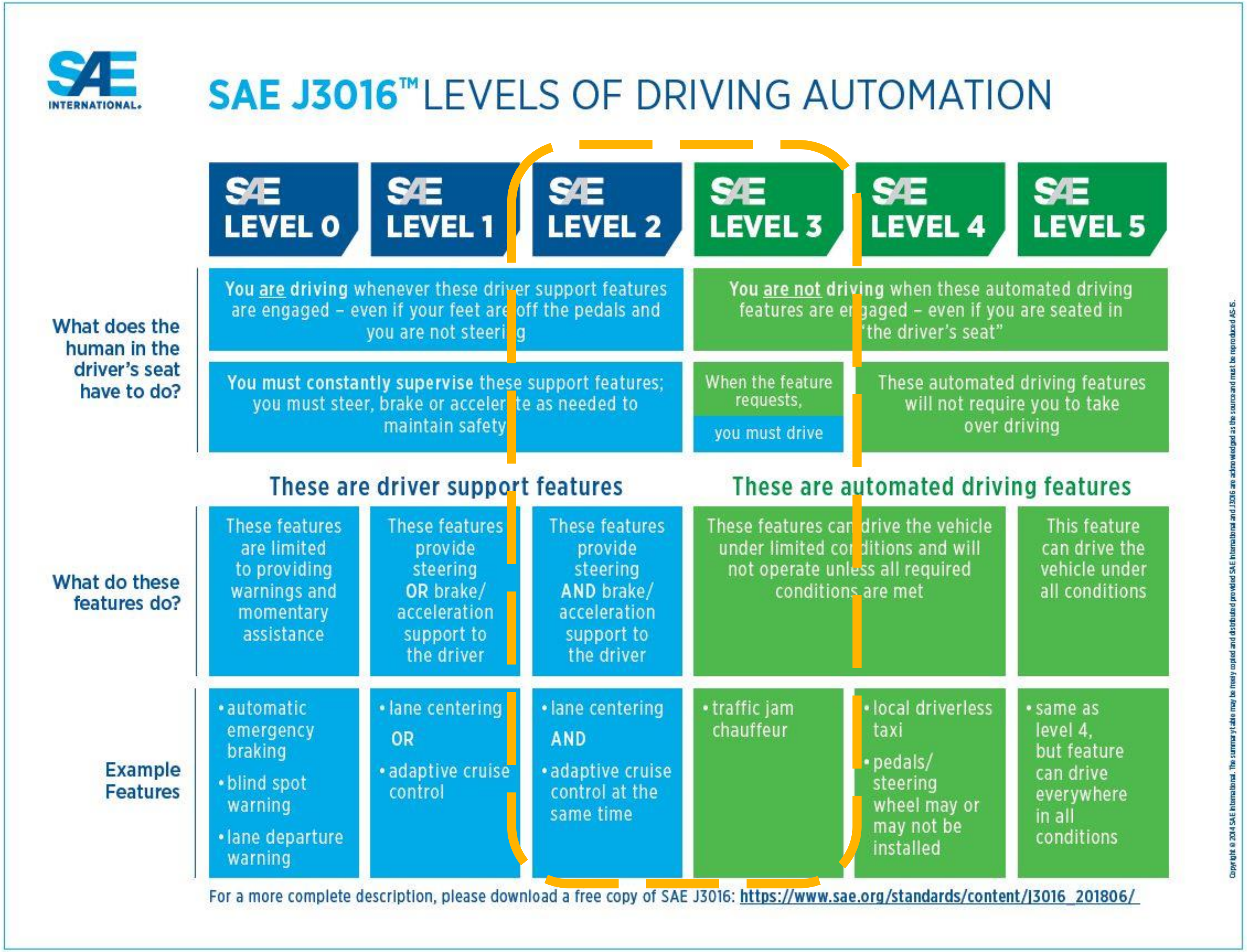
TARGET DETECTION & CLASSIFICATION IN RADAR POINT CLOUD USING MATLAB

Mr. Suresh Selvam, RNTBCI
Mr. Naresh Babu Napa Thulasiayya, RNTBCI

AGENDA

- SAE - ADAS LEVELS DEFINITION
- ADAS SENSORS
- ADAS CATEGORIES
- OUR GOAL
- FLOW CHART
- EXPERIMENTAL RESULTS
- SUMMARY

SAE – ADAS LEVEL DEFINITION



SAE INTERNATIONAL

SAE J3016™ LEVELS OF DRIVING AUTOMATION

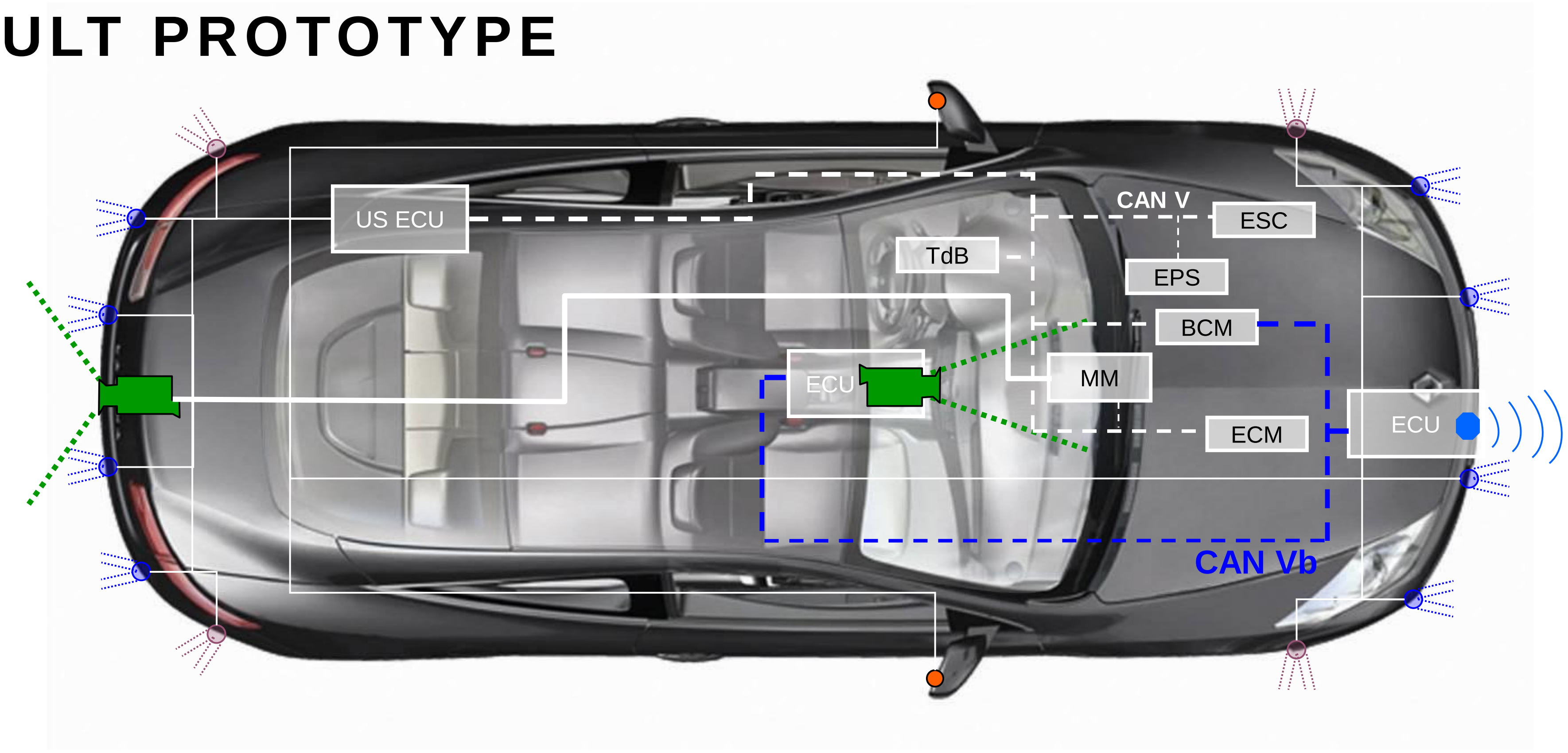
	SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
What does the human in the driver's seat have to do?	**You <u>are</u> driving** whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			**You <u>are not</u> driving** when these automated driving features are engaged – even if you are seated in "the driver's seat"		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
	These are driver support features			These are automated driving features		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/ acceleration support to the driver	These features provide steering AND brake/ acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met		This feature can drive the vehicle under all conditions
Example Features	• automatic emergency braking • blind spot warning • lane departure warning	• lane centering OR • adaptive cruise control	• lane centering AND • adaptive cruise control at the same time	• traffic jam chauffeur	• local driverless taxi • pedals/ steering wheel may or may not be installed	• same as level 4, but feature can drive everywhere in all conditions

For a more complete description, please download a free copy of SAE J3016: https://www.sae.org/standards/content/J3016_201806/

Source: www.sae.org

ADAS SENSOR

RENAULT PROTOTYPE



○ Ultra-sonic sensors type 1
For UPA only

○ Ultra-sonic sensors type 2
For BSW only

● BSW LED

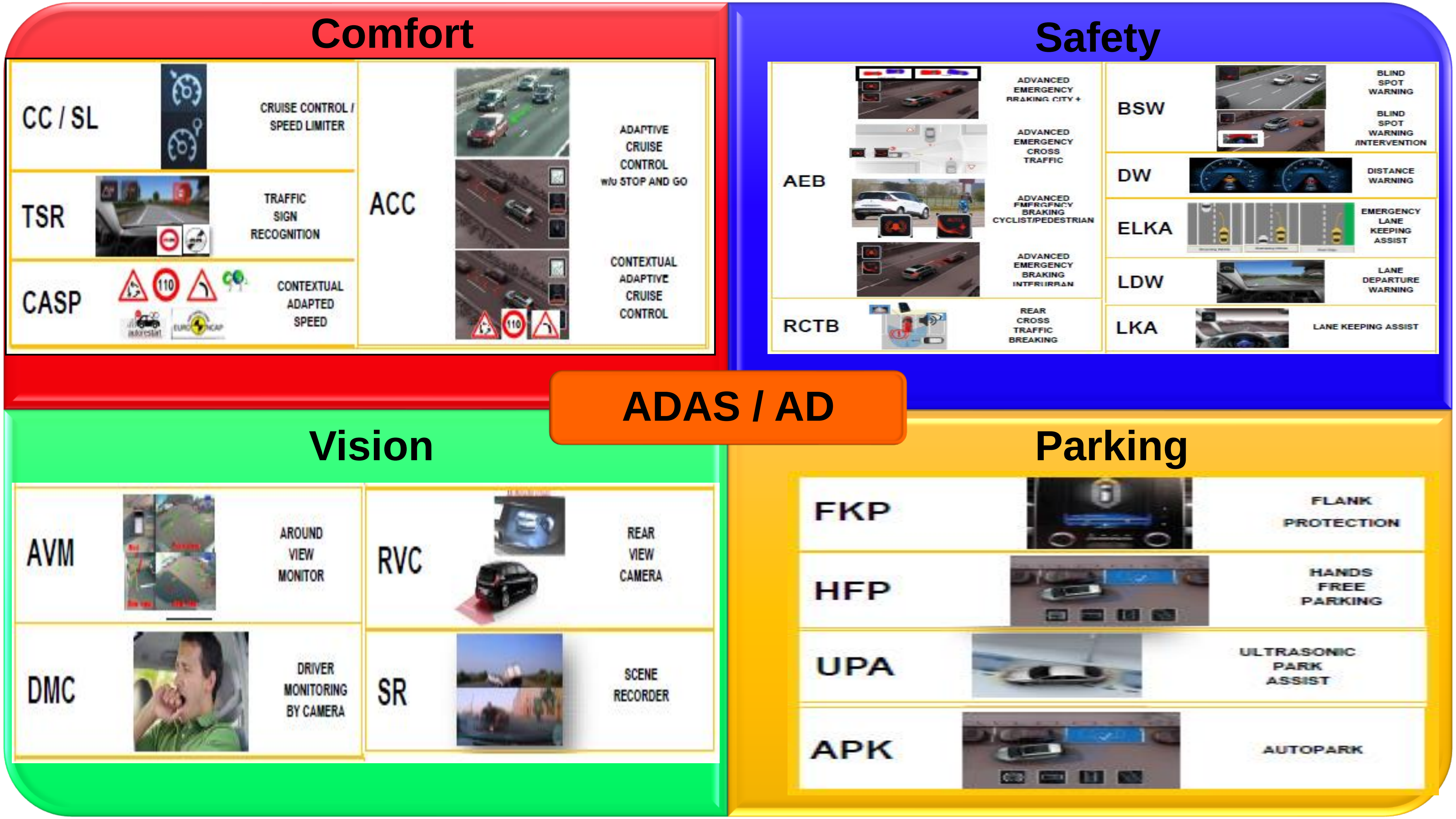
US ECU Calculateur Ultra son

ECU Radar

ECU Caméra Frontale

Caméra Arrière

ADAS CATEGORIES



ADAS / AD

Comfort

- CC / SL — CRUISE CONTROL / SPEED LIMITER
- TSR — TRAFFIC SIGN RECOGNITION
- CASP — CONTEXTUAL ADAPTED SPEED
- ACC — ADAPTIVE CRUISE CONTROL w/o STOP AND GO
- ACC — CONTEXTUAL ADAPTIVE CRUISE CONTROL

Safety

- AEB — ADVANCED EMERGENCY BRAKING CITY +
- AEB — ADVANCED EMERGENCY CROSS TRAFFIC
- AEB — ADVANCED EMERGENCY BRAKING CYCLIST/PEDESTRIAN
- AEB — ADVANCED EMERGENCY BRAKING INTERURBAN
- RCTB — REAR CROSS TRAFFIC BREAKING
- BSW — BLIND SPOT WARNING
- BSW — BLIND SPOT WARNING /INTERVENTION
- DW — DISTANCE WARNING
- ELKA — EMERGENCY LANE KEEPING ASSIST
- LDW — LANE DEPARTURE WARNING
- LKA — LANE KEEPING ASSIST

Vision

- AVM — AROUND VIEW MONITOR
- DMC — DRIVER MONITORING BY CAMERA
- RVC — REAR VIEW CAMERA
- SR — SCENE RECORDER

Parking

- FKP — FLANK PROTECTION
- HFP — HANDS FREE PARKING
- UPA — ULTRASONIC PARK ASSIST
- APK — AUTOPARK

TARGET DETECTION & CLASSIFICATION

Goal

- Our goal is to have an optimized confusion (decision) matrix which minimize False Positive Rate (FPR) and improve the classification accuracy/reliable Ground truth data considering different infra structures, RADAR mounting position and weather conditions. It’s also enables us to benchmark RADAR performance and determine the ADAS sensor configuration suitable to Renault-Nissan vehicle lines.

Input

- Point cloud data are extracted from ADAS sensor in RENAULT Vehicle

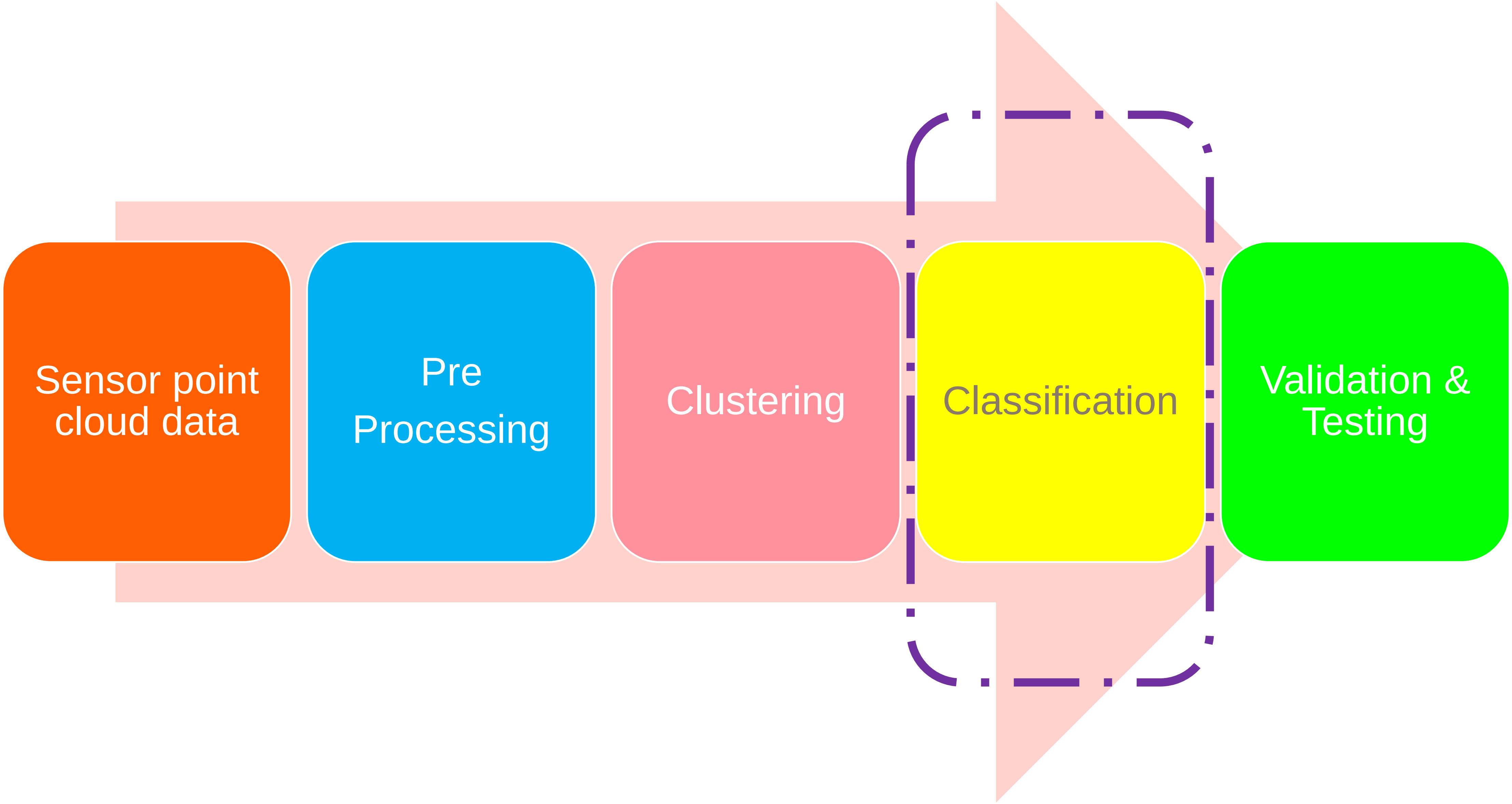
Problem Statement

- Target detection & Classification using point cloud data.
- Automatically predict the primary classes with help of Optimum Classification Model
- The primary classes are CAR, TRUCK, POWERED TWO-WHEELER, BICYCLE and PEDESTRIANS

Solution

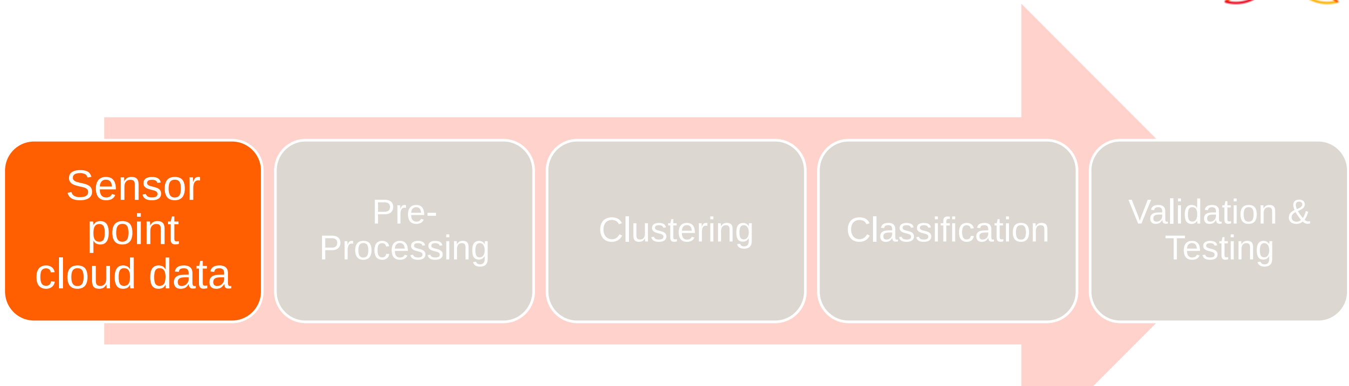
- Target detection is achieved by DBSCAN Clustering technique.
- Statistics and Machine Learning Toolbox helped for Classification Model selection & Training
- Classification Model is Trained based on Cluster parameters (Length, Width, Speed and RCS)

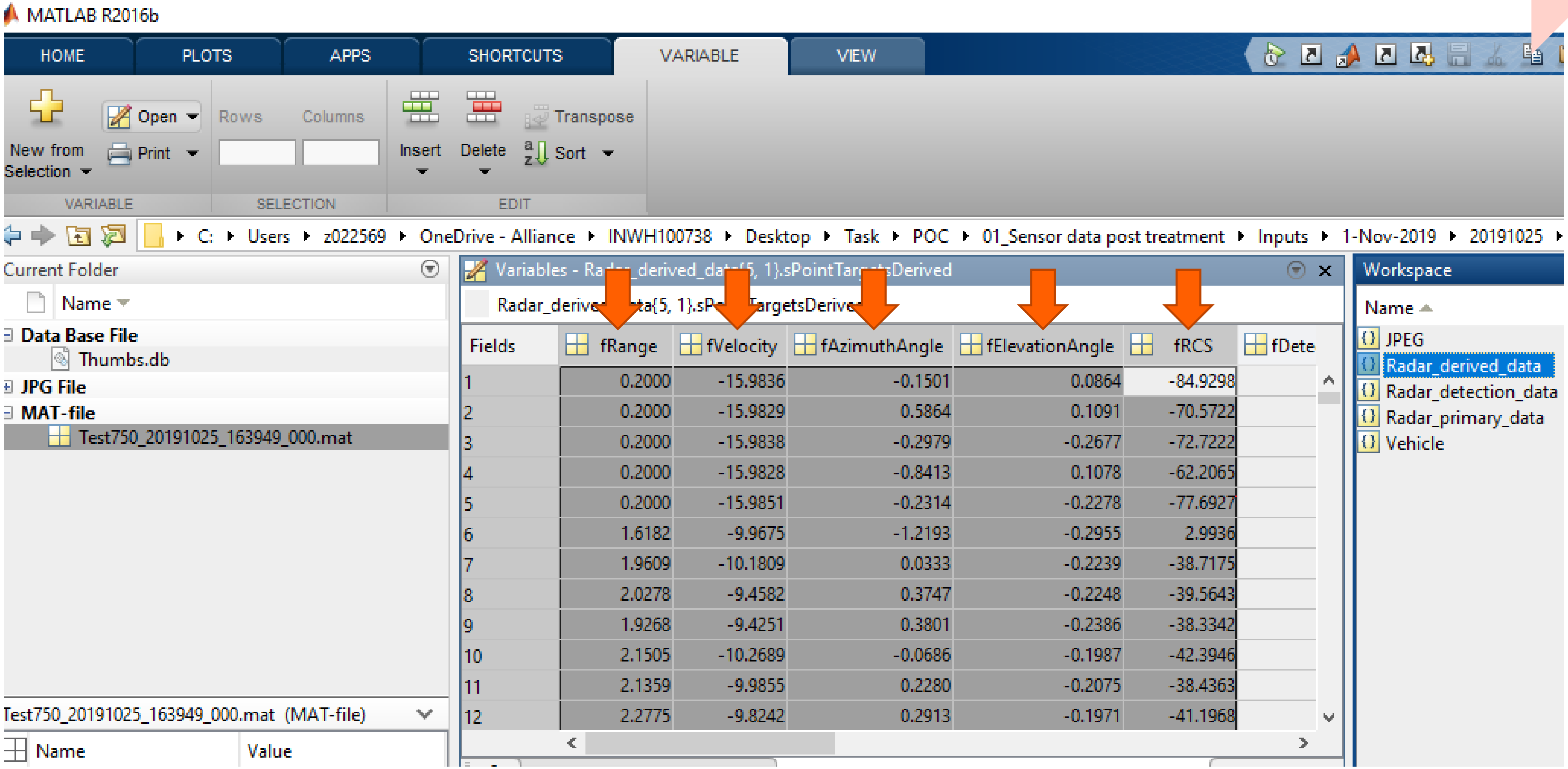
FLOW CHART


Sensor point cloud data
Pre Processing
Clustering
Classification
Validation & Testing

SENSOR POINT CLOUD DATA

- ✓ Point cloud data are captured at sensor level
- ✓ Data stored in terms of Mat file format for further processing





C: ▸ Users ▸ z022569 ▸ OneDrive - Alliance ▸ INWH100738 ▸ Desktop ▸ Task ▸ POC ▸ 01_Sensor data post treatment ▸ Inputs ▸ 1-Nov-2019 ▸ 20191025 ▸

Current Folder

Name

- Data Base File
  - Thumbs.db
- JPG File
- MAT-file
  - Test750_20191025_163949_000.mat

Test750_20191025_163949_000.mat (MAT-file)

Name | Value

Variables - Radar_derived_data{5, 1}.sPointTargetsDerived

Radar_derived_data{5, 1}.sPointTargetsDerived

Fields	fRange	fVelocity	fAzimuthAngle	fElevationAngle	fRCS	fDete
1	0.2000	-15.9836	-0.1501	0.0864	-84.9298	
2	0.2000	-15.9829	0.5864	0.1091	-70.5722	
3	0.2000	-15.9838	-0.2979	-0.2677	-72.7222	
4	0.2000	-15.9828	-0.8413	0.1078	-62.2065	
5	0.2000	-15.9851	-0.2314	-0.2278	-77.6927	
6	1.6182	-9.9675	-1.2193	-0.2955	2.9936	
7	1.9609	-10.1809	0.0333	-0.2239	-38.7175	
8	2.0278	-9.4582	0.3747	-0.2248	-39.5643	
9	1.9268	-9.4251	0.3801	-0.2386	-38.3342	
10	2.1505	-10.2689	-0.0686	-0.1987	-42.3946	
11	2.1359	-9.9855	0.2280	-0.2075	-38.4363	
12	2.2775	-9.8242	0.2913	-0.1971	-41.1968	

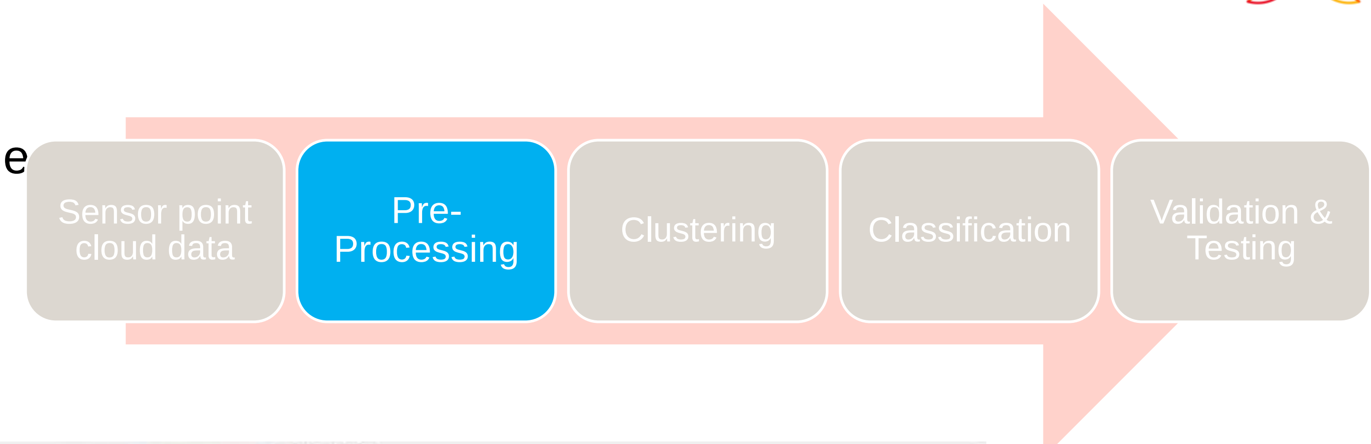
Workspace

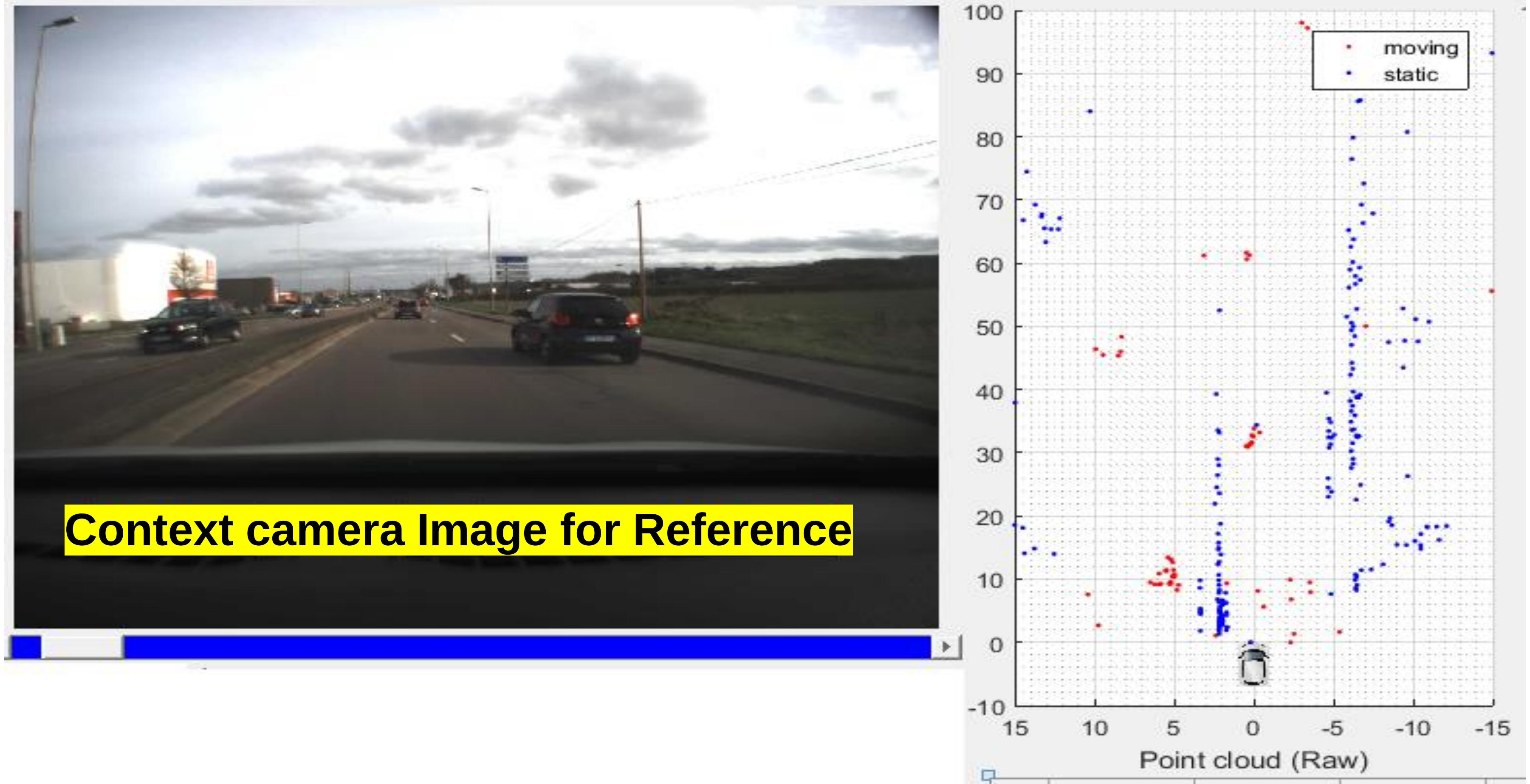
Name

- JPEG
- Radar_derived_data
- Radar_detection_data
- Radar_primary_data
- Vehicle

PRE-PROCESSING

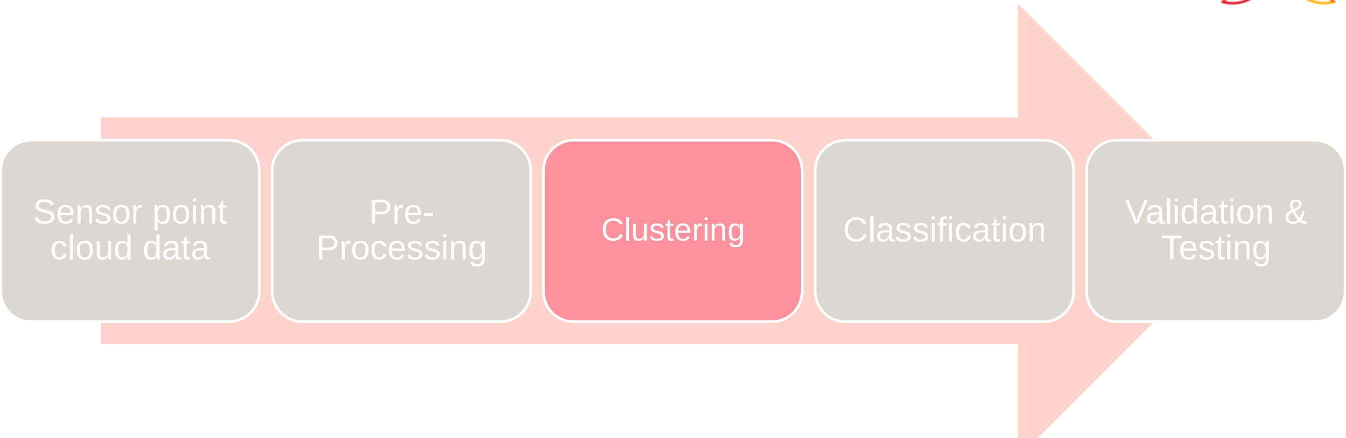
- ✓ Each Frame data are aligned and grouped as per the parameter names
- ✓ Processed data are plotted in bird eye view along with CC Image




Context camera Image for Reference

CLUSTERING

- ✓ Moving point cloud data are grouped as a Cluster
- ✓ Cluster for Target detection using DBSCAN clustering algorithm
- ✓ Rectangles are created as per the Cluster parameters



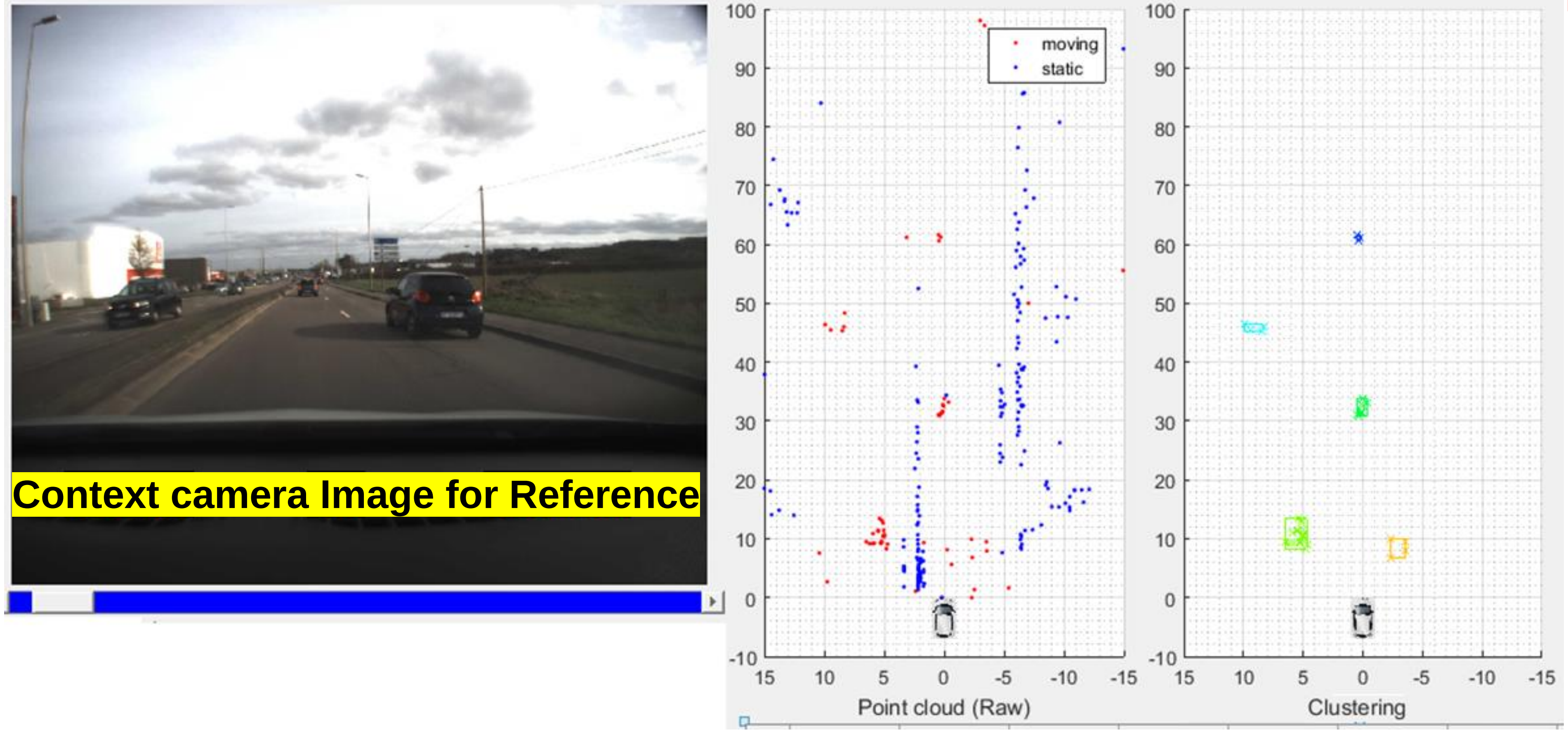
Sensor point cloud data

Pre-Processing

Clustering

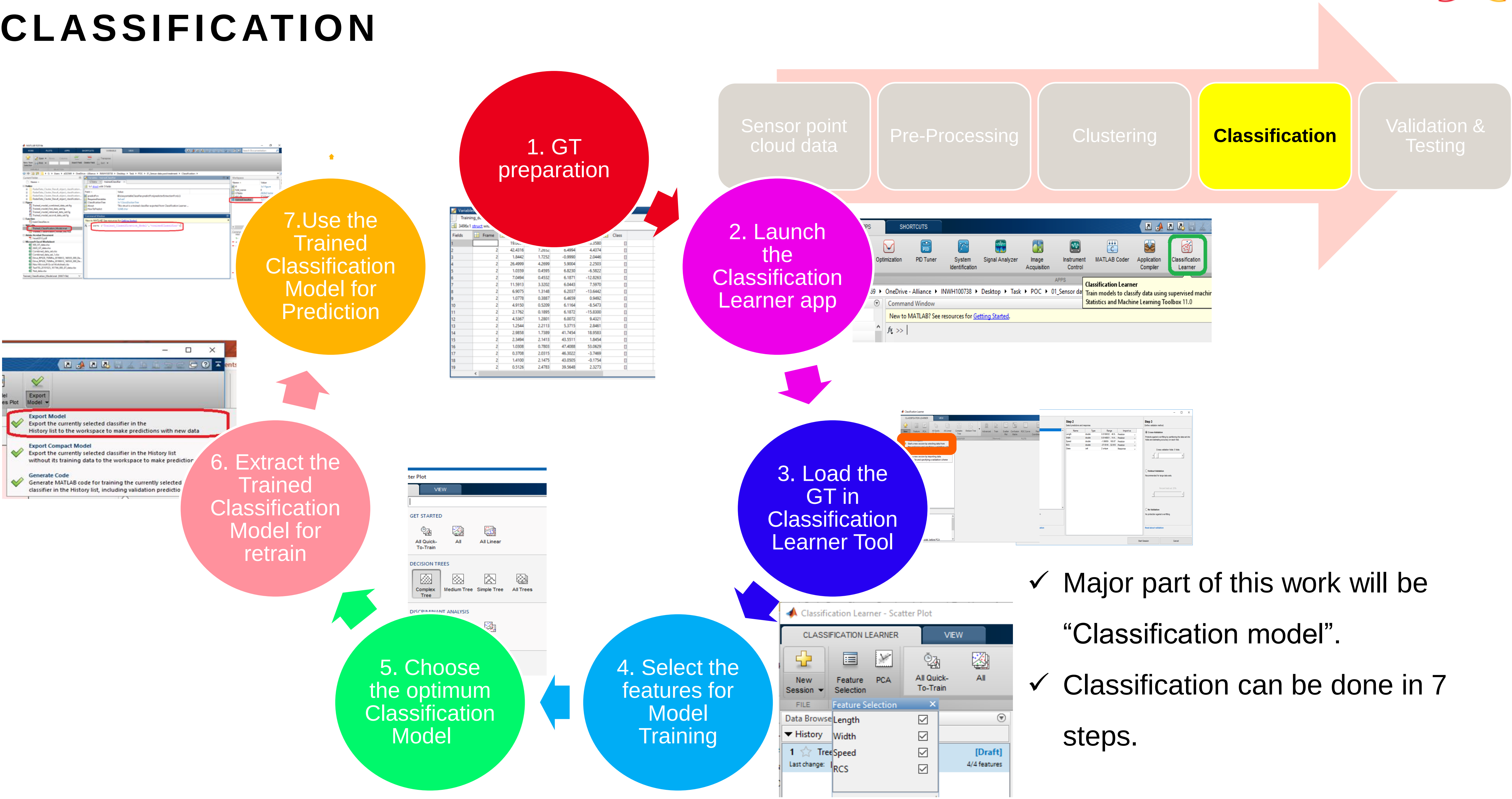
Classification

Validation & Testing



Context camera Image for Reference

CLASSIFICATION



3496x1 struct wi

Fields	Frame					Class
1		19.06	[illegible]	[illegible]	-3.3580	[]
2	2	42.4316	7.2652	6.4994	4.4374	[]
3	2	1.8442	1.7252	-0.9990	2.0446	[]
4	2	26.4999	4.2699	5.9004	2.2503	[]
5	2	1.0359	0.4595	6.8230	-6.5822	[]
6	2	7.0494	0.4532	6.1871	-12.8263	[]
7	2	11.5913	3.3202	6.0443	7.5970	[]
8	2	6.9075	1.3148	6.2037	-13.6442	[]
9	2	1.0778	0.3887	6.4659	0.9492	[]
10	2	4.9150	0.5209	6.1164	-8.5473	[]
11	2	2.1762	0.1895	6.1872	-15.8300	[]
12	2	4.5367	1.2801	6.0072	9.4321	[]
13	2	1.2544	2.2113	5.3715	2.8461	[]
14	2	2.9858	1.7389	41.7454	18.9583	[]
15	2	2.3494	2.1413	43.5511	1.8454	[]
16	2	1.0308	0.7803	47.4088	53.0629	[]
17	2	0.3708	2.0315	46.3022	-3.7469	[]
18	2	1.4100	2.1475	43.0505	-0.1754	[]
19	2	0.5126	2.4783	39.5648	2.3273	[]

Classification Learner
Train models to classify data using supervised machir
Statistics and Machine Learning Toolbox 11.0

New to MATLAB? See resources for Getting Started.

fx >>

Export Model
Export the currently selected classifier in the History list to the workspace to make predictions with new data

Export Compact Model
Export the currently selected classifier in the History list without its training data to the workspace to make predictior

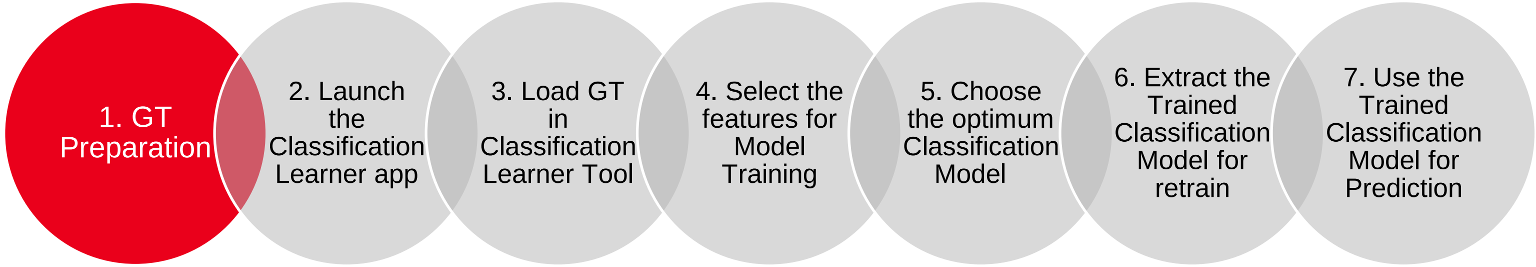
Generate Code
Generate MATLAB code for training the currently selected classifier in the History list, including validation predictio

Feature Selection

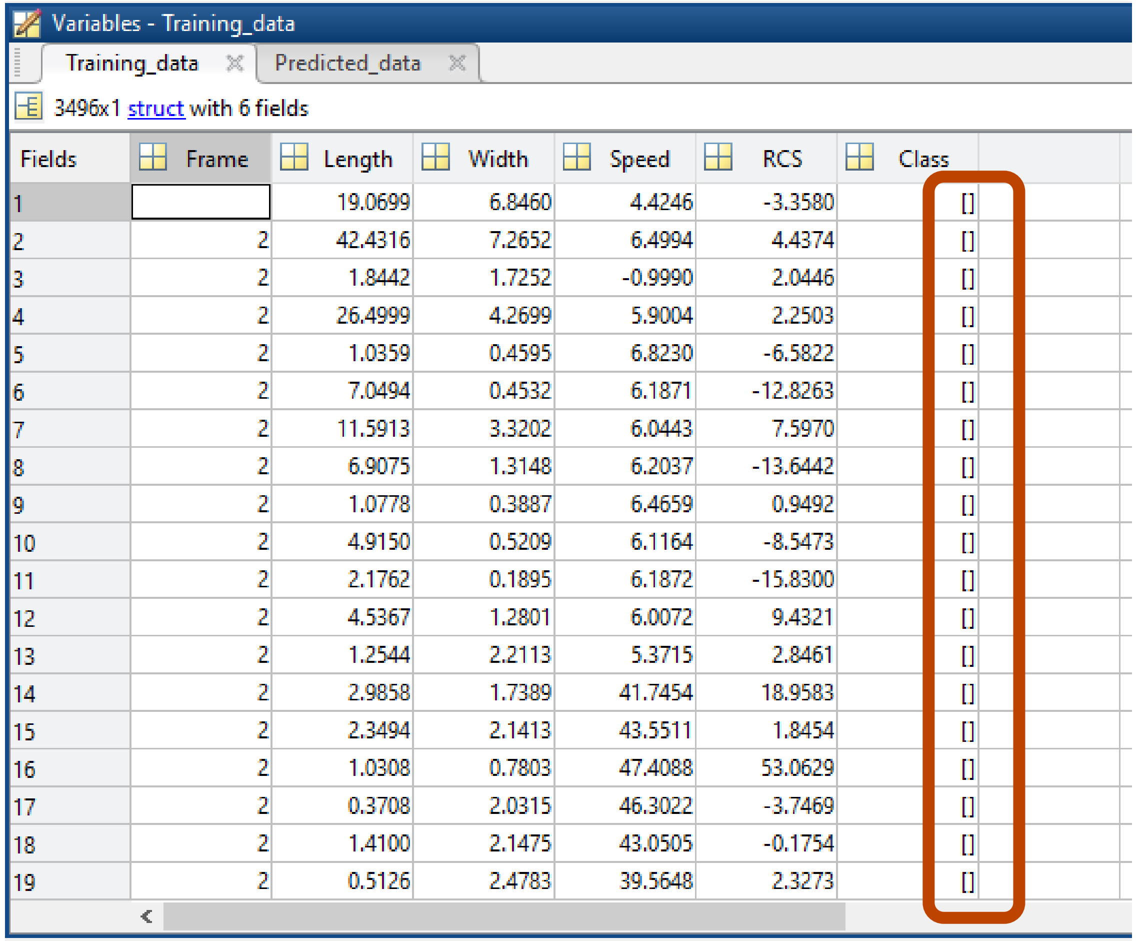
Length	☑
Width	☑
Speed	☑
RCS	☑

- ✓ Major part of this work will be "Classification model".
- ✓ Classification can be done in 7 steps.

GT PREPARATION


1. GT Preparation
2. Launch the Classification Learner app
3. Load GT in Classification Learner Tool
4. Select the features for Model Training
5. Choose the optimum Classification Model
6. Extract the Trained Classification Model for retrain
7. Use the Trained Classification Model for Prediction

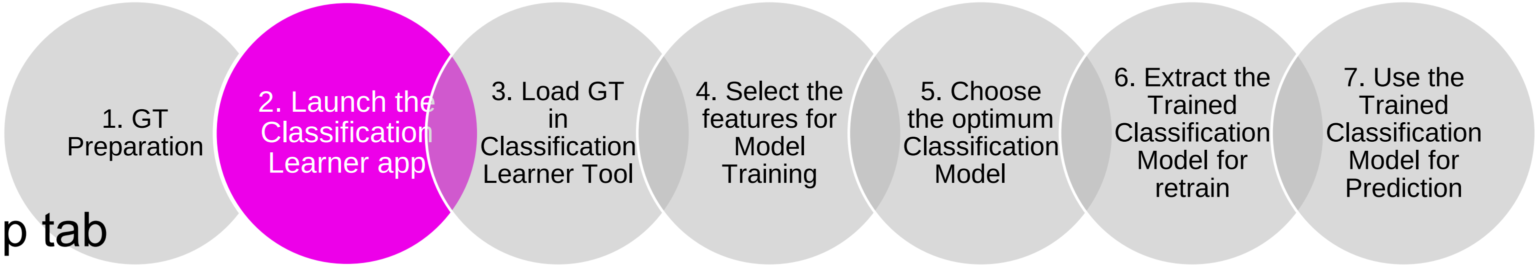
- ✓ Ground Truth Information needed for Train the Classification Model
- ✓ GT information gathered from Clustering Parameters [Length, Width, Speed, RCS]
- ✓ Manually add the class name with help of the reference camera image



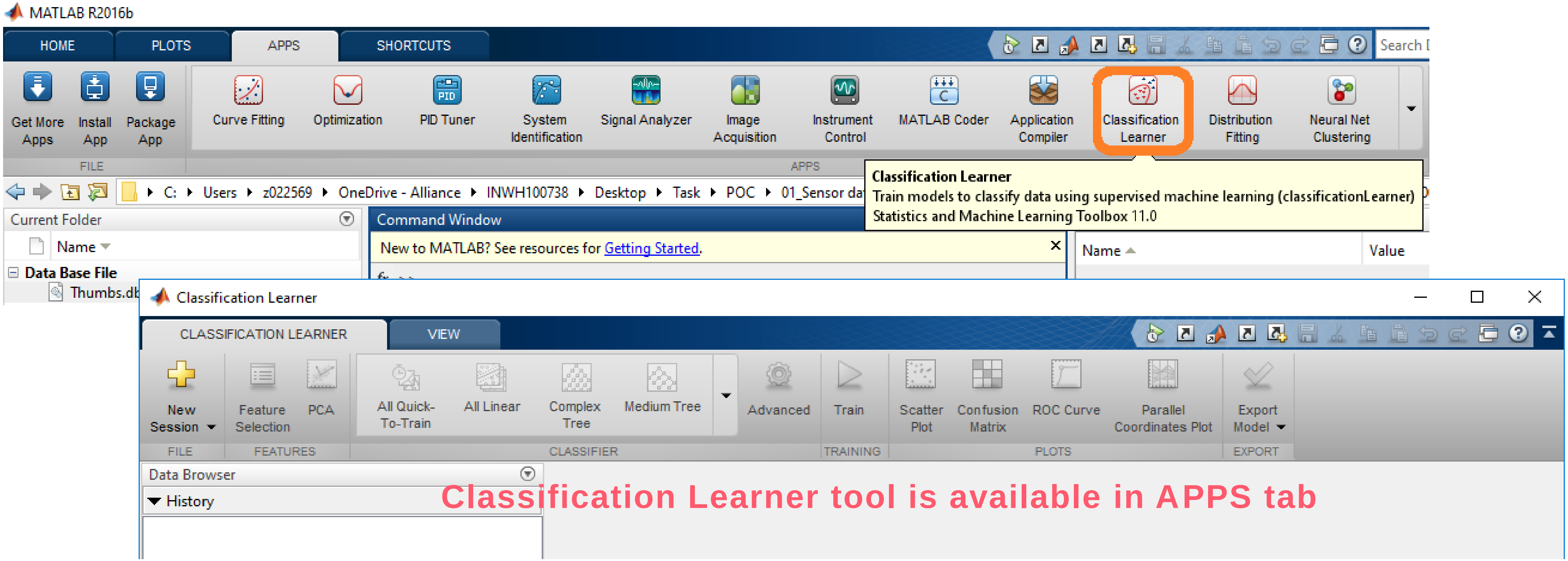
3496x1 struct with 6 fields

Fields	Frame	Length	Width	Speed	RCS	Class
1		19.0699	6.8460	4.4246	-3.3580	[]
2	2	42.4316	7.2652	6.4994	4.4374	[]
3	2	1.8442	1.7252	-0.9990	2.0446	[]
4	2	26.4999	4.2699	5.9004	2.2503	[]
5	2	1.0359	0.4595	6.8230	-6.5822	[]
6	2	7.0494	0.4532	6.1871	-12.8263	[]
7	2	11.5913	3.3202	6.0443	7.5970	[]
8	2	6.9075	1.3148	6.2037	-13.6442	[]
9	2	1.0778	0.3887	6.4659	0.9492	[]
10	2	4.9150	0.5209	6.1164	-8.5473	[]
11	2	2.1762	0.1895	6.1872	-15.8300	[]
12	2	4.5367	1.2801	6.0072	9.4321	[]
13	2	1.2544	2.2113	5.3715	2.8461	[]
14	2	2.9858	1.7389	41.7454	18.9583	[]
15	2	2.3494	2.1413	43.5511	1.8454	[]
16	2	1.0308	0.7803	47.4088	53.0629	[]
17	2	0.3708	2.0315	46.3022	-3.7469	[]
18	2	1.4100	2.1475	43.0505	-0.1754	[]
19	2	0.5126	2.4783	39.5648	2.3273	[]

LAUNCH THE CLASSIFICATION LEARNER APP


1. GT Preparation
2. Launch the Classification Learner app
3. Load GT in Classification Learner Tool
4. Select the features for Model Training
5. Choose the optimum Classification Model
6. Extract the Trained Classification Model for retrain
7. Use the Trained Classification Model for Prediction

✓ Select the “Classification Learner” App in App tab



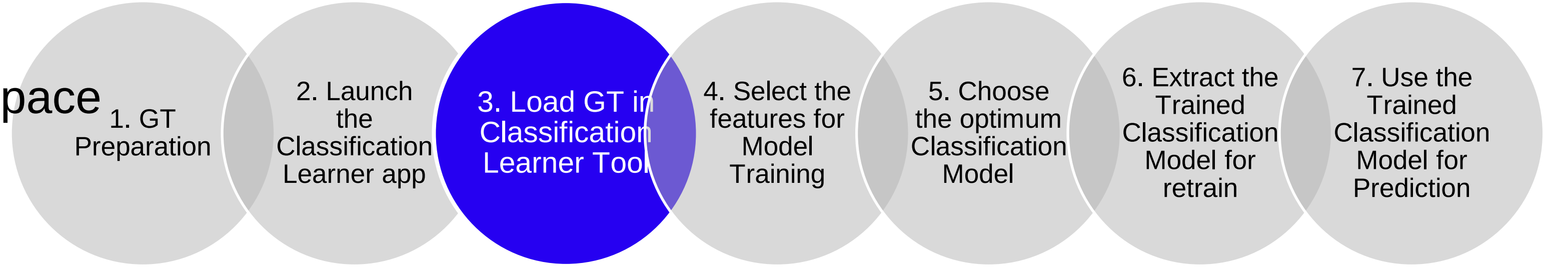
Classification Learner
Train models to classify data using supervised machine learning (classificationLearner)
Statistics and Machine Learning Toolbox 11.0

New to MATLAB? See resources for Getting Started.

Classification Learner tool is available in APPS tab

LOAD THE GT IN CLASSIFICATION LEARNER TOOL

- ✓ Load the excel GT values to MATLAB Workspace
- ✓ Import the data in Classification Learner Tool


1. GT Preparation

2. Launch the Classification Learner app

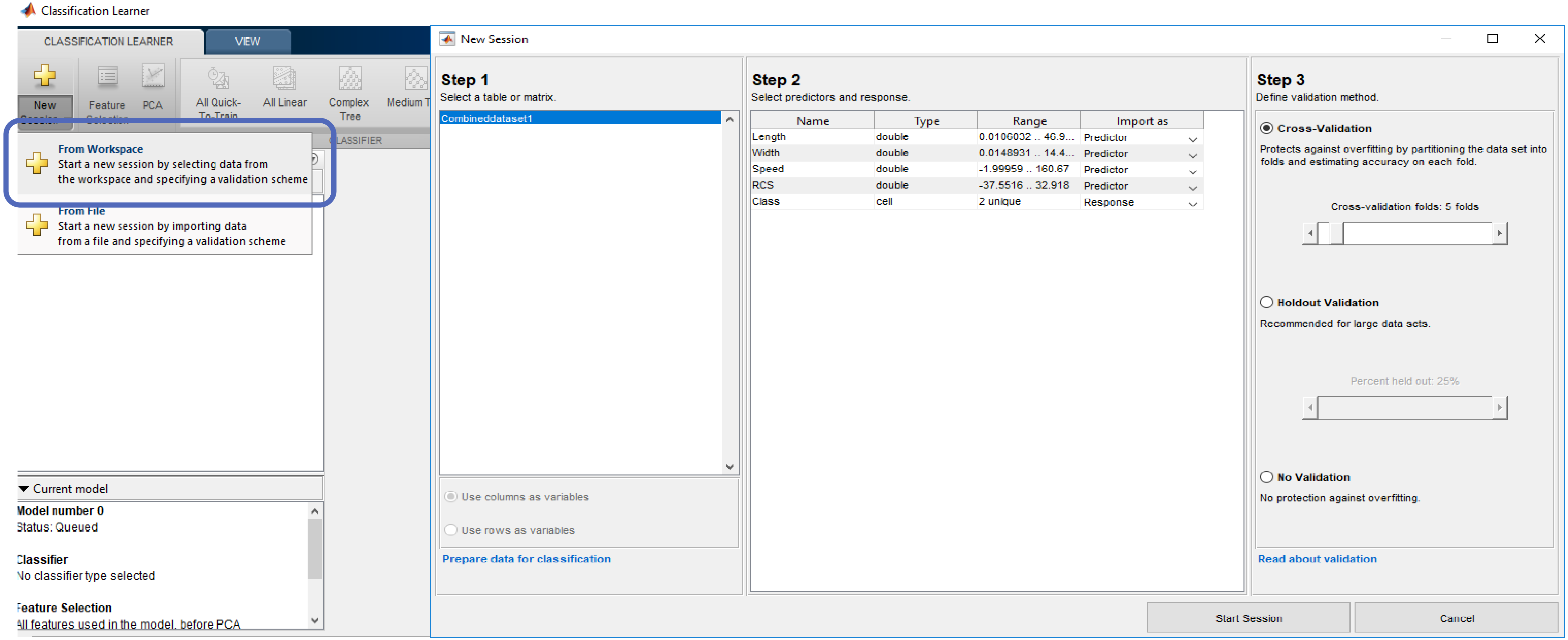
3. Load GT in Classification Learner Tool

4. Select the features for Model Training

5. Choose the optimum Classification Model

6. Extract the Trained Classification Model for retrain

7. Use the Trained Classification Model for Prediction



From Workspace
Start a new session by selecting data from the workspace and specifying a validation scheme

From File
Start a new session by importing data from a file and specifying a validation scheme

▼ Current model

Model number 0
Status: Queued

Classifier
No classifier type selected

Feature Selection
All features used in the model, before PCA

Step 1
Select a table or matrix.

Combineddataset1

○ Use columns as variables

○ Use rows as variables

Prepare data for classification

Step 2
Select predictors and response.

Name	Type	Range	Import as
Length	double	0.0106032 .. 46.9...	Predictor
Width	double	0.0148931 .. 14.4...	Predictor
Speed	double	-1.99959 .. 160.67	Predictor
RCS	double	-37.5516 .. 32.918	Predictor
Class	cell	2 unique	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.

Percent held out: 25%

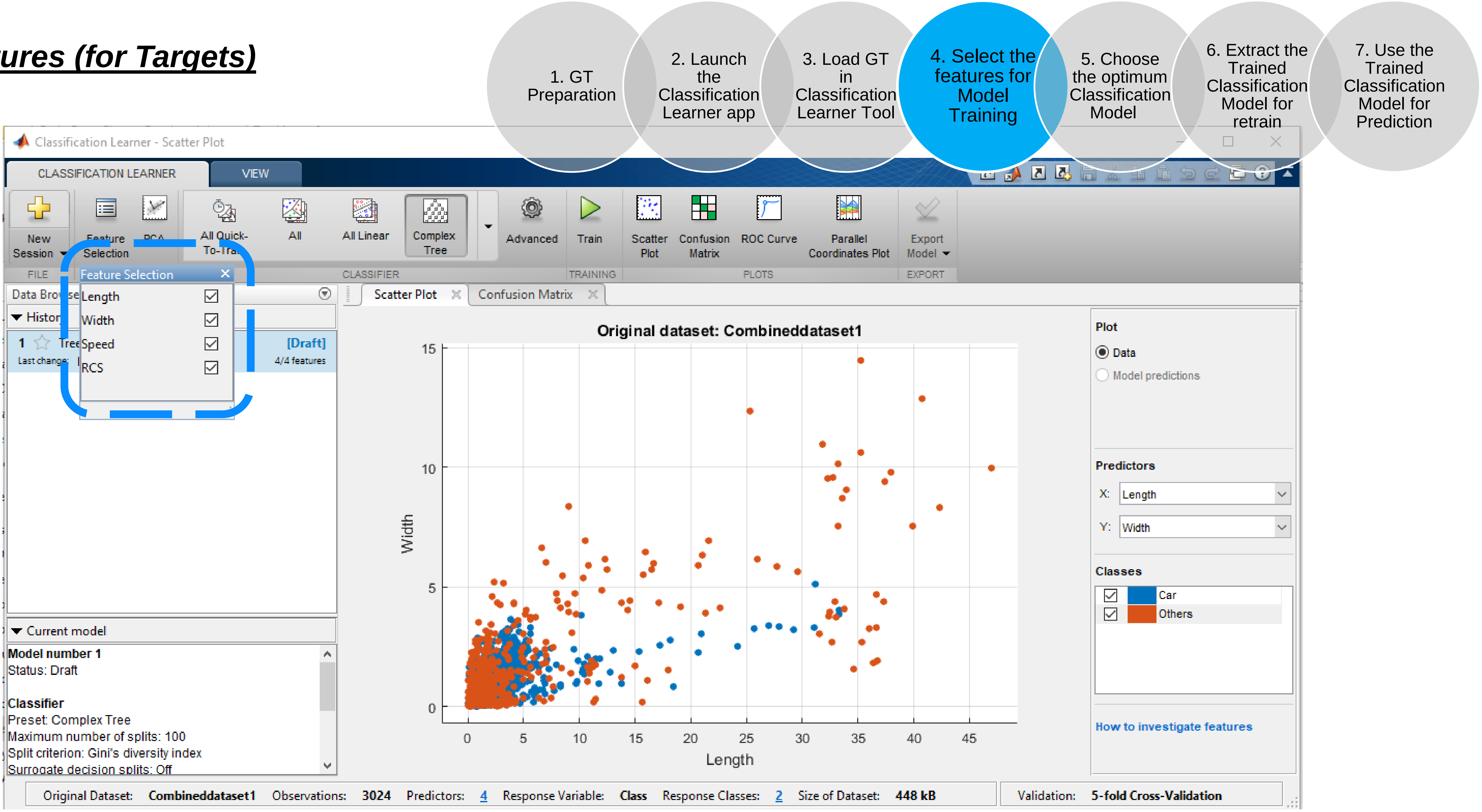
○ **No Validation**
No protection against overfitting.

Read about validation

SELECT THE FEATURES FOR MODEL TRAINING

List of features (for Targets)

- ✓ Length
- ✓ Width
- ✓ Speed
- ✓ RCS



1. GT Preparation

2. Launch the Classification Learner app

3. Load GT in Classification Learner Tool

4. Select the features for Model Training

5. Choose the optimum Classification Model

6. Extract the Trained Classification Model for retrain

7. Use the Trained Classification Model for Prediction

Classification Learner - Scatter Plot

Feature Selection

Length	☑
Width	☑
Speed	☑
RCS	☑

1 ☆ Tree [Draft]
Last change: 4/4 features

Current model

Model number 1
Status: Draft

Classifier
Preset: Complex Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off

Original dataset: Combineddataset1

Plot
● Data
○ Model predictions

Predictors
X: Length
Y: Width

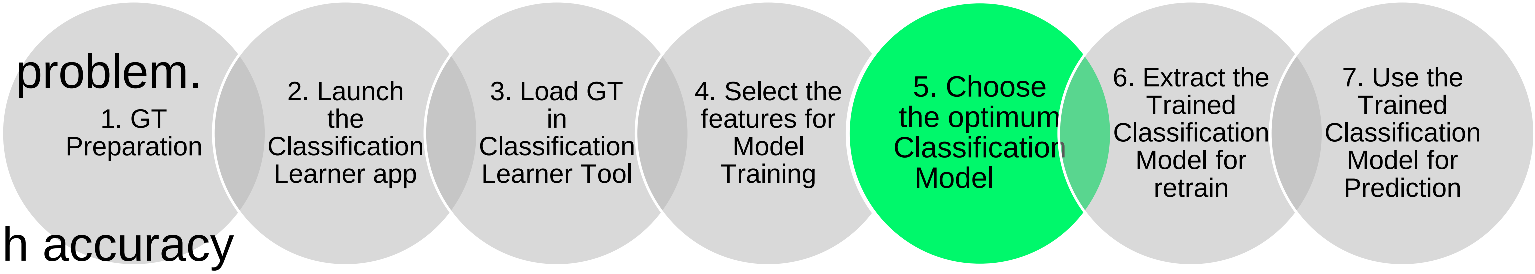
Classes
☑ 🟦 Car
☑ 🟧 Others

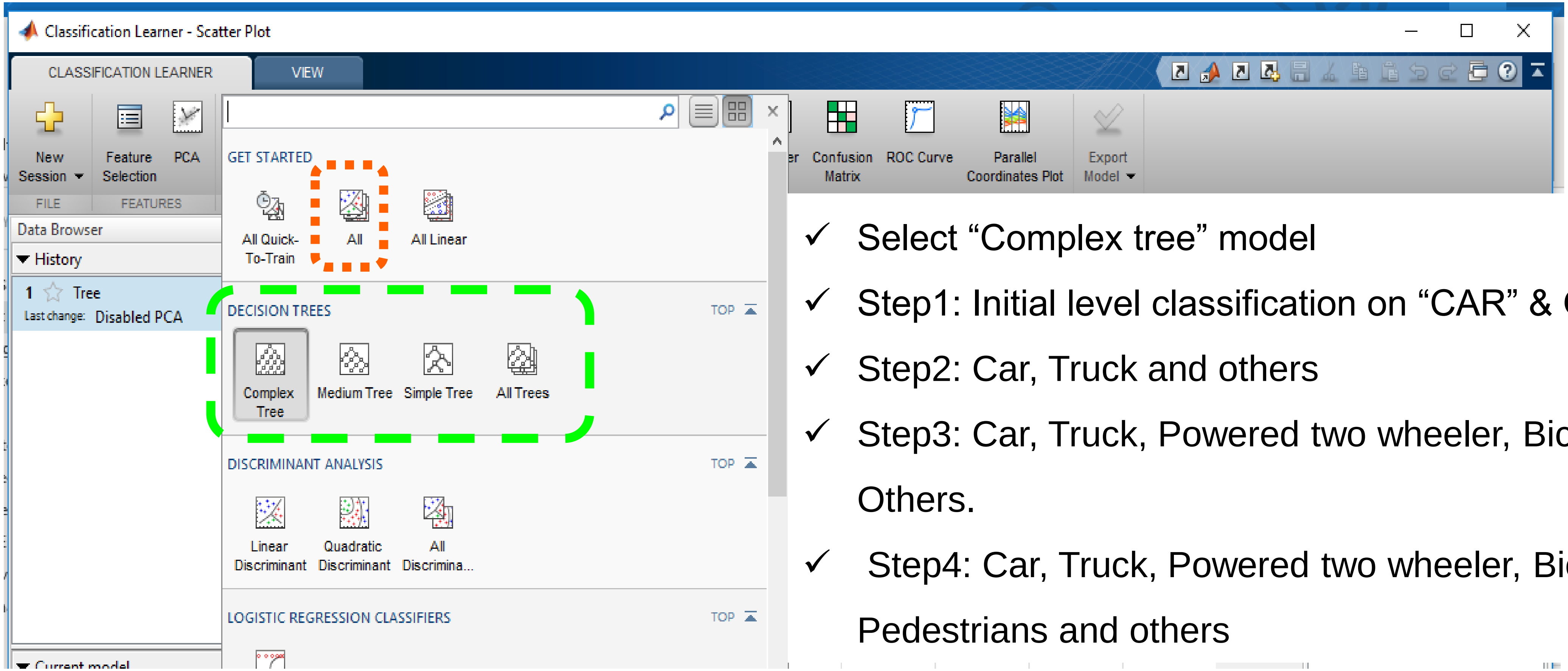
How to investigate features

Original Dataset: **Combineddataset1** Observations: **3024** Predictors: 4 Response Variable: **Class** Response Classes: 2 Size of Dataset: **448 kB** Validation: **5-fold Cross-Validation**

CHOOSE THE OPTIMUM CLASSIFICATION MODEL

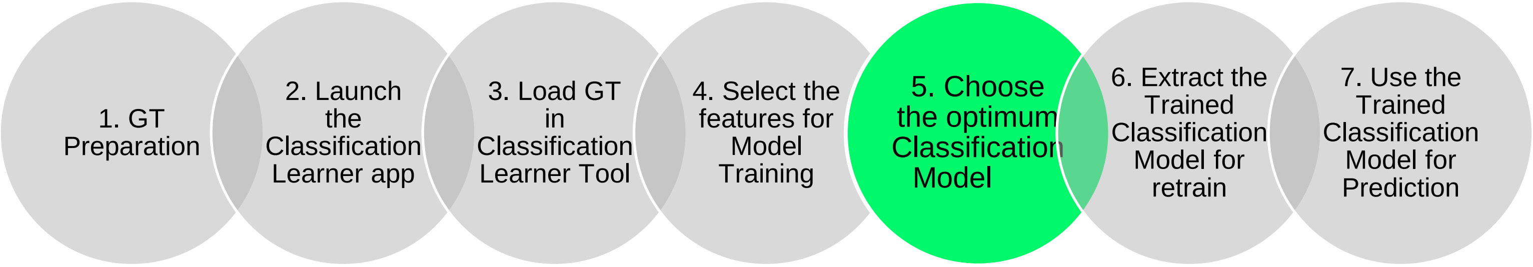
- ✓ “All Models” is not suitable for our research problem. Because found less accuracy.
- ✓ Applied “Complex Tree Model” and got high accuracy than “All Model” in Statistical and Machine learning tool box


1. GT Preparation
2. Launch the Classification Learner app
3. Load GT in Classification Learner Tool
4. Select the features for Model Training
5. Choose the optimum Classification Model
6. Extract the Trained Classification Model for retrain
7. Use the Trained Classification Model for Prediction



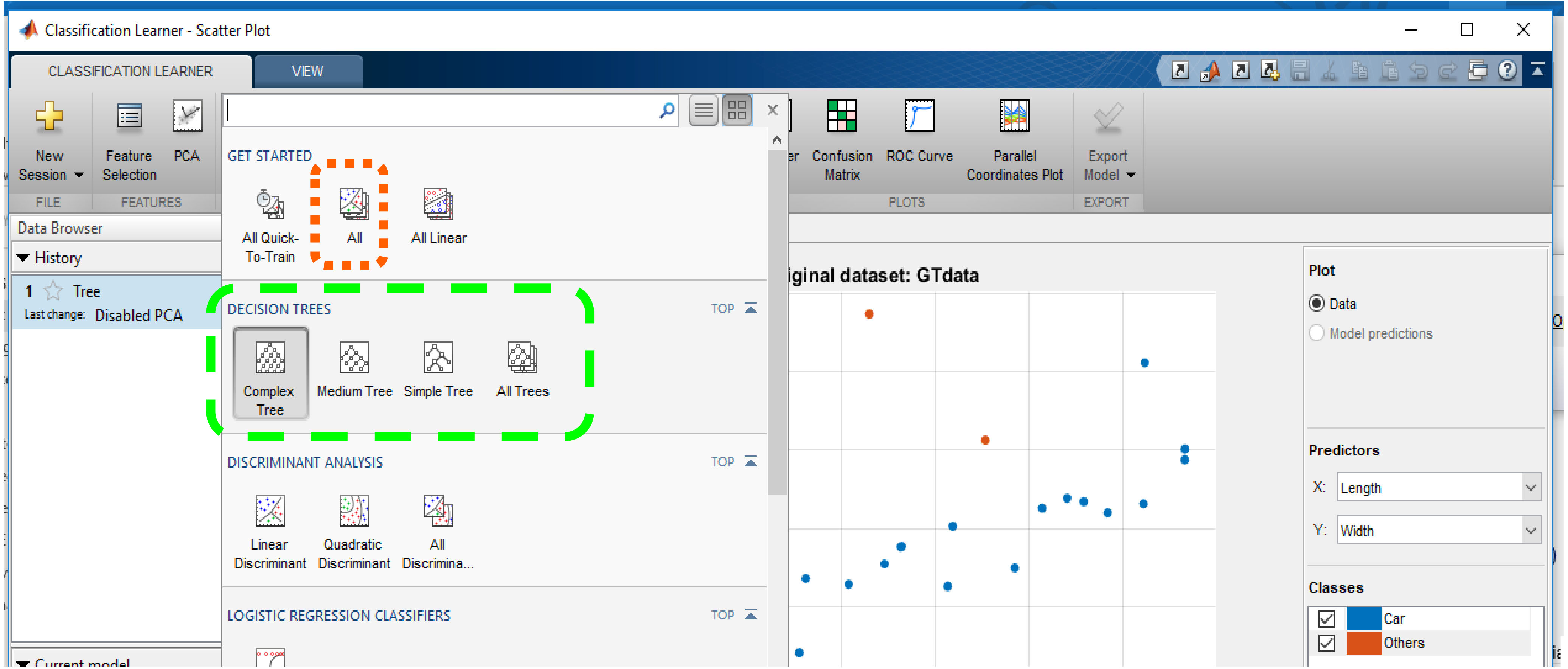
- ✓ Select “Complex tree” model
- ✓ Step1: Initial level classification on “CAR” & Others
- ✓ Step2: Car, Truck and others
- ✓ Step3: Car, Truck, Powered two wheeler, Bicycle & Others.
- ✓ Step4: Car, Truck, Powered two wheeler, Bicycle, Pedestrians and others

CHOOSE THE OPTIMUM CLASSIFICATION MODEL



1. GT Preparation
2. Launch the Classification Learner app
3. Load GT in Classification Learner Tool
4. Select the features for Model Training
5. Choose the optimum Classification Model
6. Extract the Trained Classification Model for retrain
7. Use the Trained Classification Model for Prediction

- ✓ Added value to this MATLAB tool
- ✓ Reduced manual effort by 80%
- ✓ Accuracy increased 95% (overall classes)
- ✓ Fast execution



1 ☆ Tree
Last change: Disabled PCA

iginal dataset: GTdata

Plot
● Data
○ Model predictions

Predictors
X: Length
Y: Width

Classes
☑ 🟦 Car
☑ 🟧 Others

EXTRACT THE TRAINED CLASSIFICATION MODEL FOR RETRAIN

1. GT Preparation

2. Launch the Classification Learner app

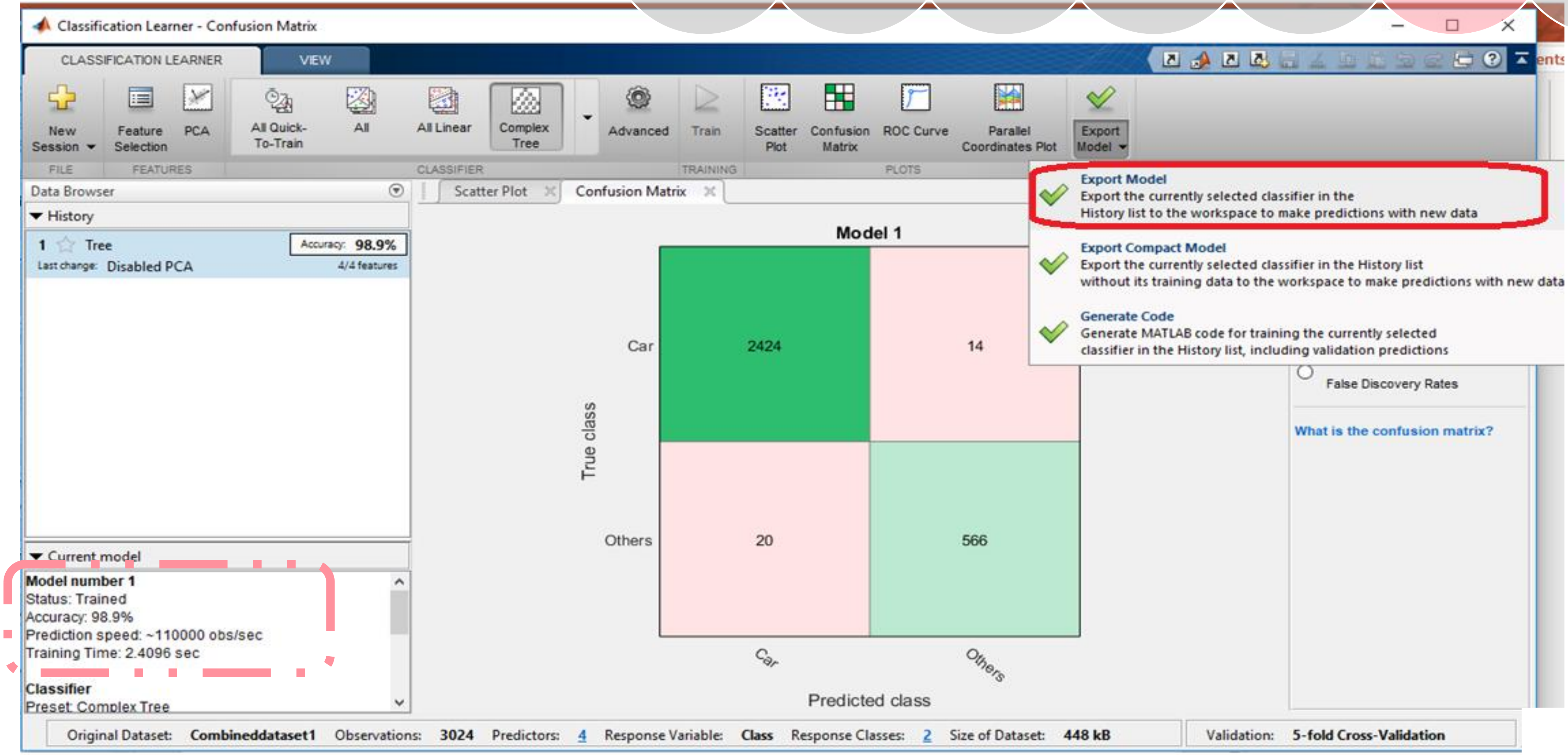
3. Load GT in Classification Learner Tool

4. Select the features for Model Training

5. Choose the optimum Classification Model

6. Extract the Trained Classification Model for retrain

7. Use the Trained Classification Model for Prediction



1 Tree Accuracy: 98.9%
Last change: Disabled PCA 4/4 features

Export Model
Export the currently selected classifier in the History list to the workspace to make predictions with new data

Export Compact Model
Export the currently selected classifier in the History list without its training data to the workspace to make predictions with new data

Generate Code
Generate MATLAB code for training the currently selected classifier in the History list, including validation predictions

Model 1

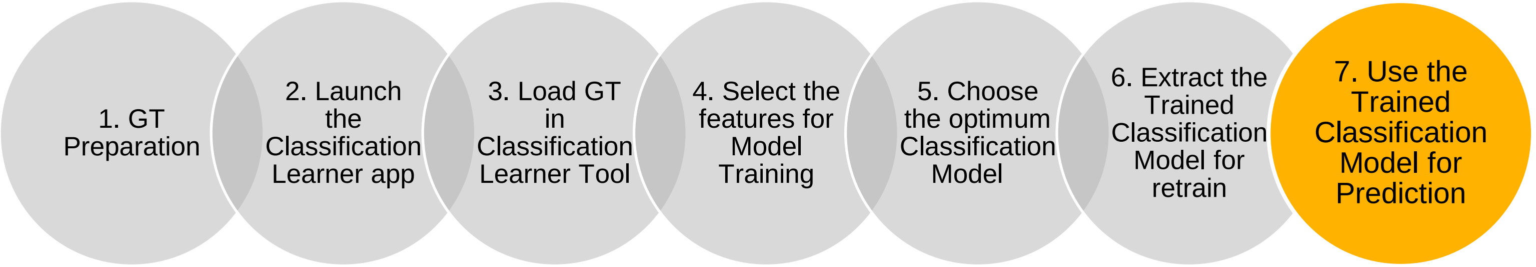
True class \ Predicted class	Car	Others
Car	2424	14
Others	20	566

Model number 1
Status: Trained
Accuracy: 98.9%
Prediction speed: ~110000 obs/sec
Training Time: 2.4096 sec

Classifier
Preset: Complex Tree

Original Dataset: **Combineddataset1** Observations: **3024** Predictors: 4 Response Variable: **Class** Response Classes: 2 Size of Dataset: **448 kB** Validation: **5-fold Cross-Validation**

USE THE TRAINED CLASSIFICATION MODEL FOR PREDICTION



1. GT Preparation

2. Launch the Classification Learner app

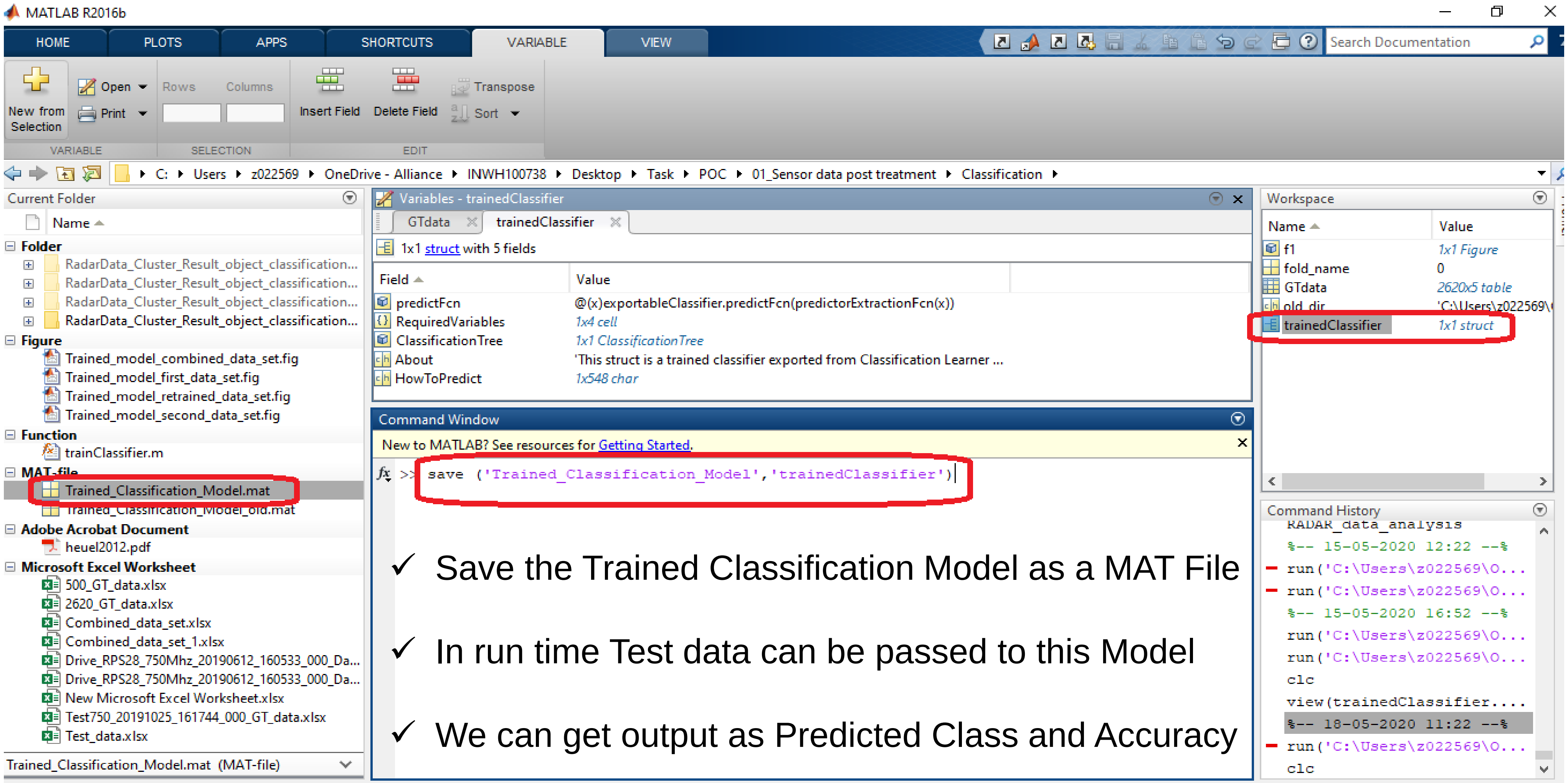
3. Load GT in Classification Learner Tool

4. Select the features for Model Training

5. Choose the optimum Classification Model

6. Extract the Trained Classification Model for retrain

7. Use the Trained Classification Model for Prediction



C: ▸ Users ▸ z022569 ▸ OneDrive - Alliance ▸ INWH100738 ▸ Desktop ▸ Task ▸ POC ▸ 01_Sensor data post treatment ▸ Classification ▸

Current Folder

Name

Folder
- RadarData_Cluster_Result_object_classification...
- RadarData_Cluster_Result_object_classification...
- RadarData_Cluster_Result_object_classification...
- RadarData_Cluster_Result_object_classification...

Figure
- Trained_model_combined_data_set.fig
- Trained_model_first_data_set.fig
- Trained_model_retrained_data_set.fig
- Trained_model_second_data_set.fig

Function
- trainClassifier.m

MAT-file
- Trained_Classification_Model.mat
- Trained_Classification_Model_old.mat

Adobe Acrobat Document
- heuel2012.pdf

Microsoft Excel Worksheet
- 500_GT_data.xlsx
- 2620_GT_data.xlsx
- Combined_data_set.xlsx
- Combined_data_set_1.xlsx
- Drive_RPS28_750Mhz_20190612_160533_000_Da...
- Drive_RPS28_750Mhz_20190612_160533_000_Da...
- New Microsoft Excel Worksheet.xlsx
- Test750_20191025_161744_000_GT_data.xlsx
- Test_data.xlsx

Trained_Classification_Model.mat (MAT-file)

Variables - trainedClassifier

GTdata | trainedClassifier

1x1 struct with 5 fields

Field	Value
predictFcn	@(x)exportableClassifier.predictFcn(predictorExtractionFcn(x))
RequiredVariables	1x4 cell
ClassificationTree	1x1 ClassificationTree
About	'This struct is a trained classifier exported from Classification Learner ...
HowToPredict	1x548 char

Workspace

Name	Value
f1	1x1 Figure
fold_name	0
GTdata	2620x5 table
old_dir	'C:\Users\z022569\(
trainedClassifier	1x1 struct

Command Window

New to MATLAB? See resources for Getting Started.

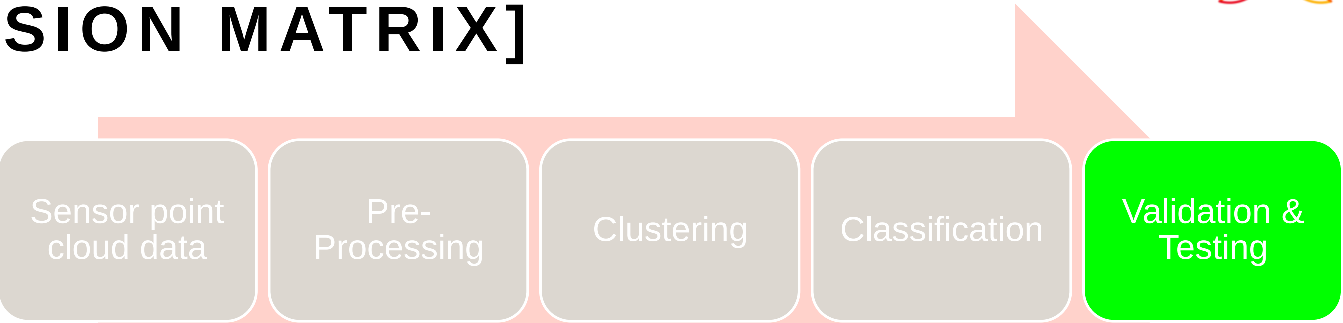
```
>> save ('Trained_Classification_Model','trainedClassifier')
```

Command History

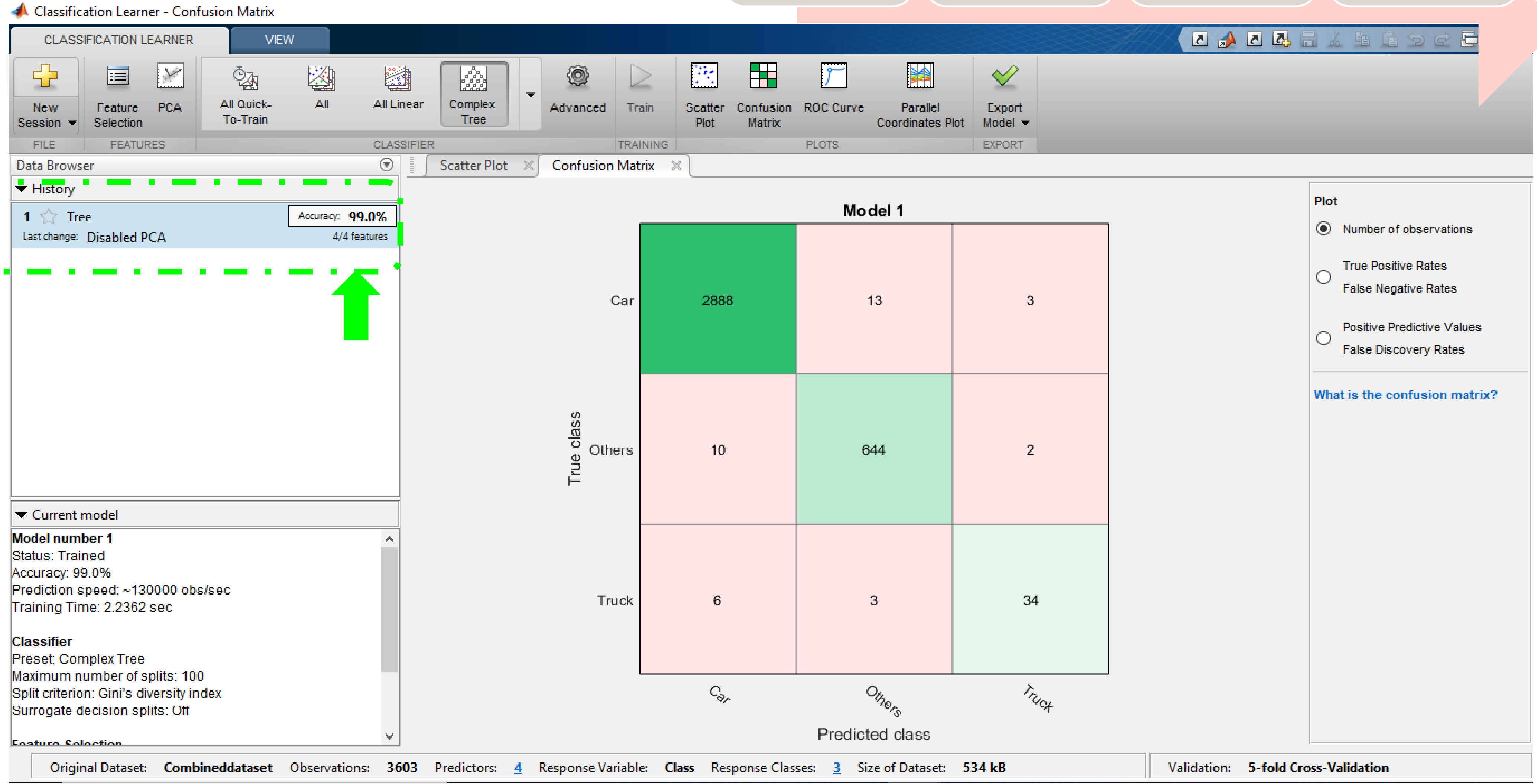
```
RADAR_data_analysis
%-- 15-05-2020 12:22 --%
run('C:\Users\z022569\O...
run('C:\Users\z022569\O...
%-- 15-05-2020 16:52 --%
run('C:\Users\z022569\O...
run('C:\Users\z022569\O...
clc
view(trainedClassifier....
%-- 18-05-2020 11:22 --%
run('C:\Users\z022569\O...
clc
```

- ✓ Save the Trained Classification Model as a MAT File
- ✓ In run time Test data can be passed to this Model
- ✓ We can get output as Predicted Class and Accuracy

VALIDATION & TESTING [CONFUSION MATRIX]



Sensor point cloud data → Pre-Processing → Clustering → Classification → Validation & Testing



1 Tree Accuracy: 99.0%
Last change: Disabled PCA 4/4 features

Current model

Model number 1
Status: Trained
Accuracy: 99.0%
Prediction speed: ~130000 obs/sec
Training Time: 2.2362 sec

Classifier
Preset: Complex Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off

Feature Selection

Model 1

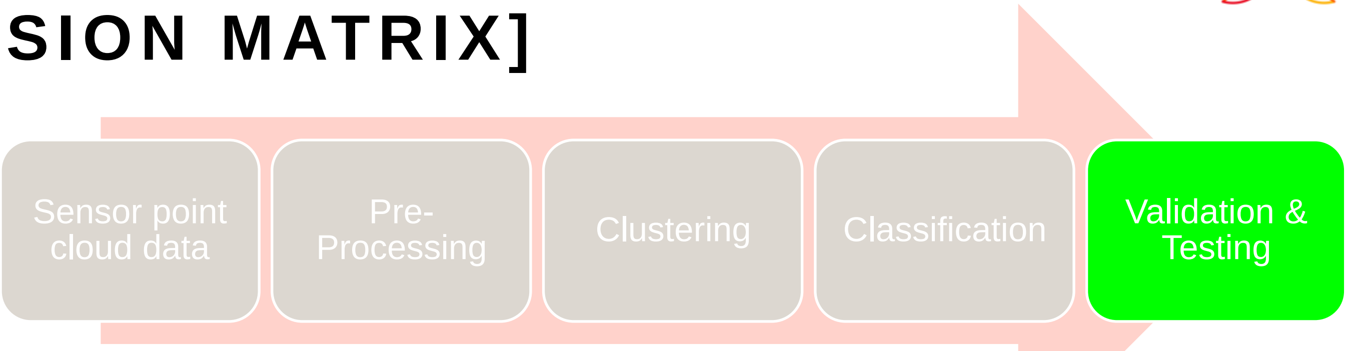
True class \ Predicted class	Car	Others	Truck
Car	2888	13	3
Others	10	644	2
Truck	6	3	34

Plot
● Number of observations
○ True Positive Rates False Negative Rates
○ Positive Predictive Values False Discovery Rates

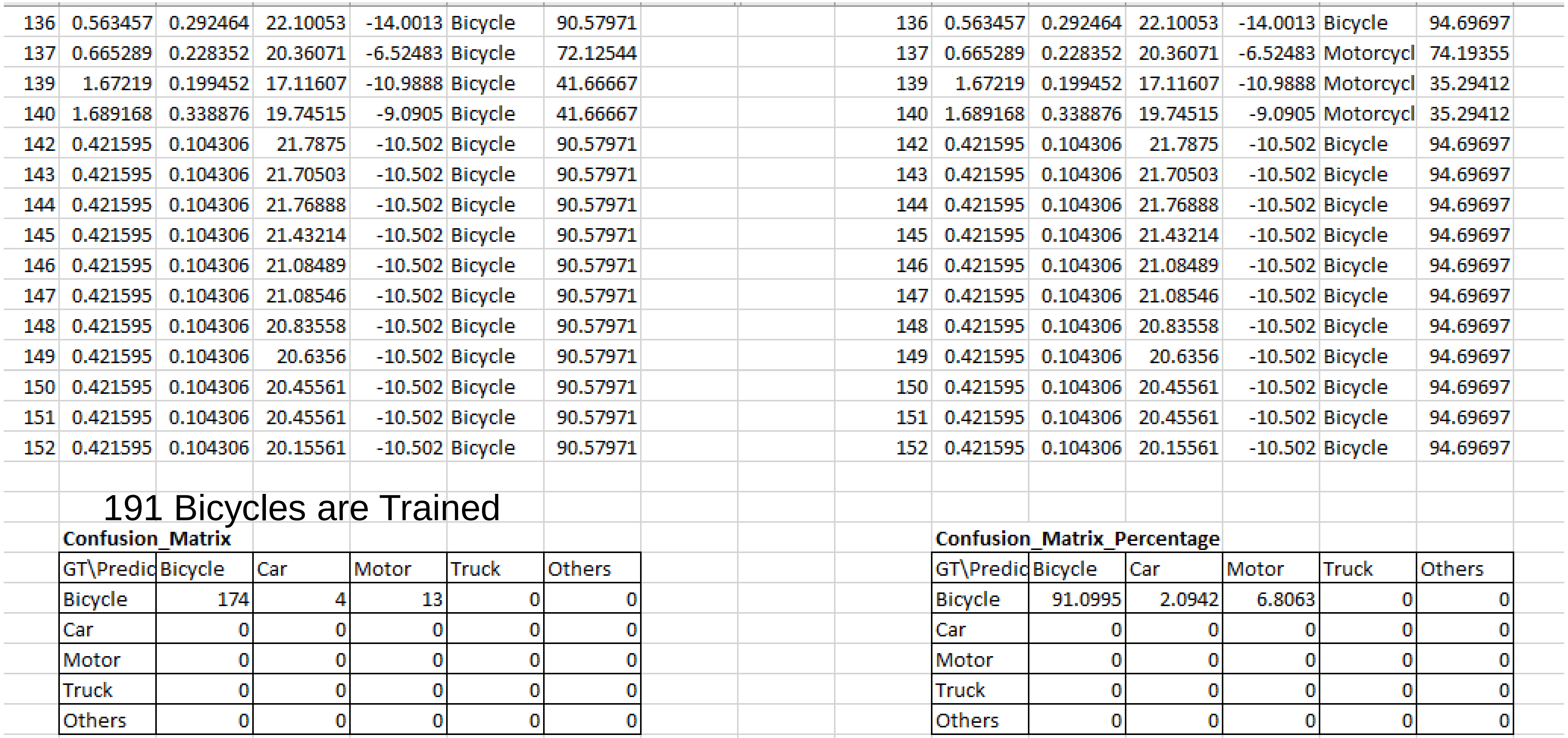
What is the confusion matrix?

Original Dataset: **Combineddataset** Observations: **3603** Predictors: 4 Response Variable: **Class** Response Classes: 3 Size of Dataset: **534 kB** Validation: **5-fold Cross-Validation**

VALIDATION & TESTING [CONFUSION MATRIX]


Sensor point cloud data
Pre-Processing
Clustering
Classification
Validation & Testing

- Bicycle Class Training and comparison with Confusion Matrix



136	0.563457	0.292464	22.10053	-14.0013	Bicycle	90.57971
137	0.665289	0.228352	20.36071	-6.52483	Bicycle	72.12544
139	1.67219	0.199452	17.11607	-10.9888	Bicycle	41.66667
140	1.689168	0.338876	19.74515	-9.0905	Bicycle	41.66667
142	0.421595	0.104306	21.7875	-10.502	Bicycle	90.57971
143	0.421595	0.104306	21.70503	-10.502	Bicycle	90.57971
144	0.421595	0.104306	21.76888	-10.502	Bicycle	90.57971
145	0.421595	0.104306	21.43214	-10.502	Bicycle	90.57971
146	0.421595	0.104306	21.08489	-10.502	Bicycle	90.57971
147	0.421595	0.104306	21.08546	-10.502	Bicycle	90.57971
148	0.421595	0.104306	20.83558	-10.502	Bicycle	90.57971
149	0.421595	0.104306	20.6356	-10.502	Bicycle	90.57971
150	0.421595	0.104306	20.45561	-10.502	Bicycle	90.57971
151	0.421595	0.104306	20.45561	-10.502	Bicycle	90.57971
152	0.421595	0.104306	20.15561	-10.502	Bicycle	90.57971

136	0.563457	0.292464	22.10053	-14.0013	Bicycle	94.69697
137	0.665289	0.228352	20.36071	-6.52483	Motorcycl	74.19355
139	1.67219	0.199452	17.11607	-10.9888	Motorcycl	35.29412
140	1.689168	0.338876	19.74515	-9.0905	Motorcycl	35.29412
142	0.421595	0.104306	21.7875	-10.502	Bicycle	94.69697
143	0.421595	0.104306	21.70503	-10.502	Bicycle	94.69697
144	0.421595	0.104306	21.76888	-10.502	Bicycle	94.69697
145	0.421595	0.104306	21.43214	-10.502	Bicycle	94.69697
146	0.421595	0.104306	21.08489	-10.502	Bicycle	94.69697
147	0.421595	0.104306	21.08546	-10.502	Bicycle	94.69697
148	0.421595	0.104306	20.83558	-10.502	Bicycle	94.69697
149	0.421595	0.104306	20.6356	-10.502	Bicycle	94.69697
150	0.421595	0.104306	20.45561	-10.502	Bicycle	94.69697
151	0.421595	0.104306	20.45561	-10.502	Bicycle	94.69697
152	0.421595	0.104306	20.15561	-10.502	Bicycle	94.69697

191 Bicycles are Trained

Confusion_Matrix

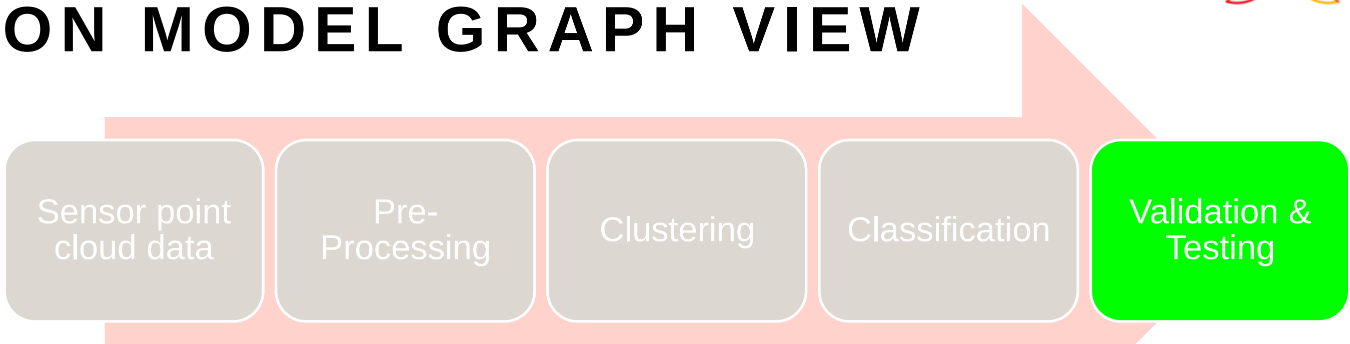
GT\Predic	Bicycle	Car	Motor	Truck	Others
Bicycle	174	4	13	0	0
Car	0	0	0	0	0
Motor	0	0	0	0	0
Truck	0	0	0	0	0
Others	0	0	0	0	0

Confusion_Matrix_Percentage

GT\Predic	Bicycle	Car	Motor	Truck	Others
Bicycle	91.0995	2.0942	6.8063	0	0
Car	0	0	0	0	0
Motor	0	0	0	0	0
Truck	0	0	0	0	0
Others	0	0	0	0	0

DECISION TREE - CLASSIFICATION MODEL GRAPH VIEW

- ✓ Selected as “Complex tree” model
- ✓ Initial level classification on “CAR” & Others



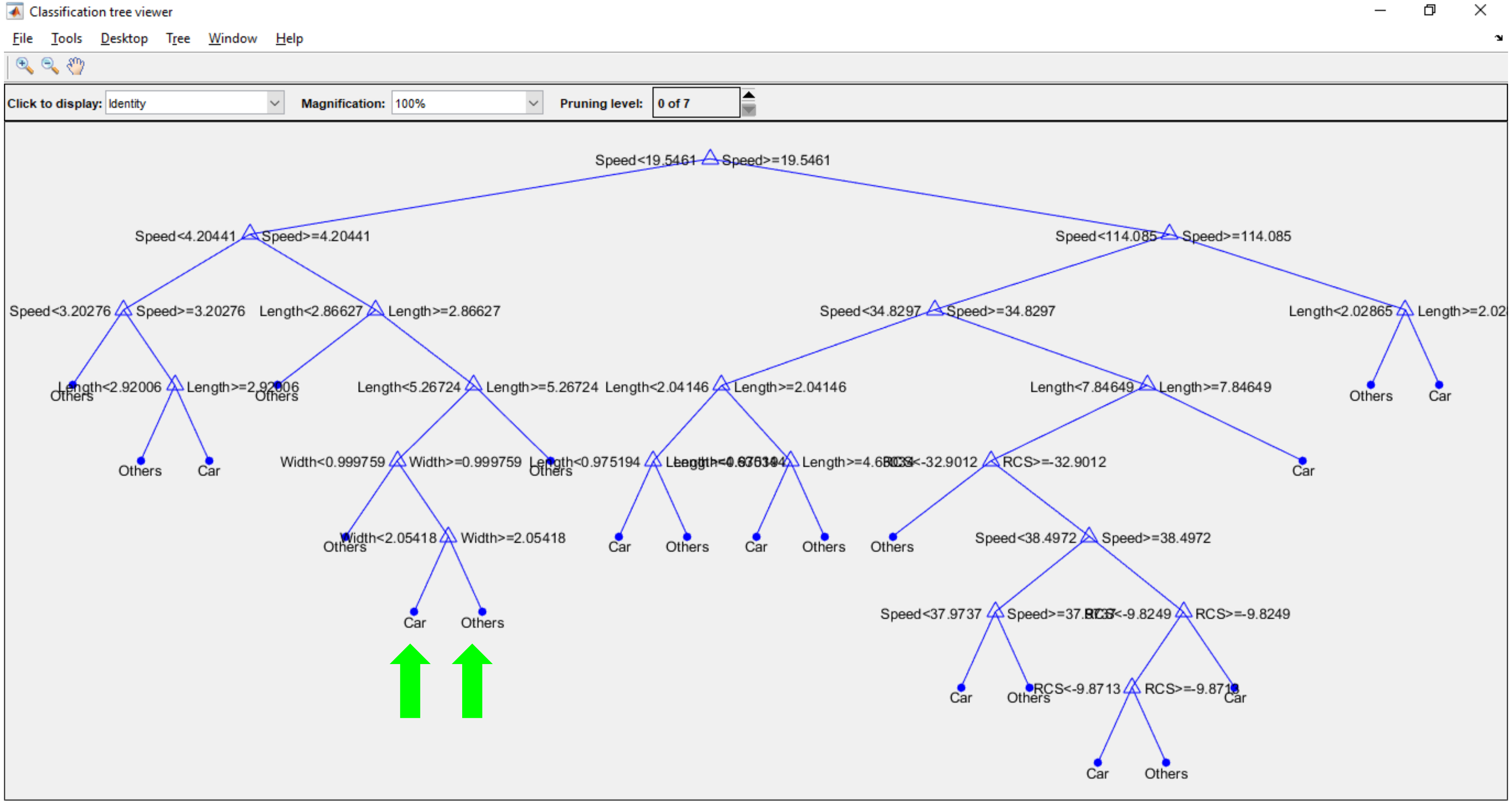
Sensor point cloud data

Pre-Processing

Clustering

Classification

Validation & Testing



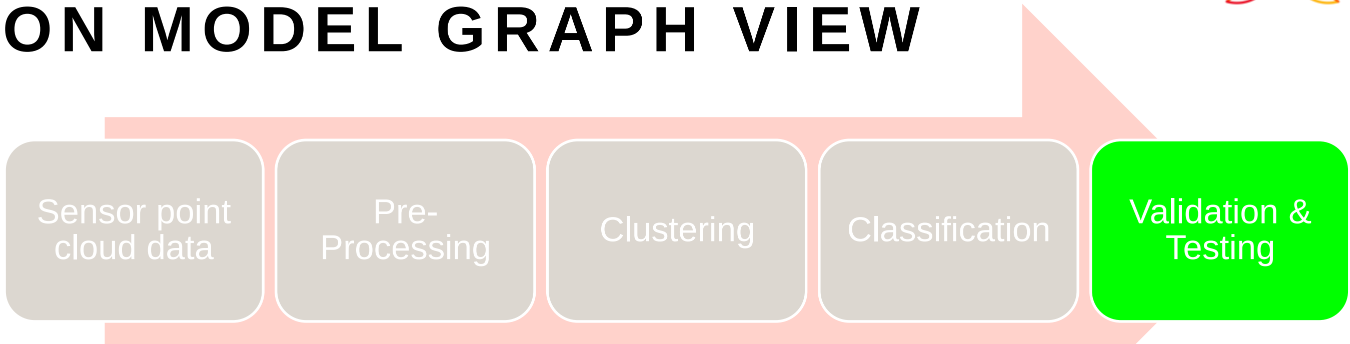
Click to display: Identity

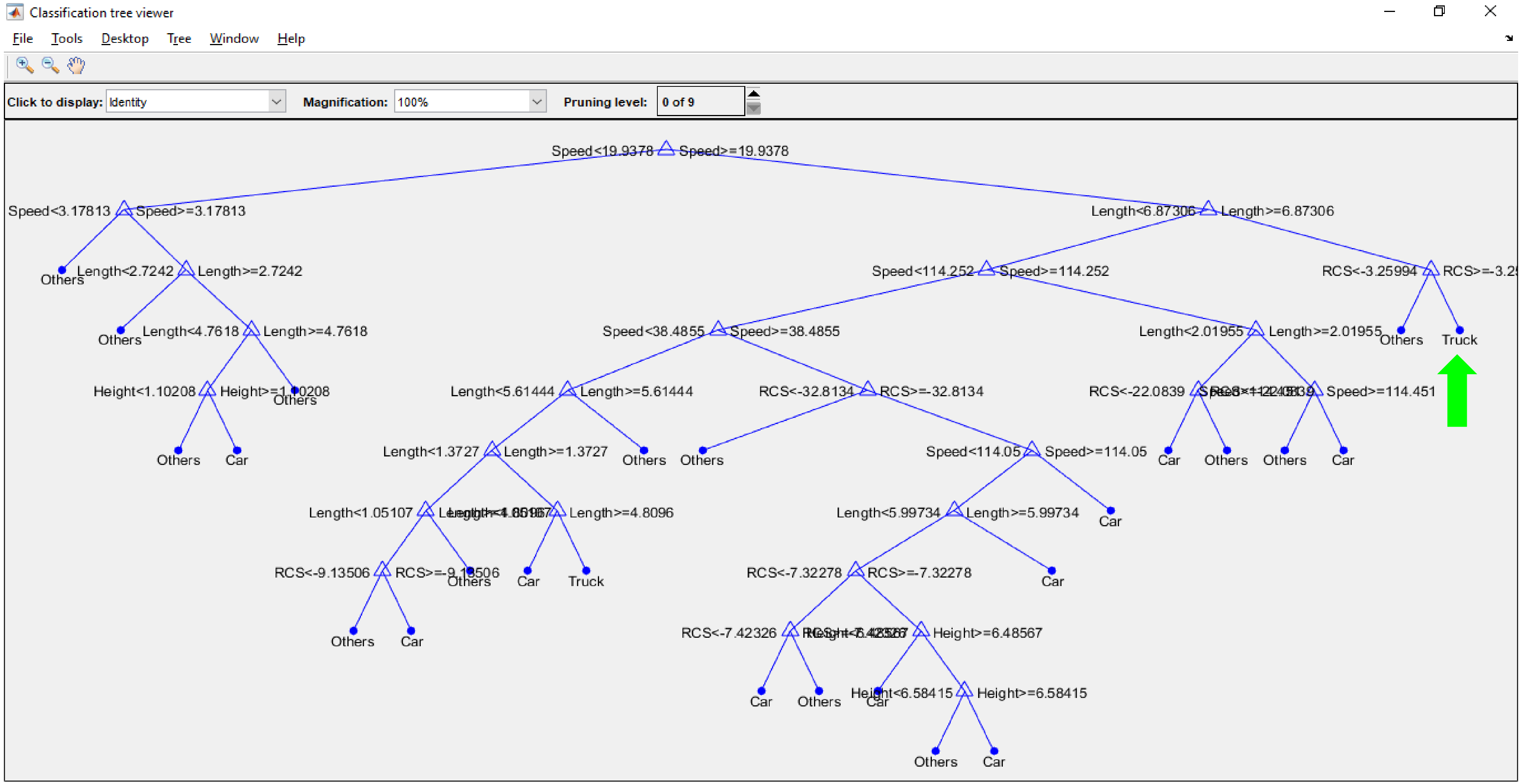
Magnification: 100%

Pruning level: 0 of 7

DECISION TREE - CLASSIFICATION MODEL GRAPH VIEW

- ✓ Selected as “Complex tree” model
- ✓ Next level: Car, Truck and others




Classification tree viewer

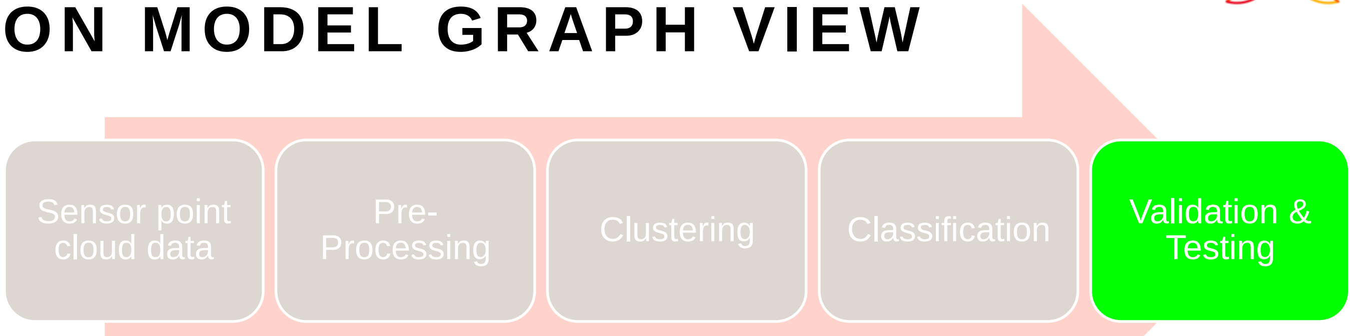
Click to display: Identity

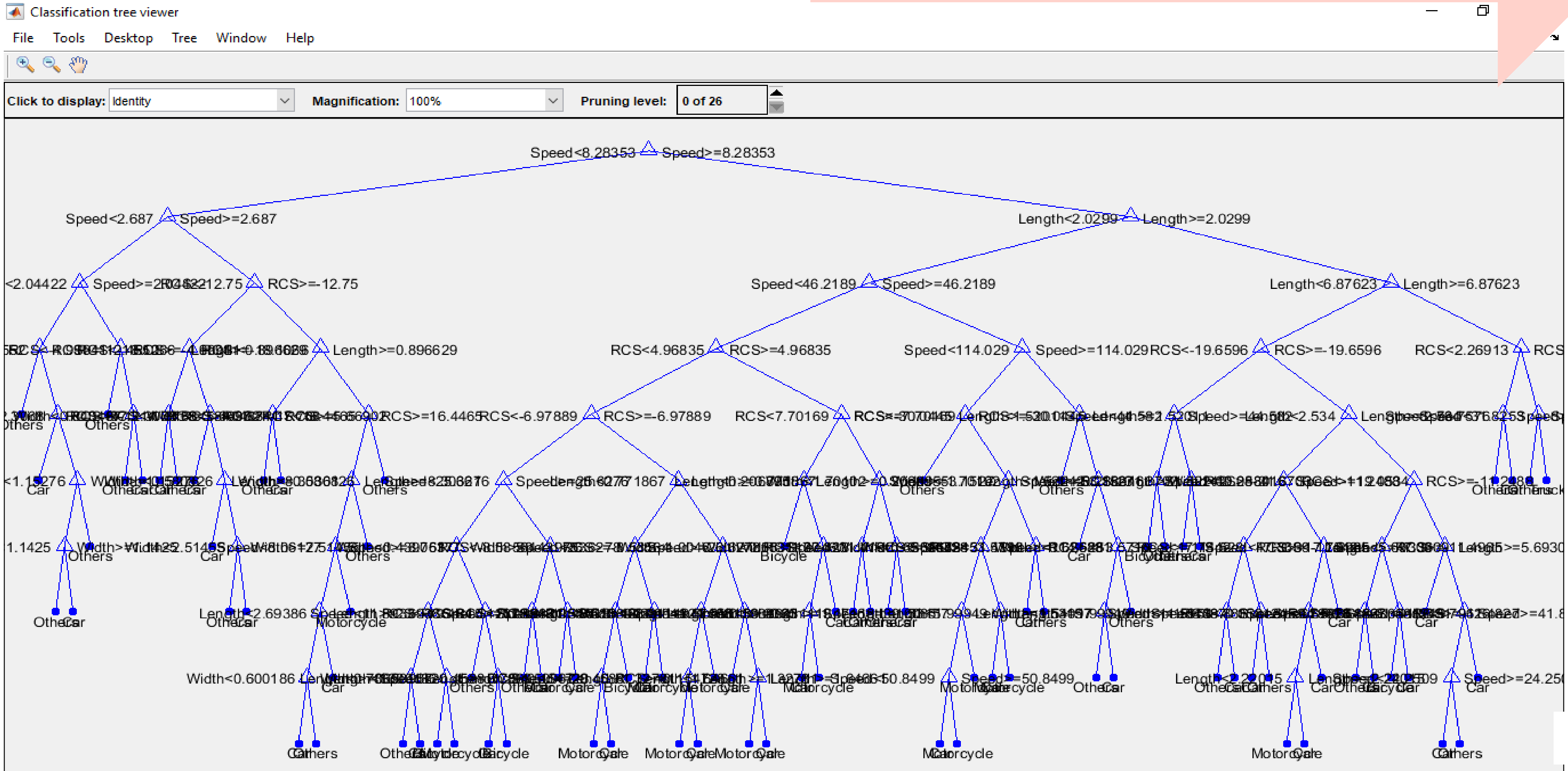
Magnification: 100%

Pruning level: 0 of 9

DECISION TREE - CLASSIFICATION MODEL GRAPH VIEW

- ✓ Selected as “Complex tree” model
- ✓ Final step: Car, Truck, Powered two wheeler, Bicycle, Pedestrians and others




Classification tree viewer

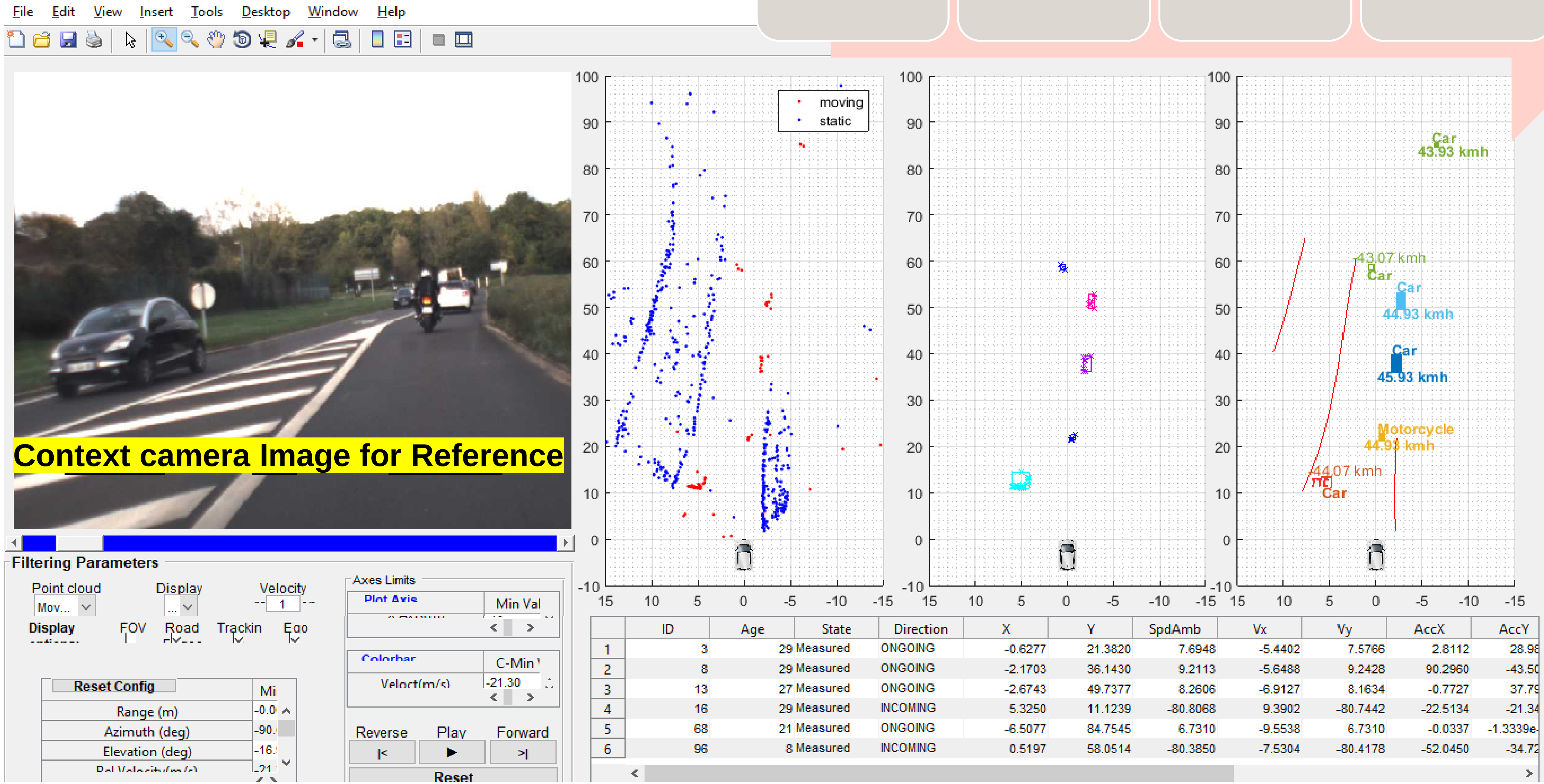
Click to display: Identity

Magnification: 100%

Pruning level: 0 of 26

EXPERIMENTAL RESULTS

Sensor point cloud data → Pre-Processing → Clustering → Classification → Validation & Testing



Context camera Image for Reference

Filtering Parameters

Point cloud: Mov...

Display: ...

Velocity: 1

Display options: FOV, Road, Trackin, Ego

Reset Config	Mi
Range (m)	-0.0
Azimuth (deg)	-90.
Elevation (deg)	-16.
Rel Velocity(m/s)	-21.

Axes Limits

Plot Axis	Min Val

Colorbar	C-Min
Veloct(m/s)	-21.30

	ID	Age	State	Direction	X	Y	SpdAmb	Vx	Vy	AccX	AccY
1	3	29	Measured	ONGOING	-0.6277	21.3820	7.6948	-5.4402	7.5766	2.8112	28.98
2	8	29	Measured	ONGOING	-2.1703	36.1430	9.2113	-5.6488	9.2428	90.2960	-43.50
3	13	27	Measured	ONGOING	-2.6743	49.7377	8.2606	-6.9127	8.1634	-0.7727	37.79
4	16	29	Measured	INCOMING	5.3250	11.1239	-80.8068	9.3902	-80.7442	-22.5134	-21.34
5	68	21	Measured	ONGOING	-6.5077	84.7545	6.7310	-9.5538	6.7310	-0.0337	-1.3339e-
6	96	8	Measured	INCOMING	0.5197	58.0514	-80.3850	-7.5304	-80.4178	-52.0450	-34.72

SUMMARY

- Added value by using Statistical & Machine learning tool box
- Reduced manual effort by 80%
- Achieved Accuracy 95% on real time data (for primary classes)
- Fast execution in training & testing phase

Thanks for your time and attention!!! ☺

Acknowledgements:

SOURIA Charaf-Eddine	- Renault	GOMATAM Srinivasan	- RNTBCI
GRANDHI Prithvi-Ram	- RNTBCI	Rishu Gupta	- MathWorks
Shobhit Shanker	- MathWorks	Rashmi Gopala Rao	- MathWorks

Comments & Feedback can be directed to:

Suresh Selvam	- suresh.selvam@rntbci.com
Naresh Babu N T	- naresh-babu.napa-Thulasiayya@rntbci.com