymentType','Fare','Tip','Total'};
ds.SelectedVariableNames = ds.VariableNames(vars);
```



Connect to the database application

```
conn = database('taxiDemo', 'root', 'matlab', ...
     'Vendor', 'MYSQL', ...
     'Server', 'localhost', ...
     'PortNumber', 3306);
```

Create a database datastore and import data of interest

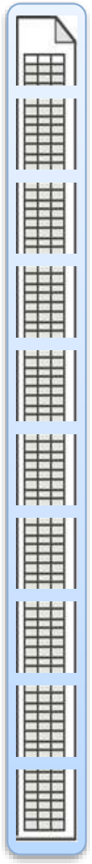
```
 sqlquery = ['select pickuptime, dropofftime, trip_distance,'...
    'payment_type, fare_amount from taxiData'];
ds = databaseDatastore(conn,sqlquery, 'ReadSize',100000);
```

`tall` arrays in R2016b

- New data type designed for data that doesn't fit into memory

- Lots of observations (hence "tall")

- Looks like a normal MATLAB array
  - Supports numeric types, tables, datetimes, strings, etc…
  - Supports several hundred functions for basic math, stats, indexing, etc.
  - **Statistics and Machine Learning Toolbox** support (clustering, classification, etc.)

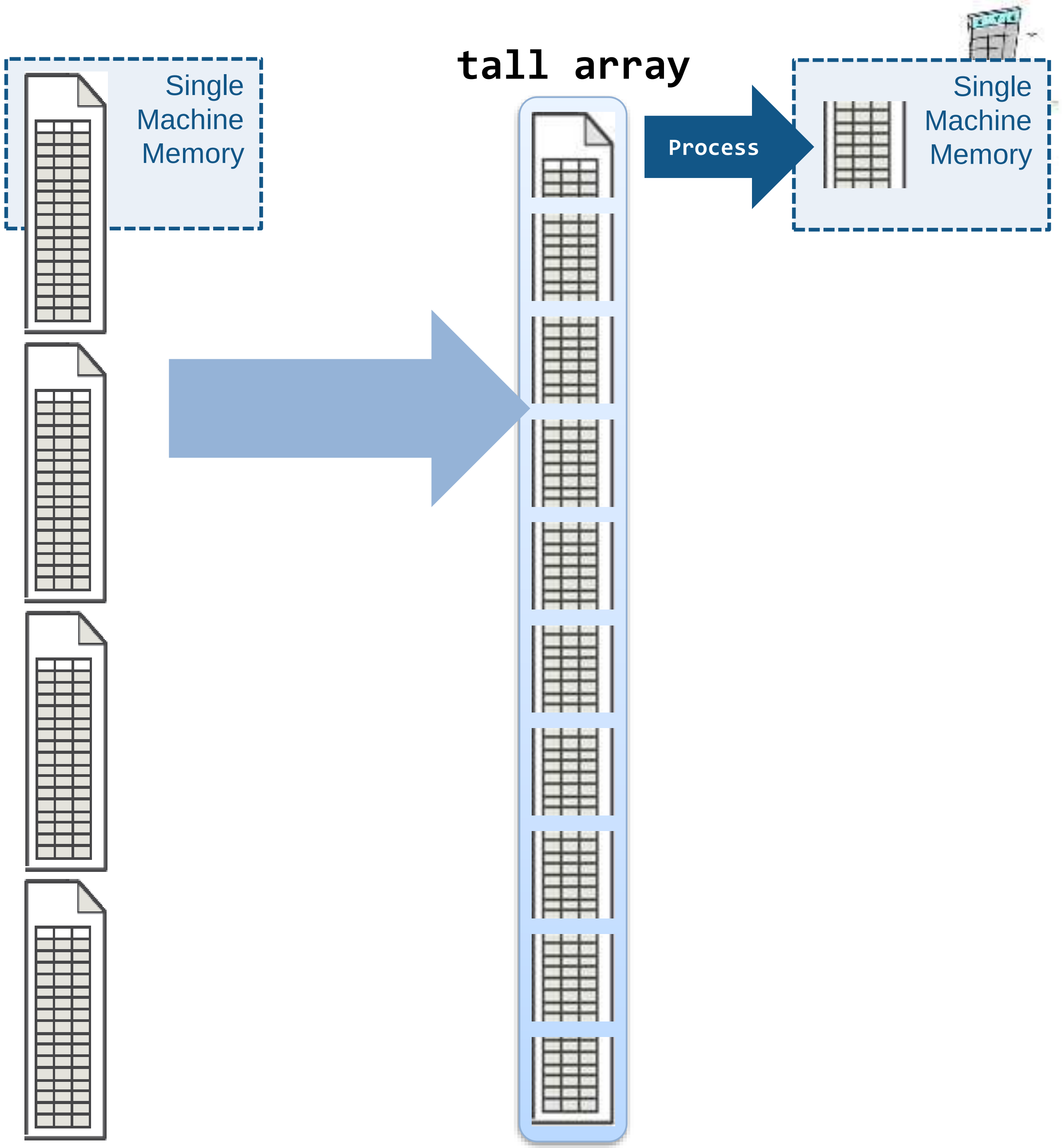


`tall` arrays R2016b

- Automatically breaks data up into s mall “chunks” that fit in memory

- Tall arrays scan through the datase t one “chunk” at a time

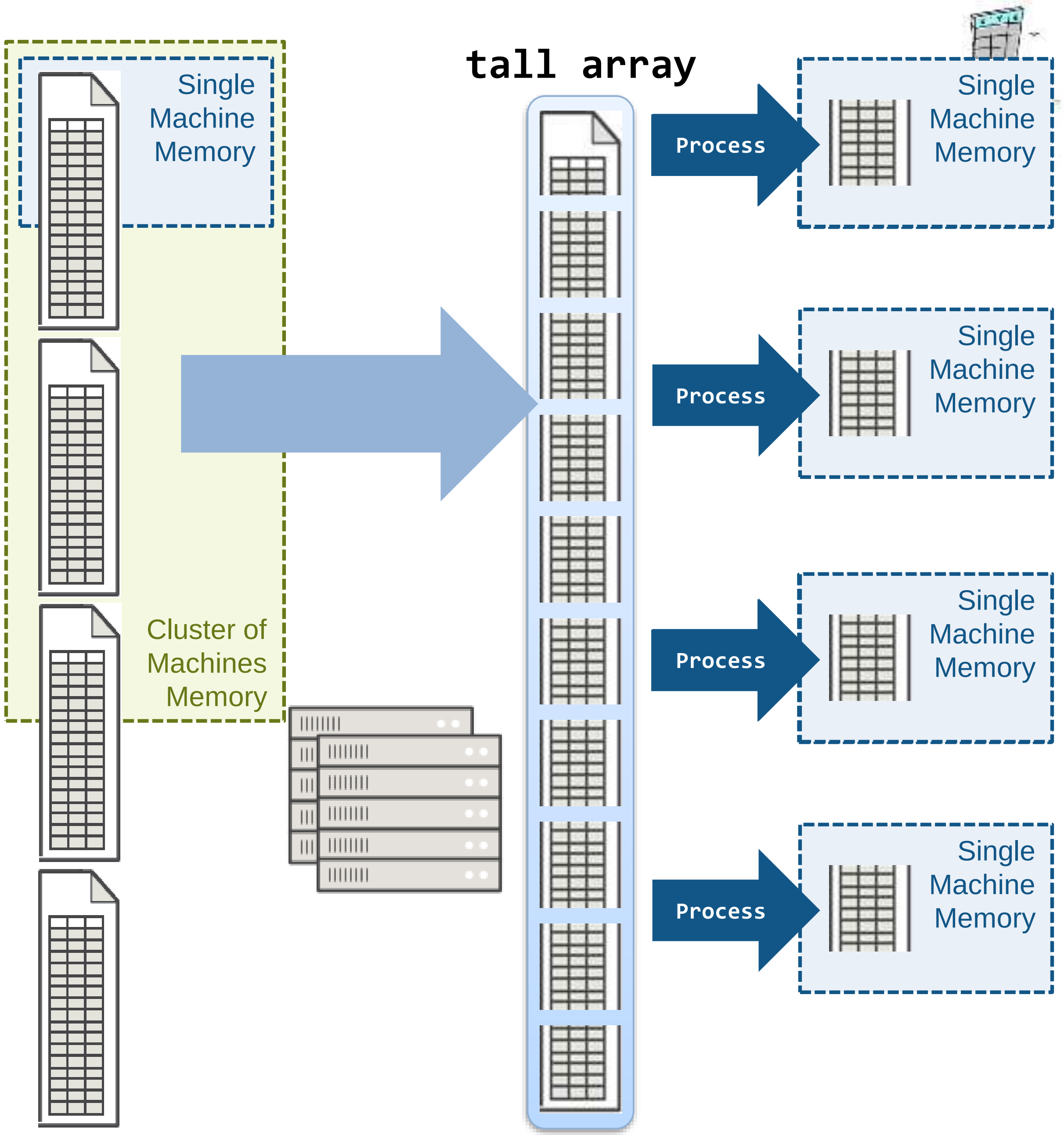
- Processing code for tall arrays is th e same as ordinary arrays



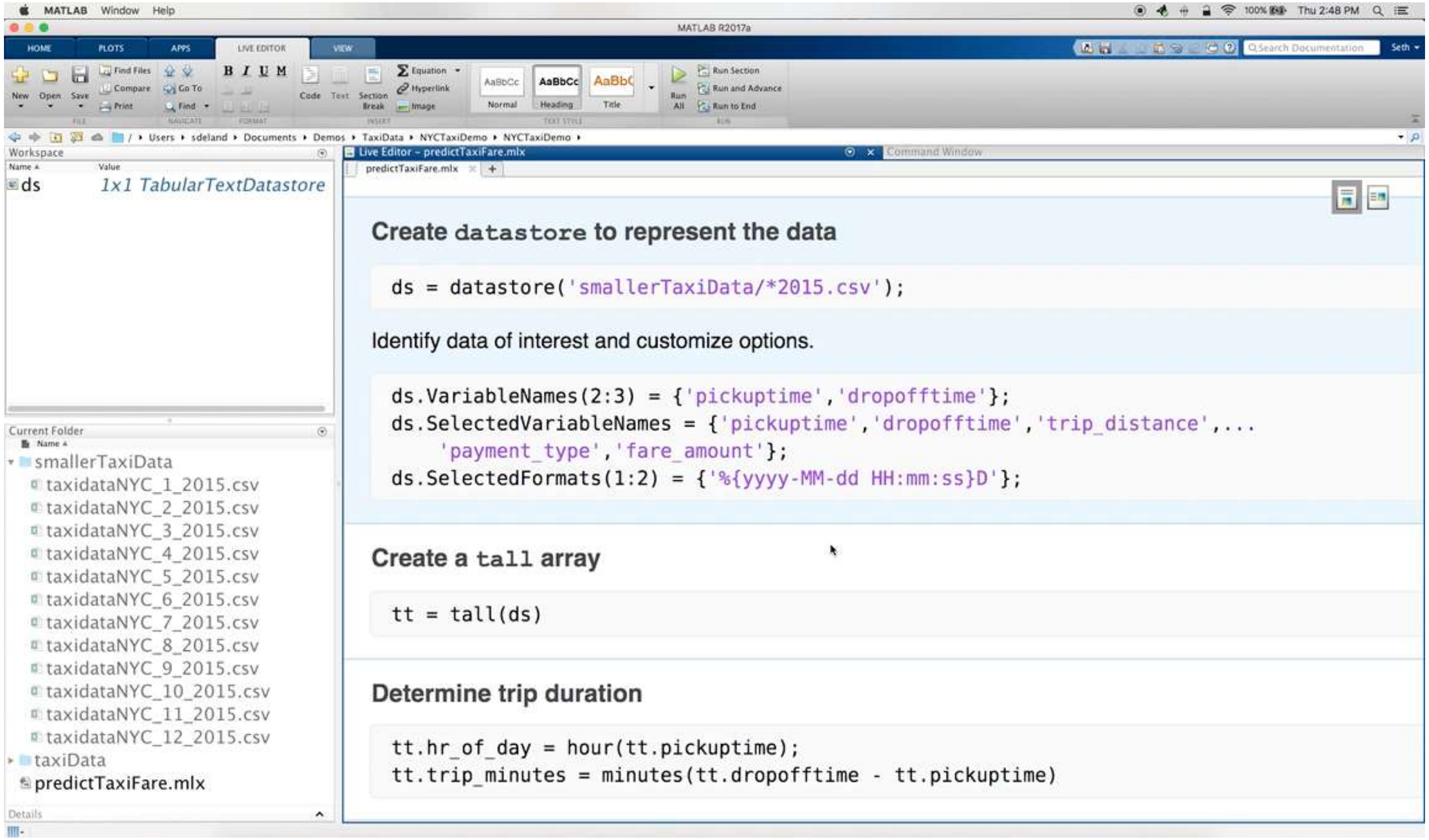
`tall` arrays R2016b

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

- Can scale up to clusters with MATL AB Distributed Computing Server



Demo: Working with Tall Arrays


MATLAB Window Help
Thu 2:48 PM
MATLAB R2017a
HOME PLOTS APPS LIVE EDITOR VIEW
Search Documentation
Seth
New Open Save Find Files Compare Print Go To Find
Code Text Section Break Equation Hyperlink Image
Normal Heading Title
Run All Run Section Run and Advance Run to End
/ Users sdeland Documents Demos TaxiData NYCTaxiDemo NYCTaxiDemo
Workspace
Name Value
Live Editor – predictTaxiFare.mlx
Command Window
predictTaxiFare.mlx
Current Folder
Name
Details

Name	Value
ds	1x1 TabularTextDatastore

- smallerTaxiData
  - taxidataNYC_1_2015.csv
  - taxidataNYC_2_2015.csv
  - taxidataNYC_3_2015.csv
  - taxidataNYC_4_2015.csv
  - taxidataNYC_5_2015.csv
  - taxidataNYC_6_2015.csv
  - taxidataNYC_7_2015.csv
  - taxidataNYC_8_2015.csv
  - taxidataNYC_9_2015.csv
  - taxidataNYC_10_2015.csv
  - taxidataNYC_11_2015.csv
  - taxidataNYC_12_2015.csv
- taxiData
- predictTaxiFare.mlx

Create `datastore` to represent the data

```
ds = datastore('smallerTaxiData/*2015.csv');
```

Identify data of interest and customize options.

```
ds.VariableNames(2:3) = {'pickuptime','dropofftime'};
ds.SelectedVariableNames = {'pickuptime','dropofftime','trip_distance',...
    'payment_type','fare_amount'};
ds.SelectedFormats(1:2) = {'%{yyyy-MM-dd HH:mm:ss}D'};
```

Create a `tall` array

```
tt = tall(ds)
```

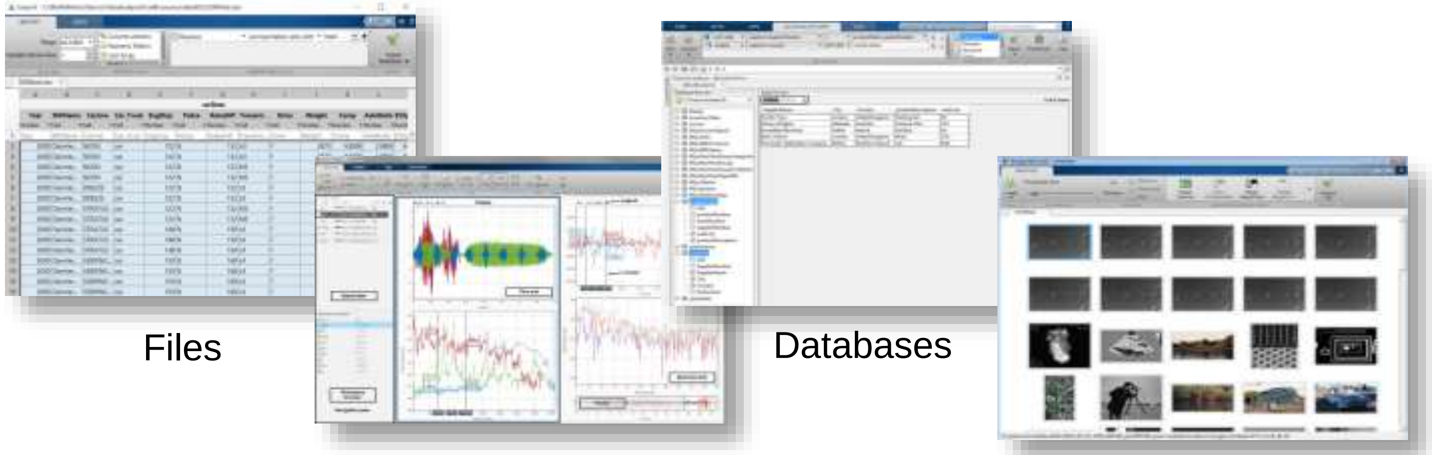
Determine trip duration

```
tt.hr_of_day = hour(tt.pickuptime);
tt.trip_minutes = minutes(tt.dropofftime - tt.pickuptime)
```

Data Access and pre-processing – challenges and solution

Challenges

- Data aggregation
  - Different sources (files, web, etc.)
  - Different types (images, text, audio, etc.)
- Data clean up
  - Poorly formatted files
  - Irregularly sampled data
  - Redundant data, outliers, missing data etc.
- Data specific processing
  - Signals: Smoothing, resampling, denoising, Wavelet transforms, etc.
  - Images: Image registration, morphological filtering, deblurring, etc.
- Dealing with out of memory data (big data)



Files

Signals

Databases

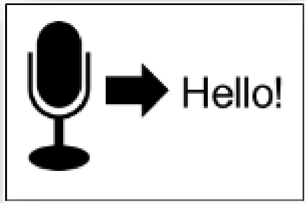
Images

- Point and click tools to access variety of data sources
- High-performance environment for **big data**

- Built-in algorithms for data preprocessing including sensor, image, audio, video and other real-time data

Consider Machine/Deep Learning When

Problem is too complex for hand written rules or equations



Speech Recognition

Object Recognition

Engine Health Monitoring

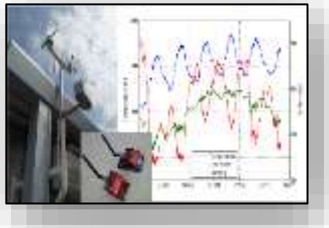
Program needs to adapt with changing data

Weather Forecasting

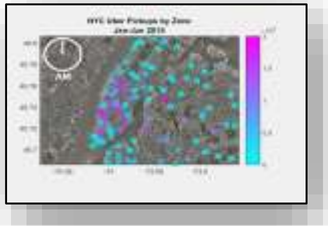
Energy Load Forecasting

Stock Market Prediction

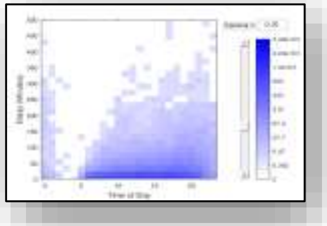
Program needs to scale



IoT Analytics



Taxi Availability



Airline Flight Delays

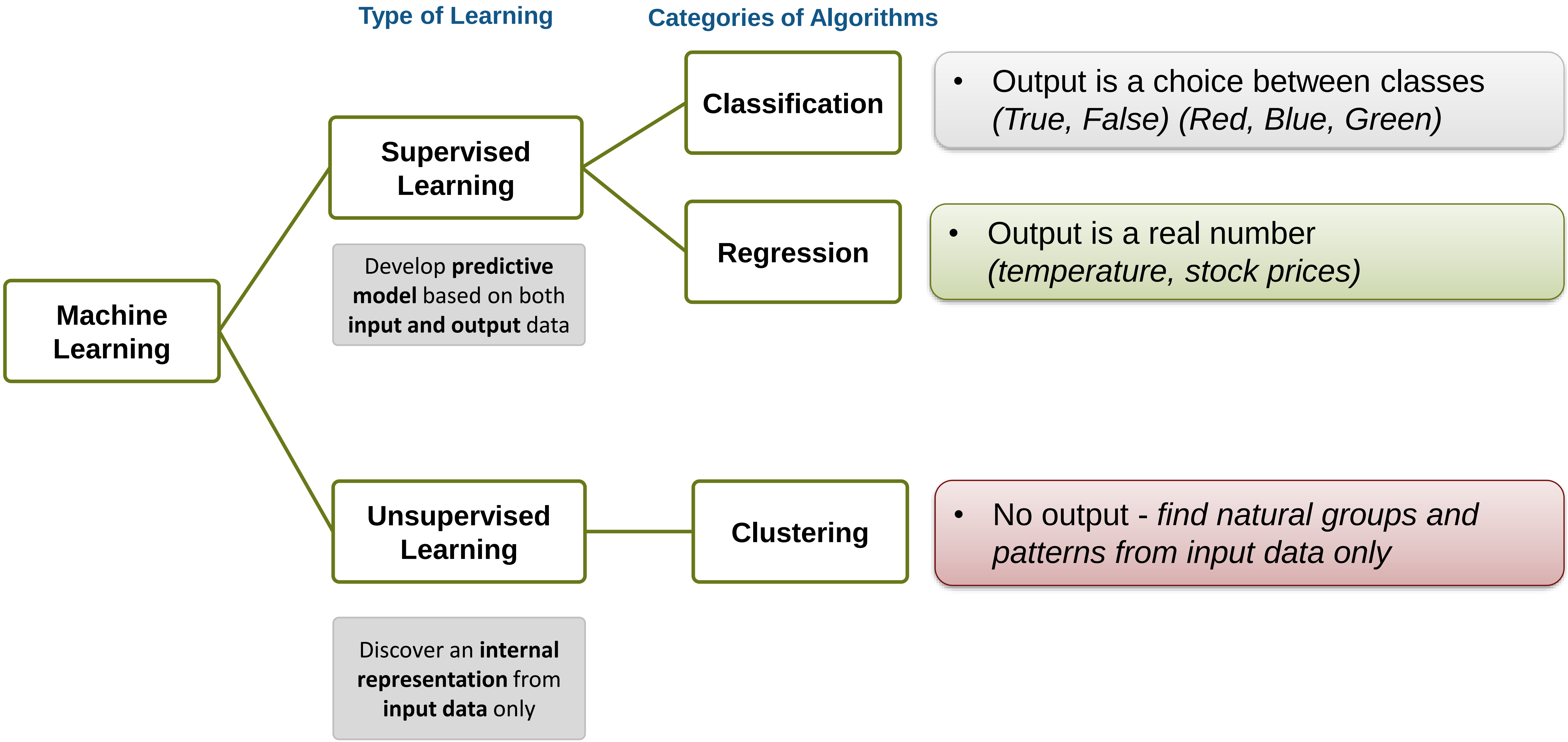
Because algorithms can

learn complex non-linear relationships

update as more data becomes available

learn efficiently from very large data sets

Different Types of Learning



Type of Learning

Categories of Algorithms

Machine Learning

Supervised Learning

Develop **predictive model** based on both **input and output** data

Classification

- Output is a choice between classes *(True, False) (Red, Blue, Green)*

Regression

- Output is a real number *(temperature, stock prices)*

Unsupervised Learning

Discover an **internal representation** from **input data** only

Clustering

- No output - *find natural groups and patterns from input data only*

Machine Learning with Big Data

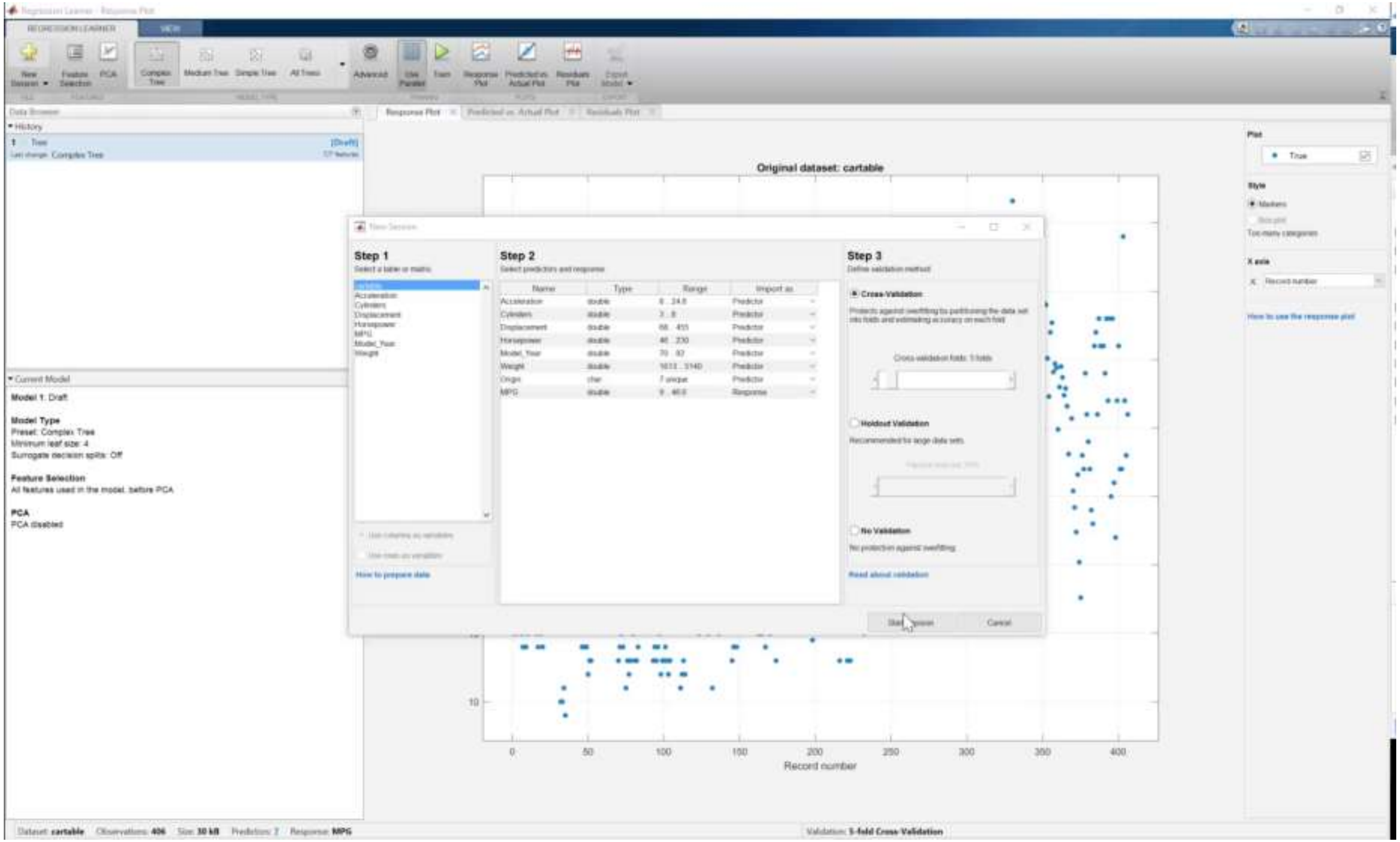
R2016b

- Descriptive statistics (skewness, tabulate, crosstab, cov, grpstats, …)
- K-means clustering (kmeans)
- Visualization (ksdensity, binScatterPlot; histogram, histogram2)
- Dimensionality reduction (pca, pcacov, factoran)
- Linear and generalized linear regression (fitlm, fitglm)
- Discriminant analysis (fitcdiscr)

R2017a

- Linear classification methods for SVM and logistic regression (fitclinear)
- Random forest ensembles of classification trees (TreeBagger)
- Naïve Bayes classification (fitcnb)
- Regularized regression (lasso)
- Prediction applied to tall arrays

Regression Learner



History

1 Tree [Draft]
Last change: Complex Tree

Current Model

Model 1: Draft

Model Type
Preset: Complex Tree
Minimum leaf size: 4
Surrogate decision splits: Off

Feature Selection
All features used in the model, before PCA

PCA
PCA disabled

Step 1
Select a table or matrix

cartable
Acceleration
Cylinders
Displacement
Horsepower
MPG
Model_Year
Weight

How to prepare data

Step 2
Select predictors and response

Name	Type	Range	Import as
Acceleration	double	8 .. 24.8	Predictor
Cylinders	double	3 .. 8	Predictor
Displacement	double	68 .. 455	Predictor
Horsepower	double	46 .. 230	Predictor
Model_Year	double	70 .. 82	Predictor
Weight	double	1613 .. 5140	Predictor
Origin	char	7 unique	Predictor
MPG	double	9 .. 46.6	Response

Step 3
Define validation method

● **Cross-Validation**
Protects against overfitting by partitioning the data set into folds and estimating accuracy on each fold

Cross-validation folds: 5 folds

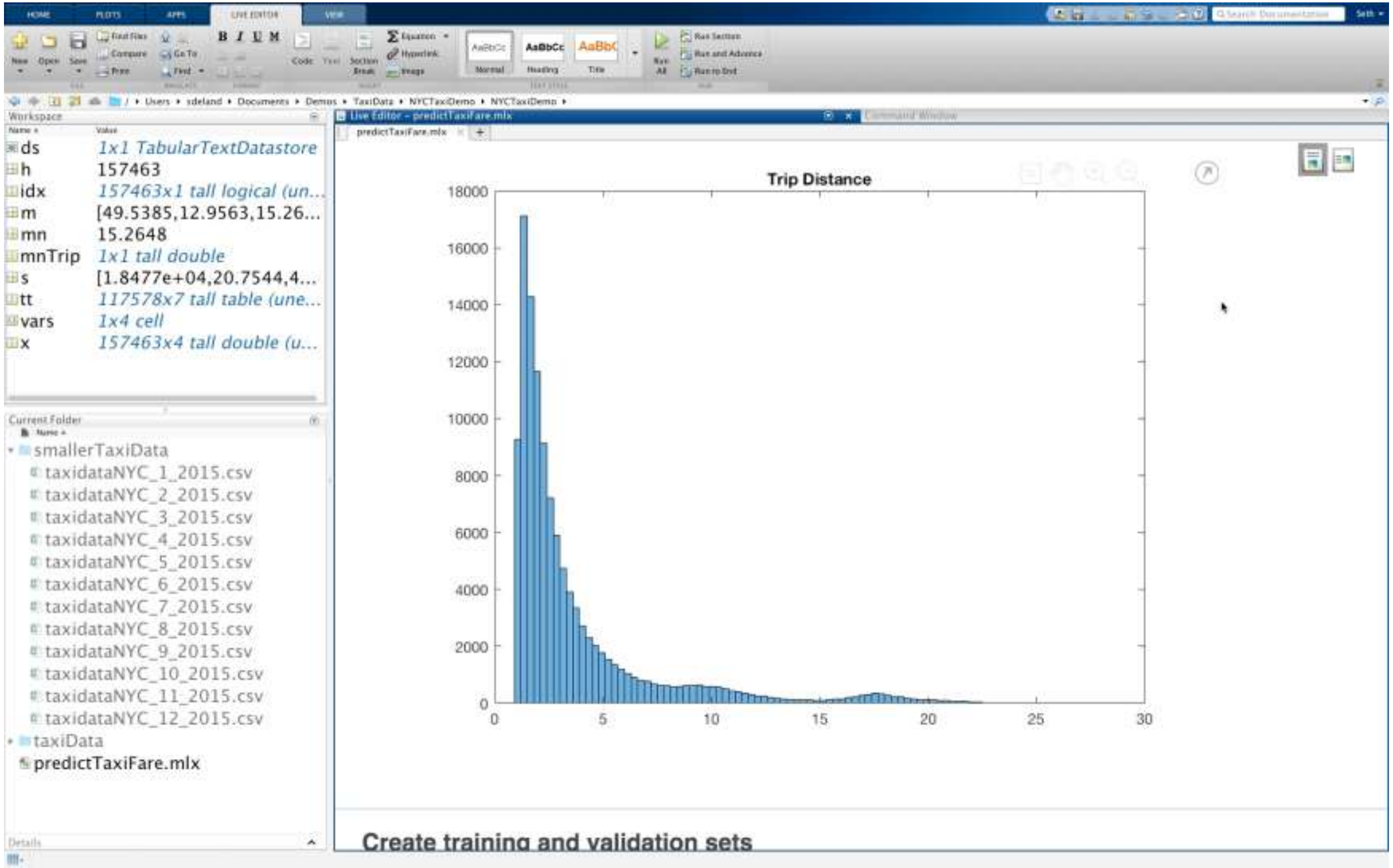
○ **Holdout Validation**
Recommended for large data sets

○ **No Validation**
No protection against overfitting

Read about validation

Dataset: **cartable** Observations: **406** Size: **30 kB** Predictors: 7 Response: **MPG** Validation: **5-fold Cross-Validation**

Demo: Training a Machine Learning Model

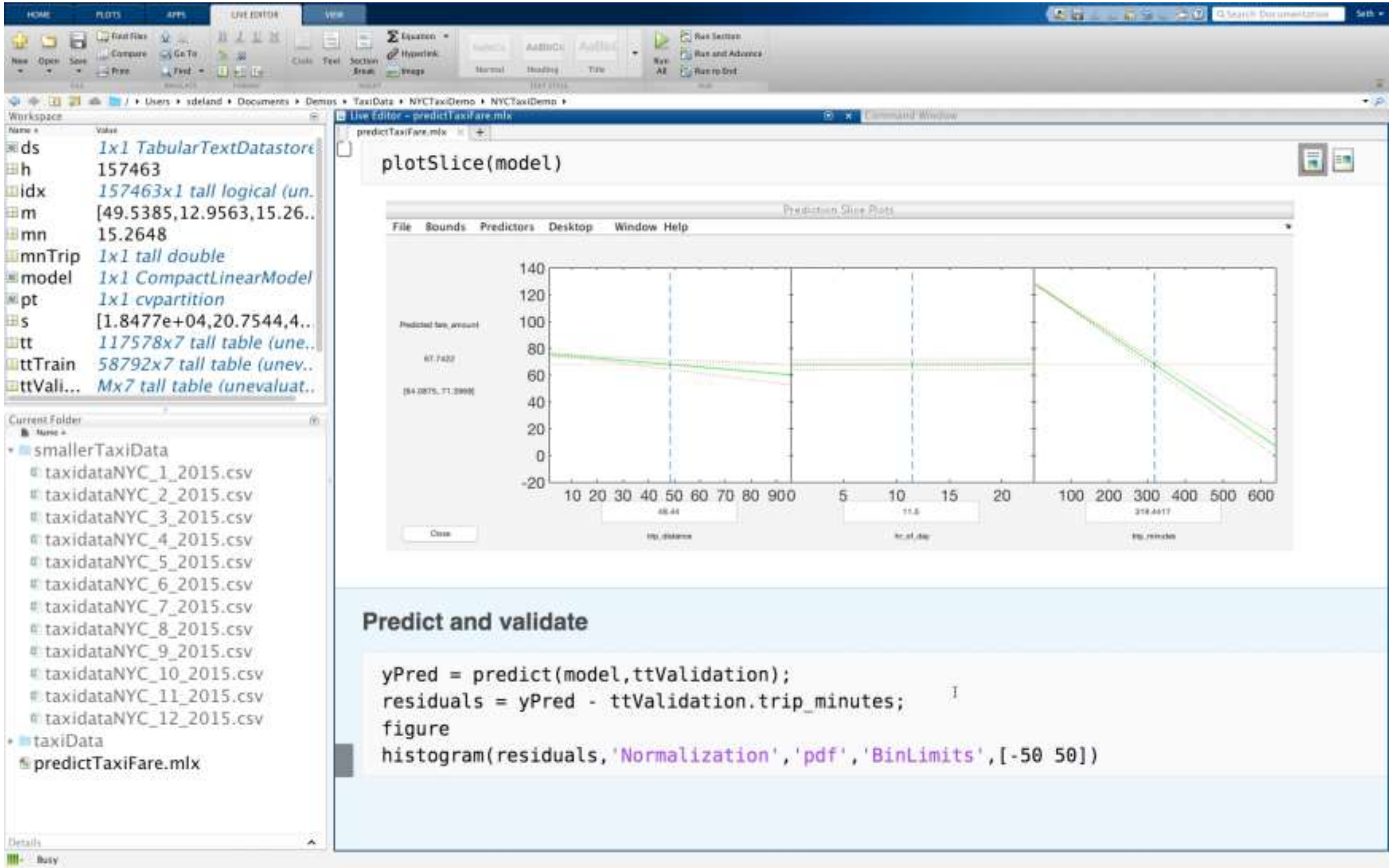


Name	Value
ds	1x1 TabularTextDatastore
h	157463
idx	157463x1 tall logical (un...
m	[49.5385,12.9563,15.26...
mn	15.2648
mnTrip	1x1 tall double
s	[1.8477e+04,20.7544,4...
tt	117578x7 tall table (une...
vars	1x4 cell
x	157463x4 tall double (u...

- smallerTaxiData
  - taxidataNYC_1_2015.csv
  - taxidataNYC_2_2015.csv
  - taxidataNYC_3_2015.csv
  - taxidataNYC_4_2015.csv
  - taxidataNYC_5_2015.csv
  - taxidataNYC_6_2015.csv
  - taxidataNYC_7_2015.csv
  - taxidataNYC_8_2015.csv
  - taxidataNYC_9_2015.csv
  - taxidataNYC_10_2015.csv
  - taxidataNYC_11_2015.csv
  - taxidataNYC_12_2015.csv
- taxiData
- predictTaxiFare.mlx

Create training and validation sets

Demo: Training a Machine Learning Model



Workspace

Name ▲	Value
ds	1x1 TabularTextDatastore
h	157463
idx	157463x1 tall logical (un..
m	[49.5385,12.9563,15.26..
mn	15.2648
mnTrip	1x1 tall double
model	1x1 CompactLinearModel
pt	1x1 cvpartition
s	[1.8477e+04,20.7544,4..
tt	117578x7 tall table (une..
ttTrain	58792x7 tall table (unev..
ttVali...	Mx7 tall table (unevaluat..

Current Folder

Name ▲

- smallerTaxiData
  - taxidataNYC_1_2015.csv
  - taxidataNYC_2_2015.csv
  - taxidataNYC_3_2015.csv
  - taxidataNYC_4_2015.csv
  - taxidataNYC_5_2015.csv
  - taxidataNYC_6_2015.csv
  - taxidataNYC_7_2015.csv
  - taxidataNYC_8_2015.csv
  - taxidataNYC_9_2015.csv
  - taxidataNYC_10_2015.csv
  - taxidataNYC_11_2015.csv
  - taxidataNYC_12_2015.csv
- taxiData
- predictTaxiFare.mlx

```
plotSlice(model)
```

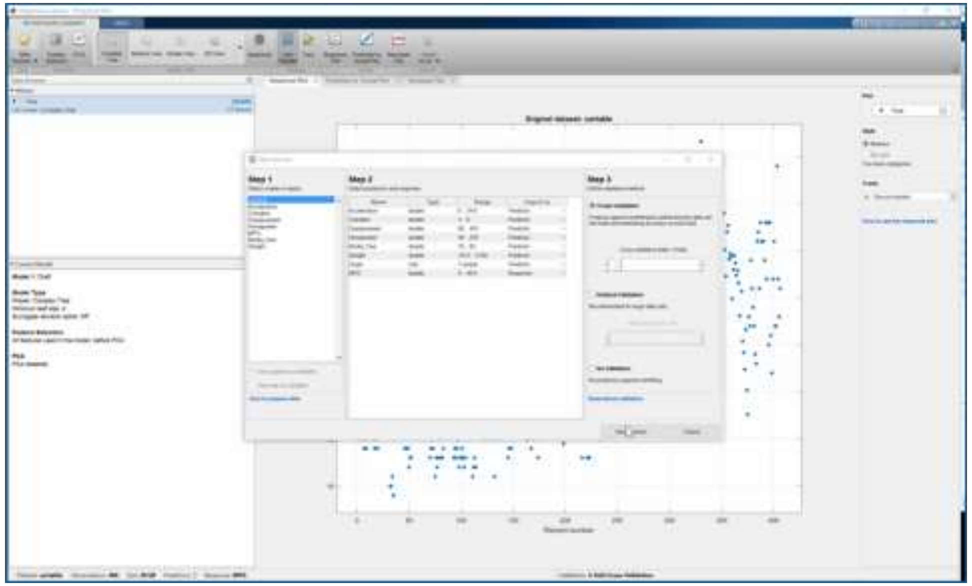
Predict and validate

```
yPred = predict(model,ttValidation);
residuals = yPred - ttValidation.trip_minutes;
figure
histogram(residuals,'Normalization','pdf','BinLimits',[-50 50])
```

Regression Learner

App to apply advanced regression methods to your data

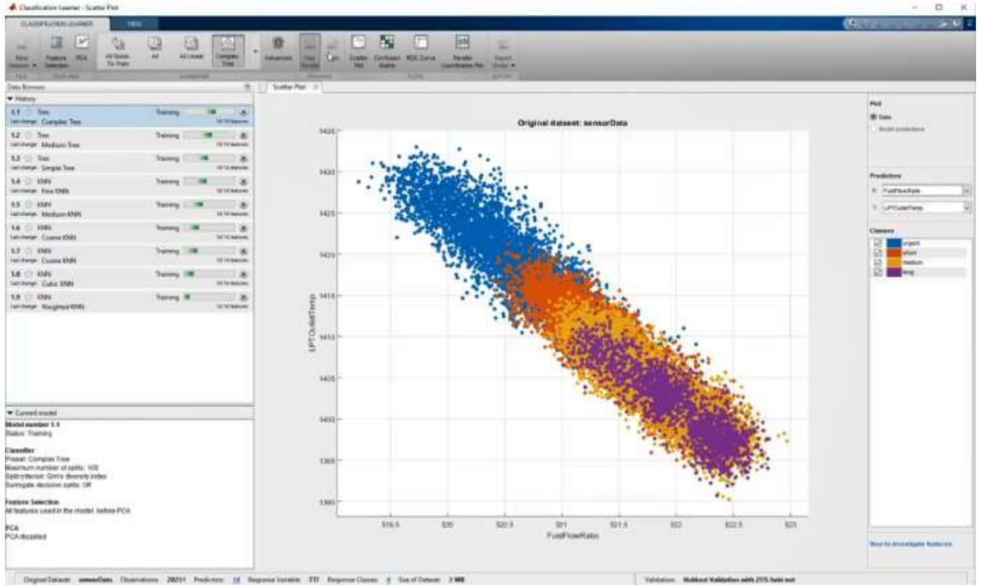
- Added to Statistics and Machine Learning Toolbox in R2017a
- Point and click interface – no coding requi red
- Quickly evaluate, compare and select regr ession models
- Export and share MATLAB code or traine d models



Classification Learner

App to apply advanced classification methods to your data

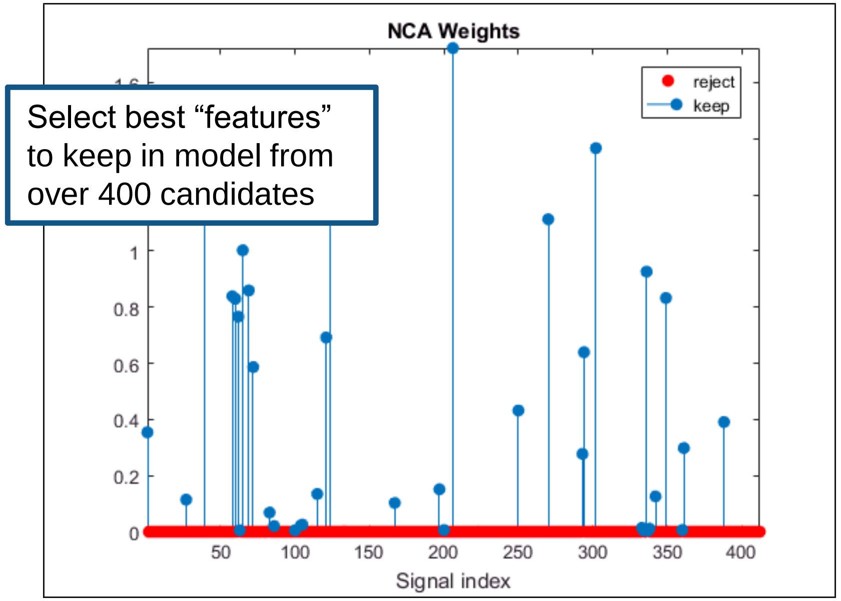
- Added to Statistics and Machine Learning Toolbox in R2014a
- Point and click interface – no coding required
- Quickly evaluate, compare and select classification models
- Export and share MATLAB code or trained models



Tuning Machine Learning Models

Get more accurate models in less time

Automatically select best machine leaning “features”



Select best “features” to keep in model from over 400 candidates

R2016b

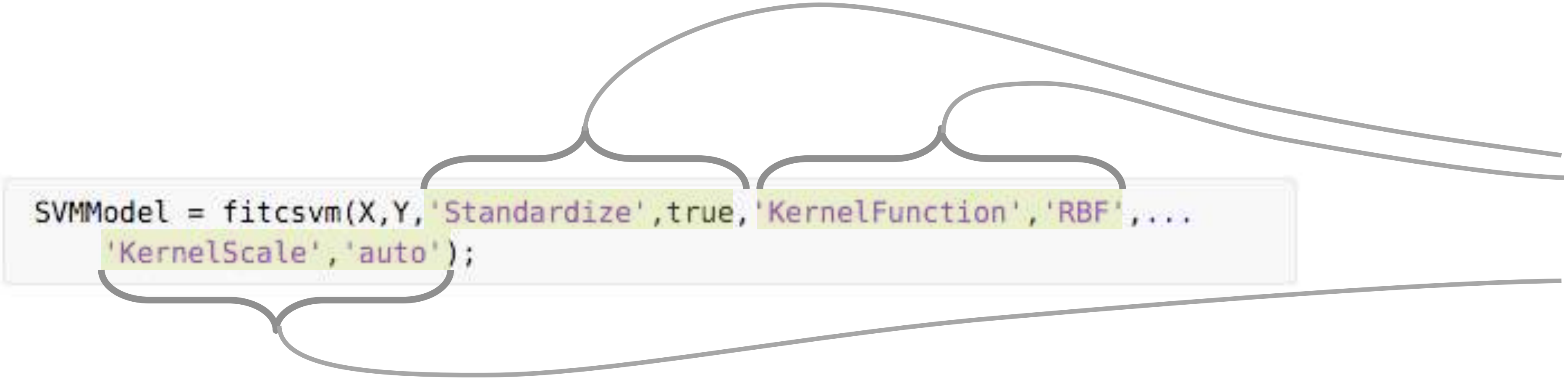
NCA: Neighborhood Component Analysis

Automatically fine-tune machine learning parameters

R2016b

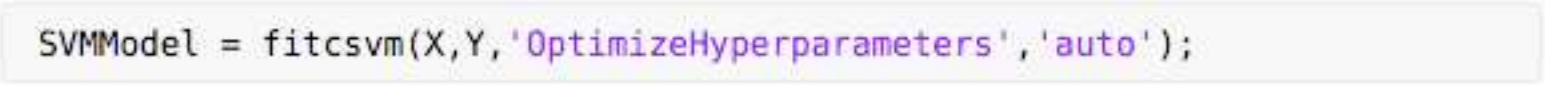
Hyperparameter Tuning

Machine Learning Hyperparameters



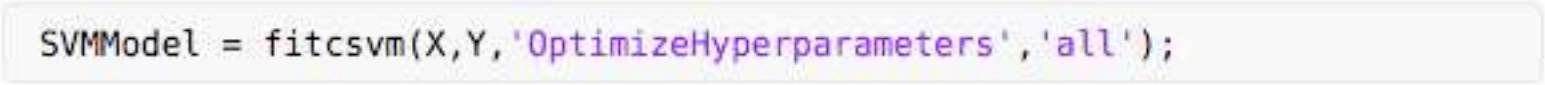
```
SVMModel = fitcsvm(X,Y,'Standardize',true,'KernelFunction','RBF',...
    'KernelScale','auto');
```

Hyperparameters



```
SVMModel = fitcsvm(X,Y,'OptimizeHyperparameters','auto');
```

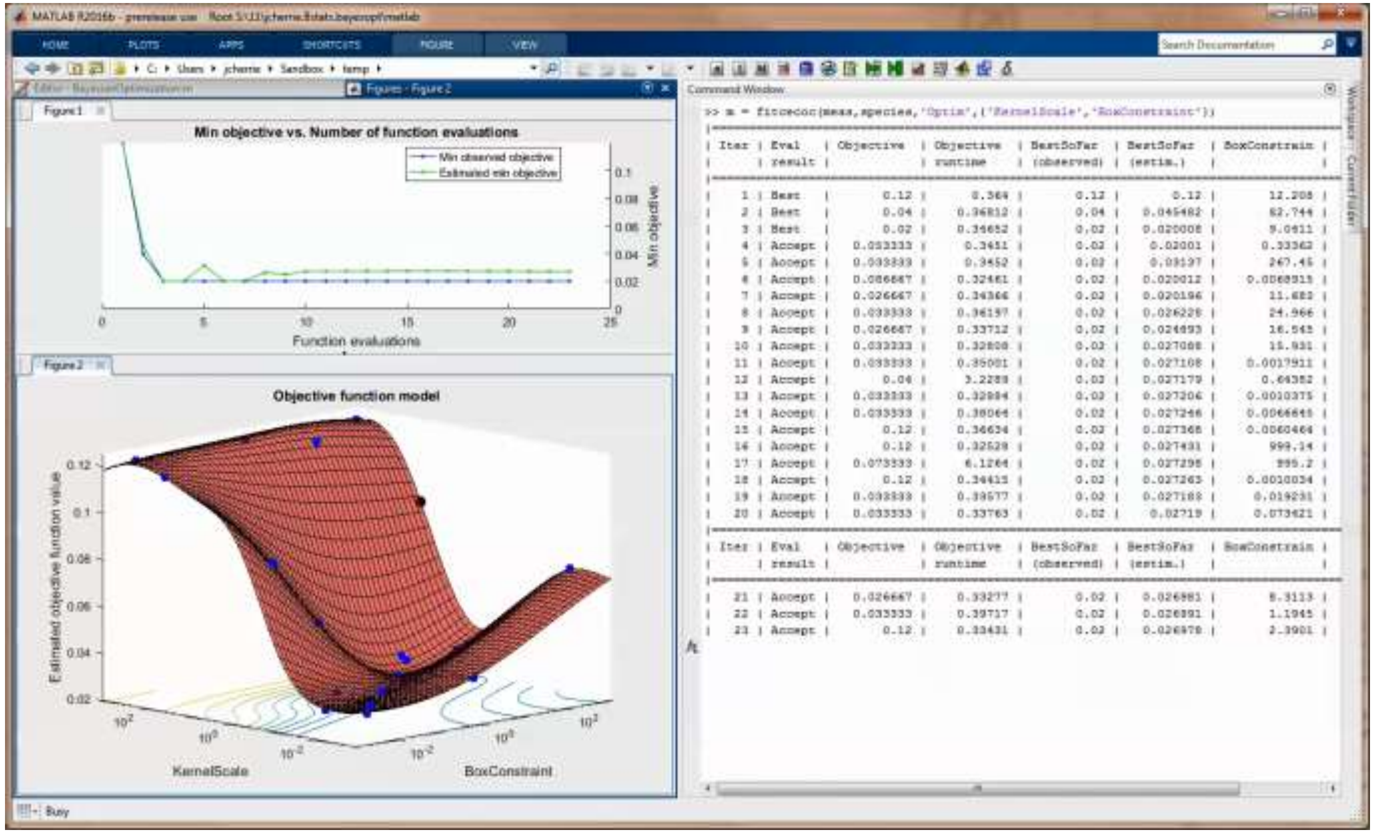
Tune a typical set of
hyperparameters for this model



```
SVMModel = fitcsvm(X,Y,'OptimizeHyperparameters','all');
```

Tune all
hyperparameters for this model

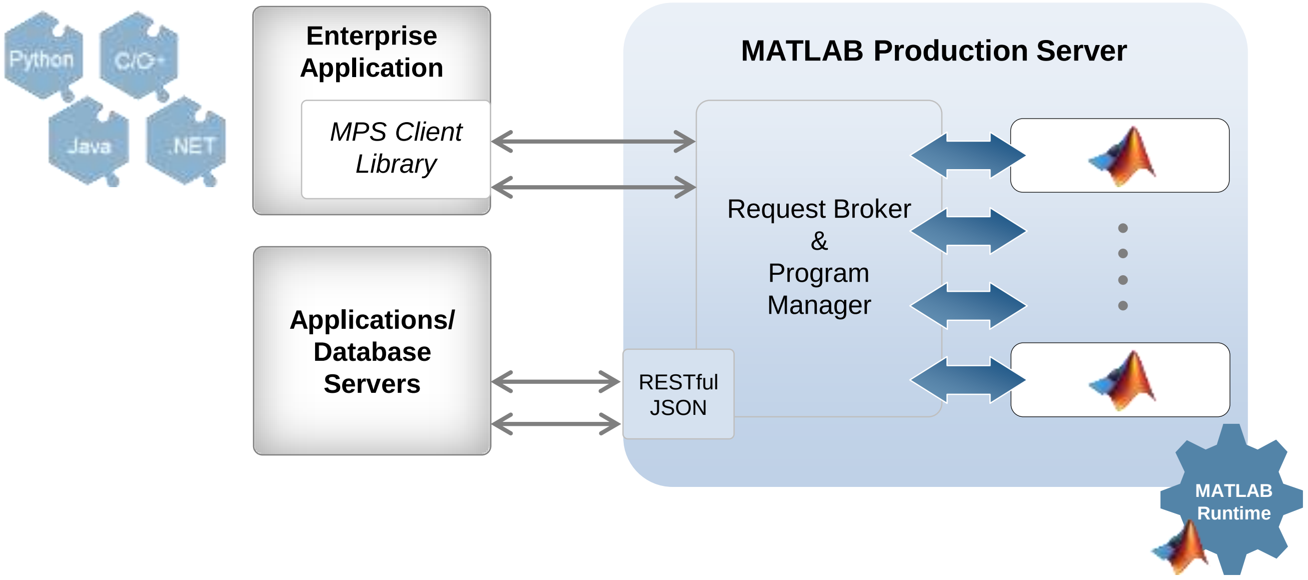
Bayesian Optimization in Action



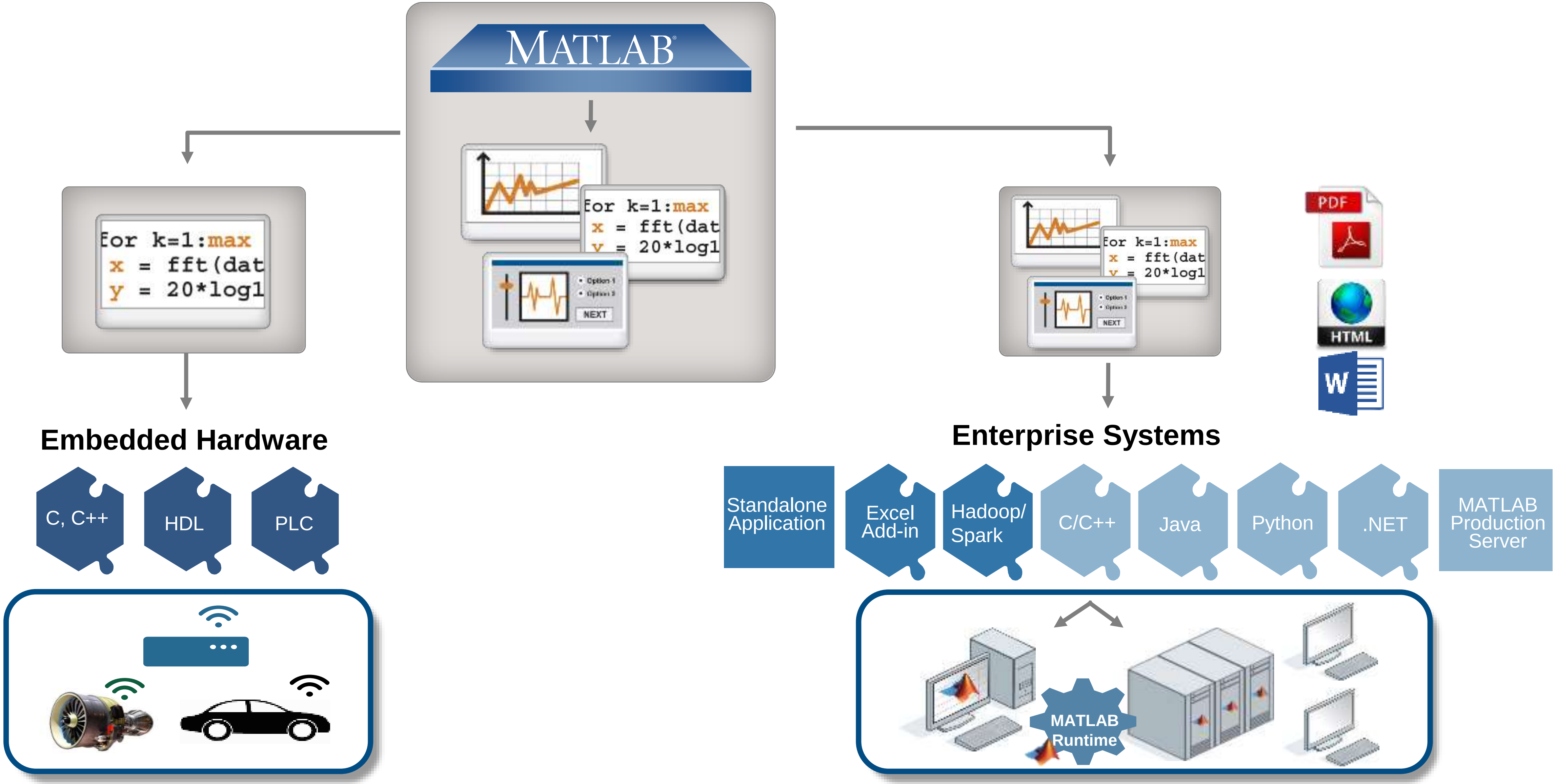
```
>> m = fitcecoc(meas,species,'Optim',{'KernelScale','BoxConstraint'})
|=====================================================================================================|
| Iter | Eval   | Objective  | Objective  | BestSoFar  | BestSoFar  | BoxConstrain |
|      | result |            | runtime    | (observed) | (estim.)   |              |
|=====================================================================================================|
|    1 | Best   |       0.12 |      0.364 |       0.12 |       0.12 |       12.208 |
|    2 | Best   |       0.04 |    0.36812 |       0.04 |   0.045482 |       62.744 |
|    3 | Best   |       0.02 |    0.34652 |       0.02 |   0.020008 |       9.0611 |
|    4 | Accept |   0.033333 |     0.3451 |       0.02 |    0.02001 |      0.33362 |
|    5 | Accept |   0.033333 |     0.3452 |       0.02 |    0.03137 |       267.45 |
|    6 | Accept |   0.066667 |    0.32461 |       0.02 |   0.020012 |    0.0068515 |
|    7 | Accept |   0.026667 |    0.34366 |       0.02 |   0.020196 |       11.682 |
|    8 | Accept |   0.033333 |    0.36157 |       0.02 |   0.026228 |       24.966 |
|    9 | Accept |   0.026667 |    0.33712 |       0.02 |   0.024693 |       16.545 |
|   10 | Accept |   0.033333 |    0.32808 |       0.02 |   0.027088 |       15.931 |
|   11 | Accept |   0.033333 |    0.35001 |       0.02 |   0.027108 |    0.0017911 |
|   12 | Accept |       0.04 |     3.2289 |       0.02 |   0.027179 |      0.64352 |
|   13 | Accept |   0.033333 |    0.32984 |       0.02 |   0.027206 |    0.0010375 |
|   14 | Accept |   0.033333 |    0.38064 |       0.02 |   0.027246 |    0.0066645 |
|   15 | Accept |       0.12 |    0.36634 |       0.02 |   0.027368 |    0.0060464 |
|   16 | Accept |       0.12 |    0.32528 |       0.02 |   0.027431 |       999.14 |
|   17 | Accept |   0.073333 |     6.1264 |       0.02 |   0.027298 |        995.2 |
|   18 | Accept |       0.12 |    0.34415 |       0.02 |   0.027263 |    0.0010034 |
|   19 | Accept |   0.033333 |    0.39577 |       0.02 |   0.027188 |     0.019231 |
|   20 | Accept |   0.033333 |    0.33763 |       0.02 |    0.02719 |     0.073621 |
|=====================================================================================================|
| Iter | Eval   | Objective  | Objective  | BestSoFar  | BestSoFar  | BoxConstrain |
|      | result |            | runtime    | (observed) | (estim.)   |              |
|=====================================================================================================|
|   21 | Accept |   0.026667 |    0.33277 |       0.02 |   0.026981 |       8.3113 |
|   22 | Accept |   0.033333 |    0.39717 |       0.02 |   0.026891 |       1.1945 |
|   23 | Accept |       0.12 |    0.33431 |       0.02 |   0.026978 |       2.3901 |
```

MATLAB Production Server

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



Integrate analytics with systems



```
for k=1:max
 x = fft(dat
 y = 20*log1
```

Embedded Hardware

C, C++

HDL

PLC

Enterprise Systems

Standalone Application

Excel Add-in

Hadoop/ Spark

C/C++

Java

Python

.NET

MATLAB Production Server

MATLAB Runtime

Key Takeaways

1

MATLAB Analytics work with **business** and **engineering data**

- Utilize all of your data.

2

MATLAB enables **domain experts to do Data Science**

- Apply advanced analytics techniques.

3

MATLAB Analytics **run anywhere**

- Operationalize analytics to enterprise systems and embedded devices.

Resources to learn and get started

mathworks.com/big-data

Big Data with MATLAB

Trial software Contact sales

Demo: MATLAB Tall Arrays in Action

Watch video

Predictive Analytics with MATLAB

Use machine learning with big data for engineering-driven analytics.

Download white paper

MATLAB® provides a single, high-performance environment for working with big data. MATLAB is:

Easy — Use familiar MATLAB functions and syntax to work with big datasets, even if they don't fit in memory.

Convenient — Work with the big data storage systems you already use, including traditional file systems, **SQL** and **NoSQL databases**, and **Hadoop/HDFS**.

Scalable — Use the processing platform that suits your needs, from your local **desktop machine** to Hadoop — without rewriting your algorithms.

Advanced Crash Detection: The Road from Deployment to Production

Watch video (4:46)

Beyond Excel: The Manager's Guide to Solving the Big Data Conundrum

mathworks.com/machine-learning

Machine Learning with MATLAB

Overview Examples

Trial software Contact sales

Machine Learning with MATLAB Webinar

Learn how to get started using machine learning tools to detect patterns and build predictive models from your data sets.

Watch video

Engineers and data scientists work with large amounts of data in a variety of formats such as sensor, image, video, telemetry, databases, and more. They use machine learning to find patterns in data and to build models that predict future outcomes based on historical data. With MATLAB®, you have immediate access to prebuilt functions, extensive toolboxes, and specialized apps for classification, regression, and clustering. You can:

- Compare approaches such as logistic regression, classification trees, support vector machines, ensemble methods, and deep learning.
- Use model refinement and reduction techniques to create an accurate model that best captures the predictive power of your data.
- Integrate machine learning models into enterprise systems, clusters, and clouds, and target models to real-time embedded hardware.

Machine Learning with MATLAB

Machine Learning with MATLAB Overview

Deep Learning in 11 Lines of MATLAB Code

Deep Learning with MATLAB

Contact sales Get a trial Read: The Netflix Prize and Production Machine Learning

Choosing the Best Classification Model and Avoiding Overfitting

Download white paper

Explore Products for Machine Learning

Statistics and Machine Learning Toolbox™

Neural Network Toolbox™

Computer Vision System Toolbox™

Fuzzy Logic Toolbox™

Machine Learning Challenges: Choosing the Best Model and Avoiding Overfitting

WHITE PAPER

eBook

MathWorks

Introducing Machine Learning