



Volkswagen Data Lab

Connected Car – Fahrererkennung mit MATLAB

K-SI/LD1 | Julia Fumbarev | München, 27.06.2017

Mega-Trend: Fahrzeugvernetzung

Herausforderungen für die OEMs:

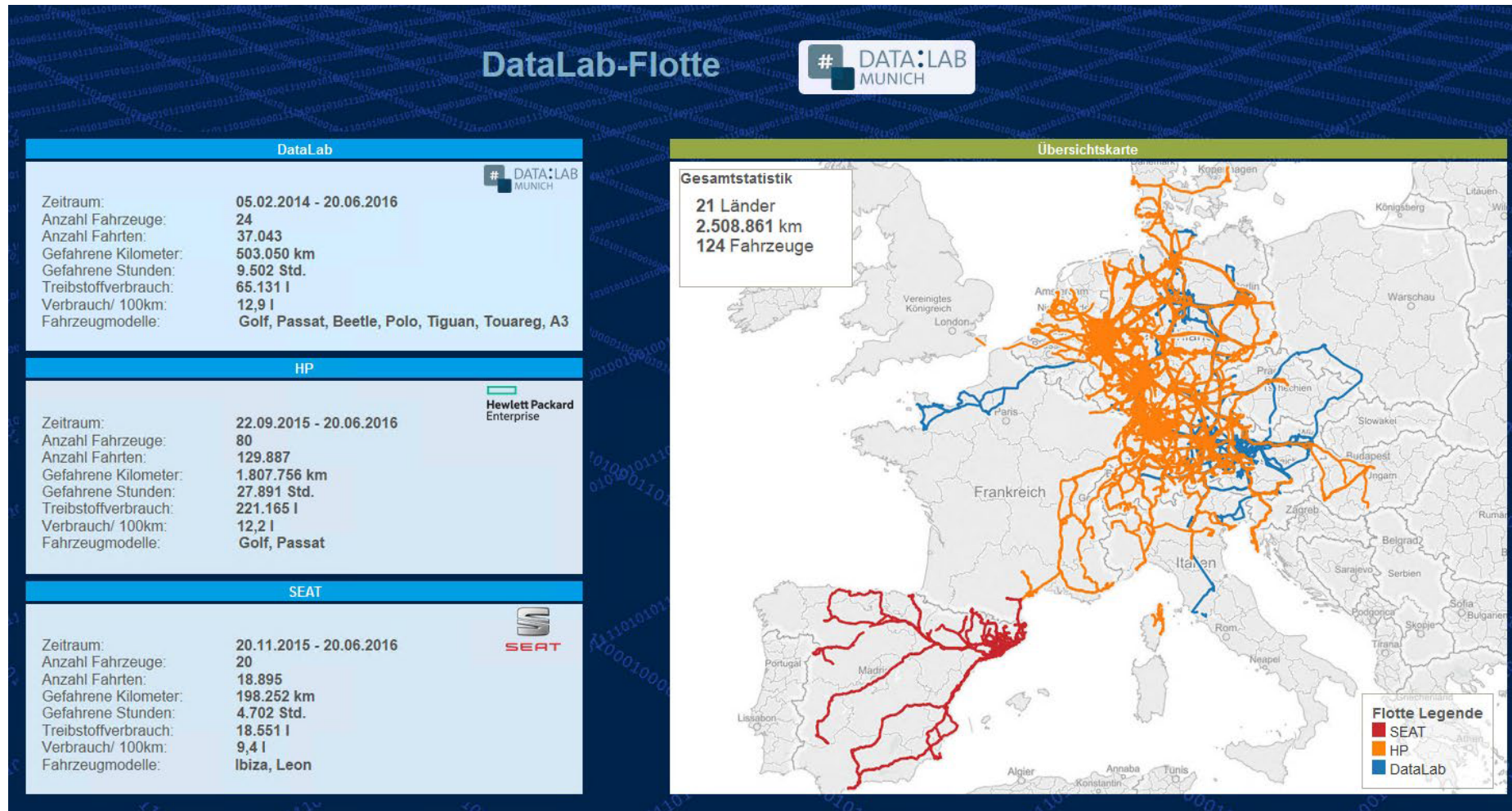
- ▶ Synchronisierung unterschiedlicher Lebenszyklen
- ▶ Integration der Endgeräte und der Anwendungen
- ▶ Sicherstellung der Konnektivität

Wünsche der Kunden zum vernetzten Fahrzeug:

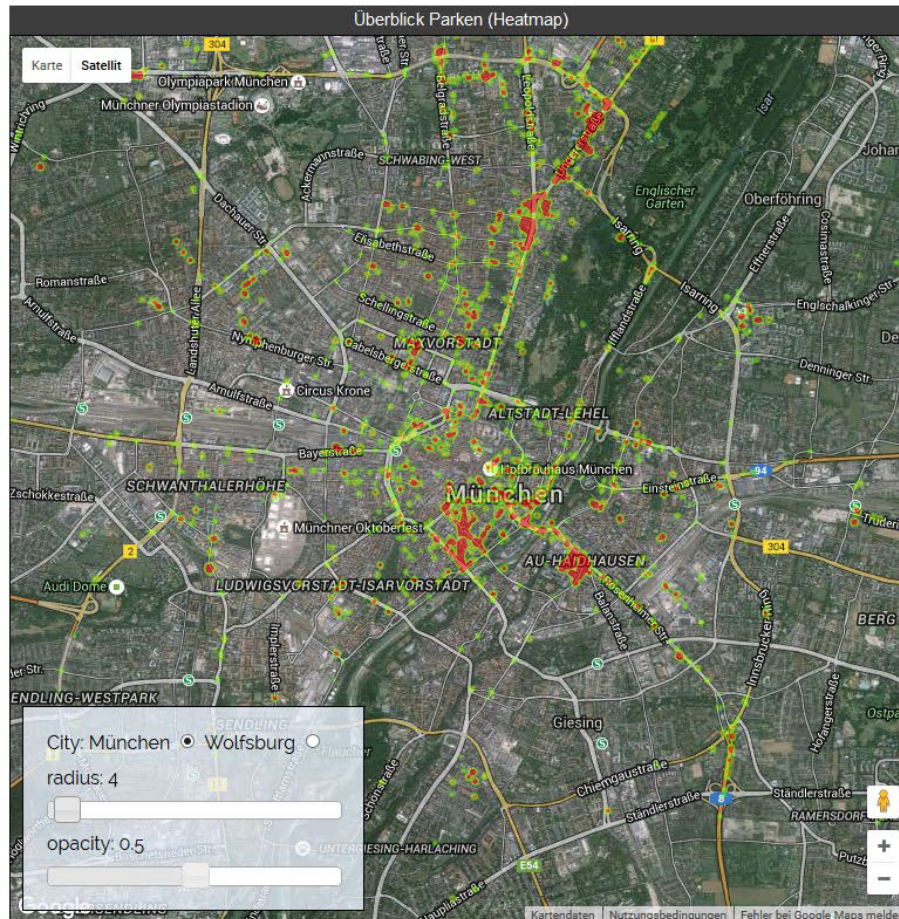
- ▶ Informationen zu Staus
- ▶ Informationen zu freien Parkplätzen
- ▶ **Personalisierte Services**

Flotteübersicht

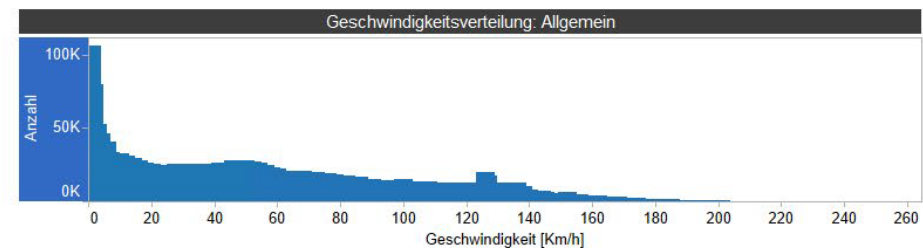
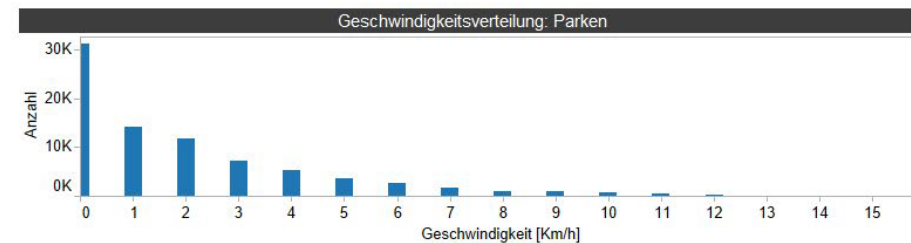
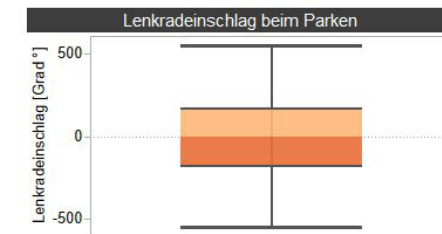
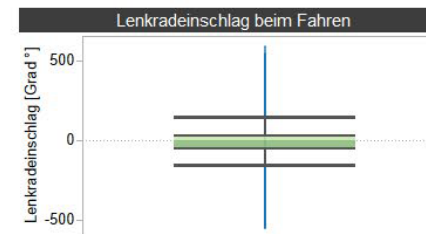
2,5 Jahre / 124 Fzg / > 2,5 Mio. km in Europa (21 Länder)



Beispiel Parkplatz-Erkennung



Identifikation von Parkmanövern durch Mustererkennung: Geschwindigkeit und Lenkradeinschlag



Data Science – Fahrererkennung durch Fahrverhalten

Ziel: Identifizierung einzelner Fahrer aus ihrem Fahrverhalten

Motivation: Verbesserung der Fahrer- und Flottensicherheit, gezieltes Fahrer-Coaching, automatisiertes Fahrtenbuch, Fahrer-individuelle Versicherungen etc.

Herausforderungen:

- ▶ **Genauigkeit:** Klassifizierung vieler Fahrer trotz geringer Trainingsdaten
- ▶ **Robustheit:** Unabhängigkeit von Umfeldbedingungen
- ▶ **Rechenzeit:** Vor allem wichtig für eingebettete Systeme

Setup:

- ▶ Geloggte CAN-Bus Daten
- ▶ Fahrtenbuch als Ground-Truth
- ▶ Filterung nach Autobahnfahrten

Data Science – Fahrererkennung durch Fahrverhalten

Wie wurde der „telematic fingerprint“ berechnet?

1. Daten extrahieren

2. Features bestimmen

3. Classifier Design

4. Auswertung

5. Visualisierung

- Erstellung eines Datenextrakts, welches die Daten von sieben Fahrern mit jeweils mehreren Fahrten beinhaltet
- Extraktion von Feldern, die sich auf das Fahrverhalten beziehen und möglichst unabhängig von der Umfeldsituation sind

```
d = data;  
% columns: driverId, driverTripId, vel, stwa, t_diff, acc
```

Data Science – Fahrererkennung durch Fahrverhalten

Wie wurde der „telematic fingerprint“ berechnet?

1. Daten extrahieren

2. Features bestimmen

3. Classifier Design

4. Auswertung

5. Visualisierung

- Berechnung von Features, die die Dynamik der Fahrweise abbilden

```
% calculate features
```

```
d(:, end +1) = d(:,3).*d(:,4); % vel x swta  
d(:, end +1) = d(:,3).*d(:,6); % vel x acc  
d(:, end +1) = [0; diff(d(:,6))]./d(:,5); % da/dt
```

- Bestimmung von unterschiedlichen statistischen Maßen mit Hilfe der Sliding-Window-Methode basierend auf Sequenzen von 25 Samples
- Sukzessive Selektion der Features
 - Mittelwert, 80%-Quantil, Standardabweichung und Minimalwert

Data Science – Fahrererkennung durch Fahrverhalten

Wie wurde der „telematic fingerprint“ berechnet?

1. Daten extrahieren

2. Features bestimmen

3. Classifier Design

4. Auswertung

5. Visualisierung

- Teilen des Datensatzes in ein Trainings- und Testset
- Bestimmung der besten Methode (basierend auf Accuracy) mit Hilfe Classification Learner App

```
[d_train, d_test] = splitData(d, driver);
```

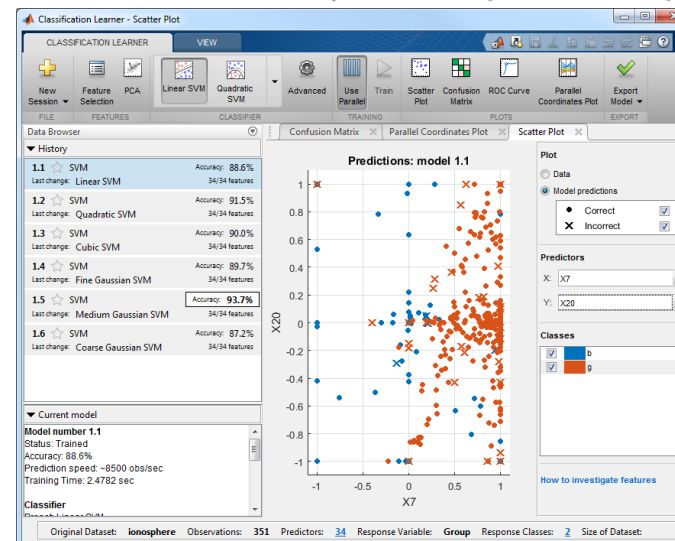
```
ws = 25;
```

```
[featMatTrain] = calculateFeatMat(d_train, driver, ws);
```

```
[featMatTest] = calculateFeatMat(d_test, driver, ws);
```

```
[trainedClassifier, validationAccuracy] = trainClassifier(featMatTrain);
```

```
yHat = trainedClassifier.predictFcn(featMatTest(:,1:end-1));
```



Data Science – Fahrererkennung durch Fahrverhalten

Wie wurde der „telematic fingerprint“ berechnet?

1. Daten extrahieren

2. Features bestimmen

3. Classifier Design

4. Auswertung

5. Visualisierung

■ Metriken zur Auswertung: Precision, Recall, f-Score

```
% stats.precision = TP / (TP + FP)
% stats.recall = TP / (TP + FN)
```

```
stats = confusionmatStats(featMatTest(:,end),yHat);
```

```
precision = sum(stats.precision)/length(driver)
recall = sum(stats.recall)/length(driver)
fScore = sum(stats.Fscore)/length(driver)
```

```
precision =
```

```
0.6944
```

```
recall =
```

```
0.5905
```

```
fScore =
```

```
0.5803
```

Data Science – Fahrererkennung durch Fahrverhalten

Wie wurde der „telematic fingerprint“ berechnet?

1. Daten extrahieren

2. Features bestimmen

3. Classifier Design

4. Auswertung

5. Visualisierung

- Analyse der Klassifizierungsergebnisse mittels Konfusionsmatrix
- Interaktive Visualisierung mittels Tableau

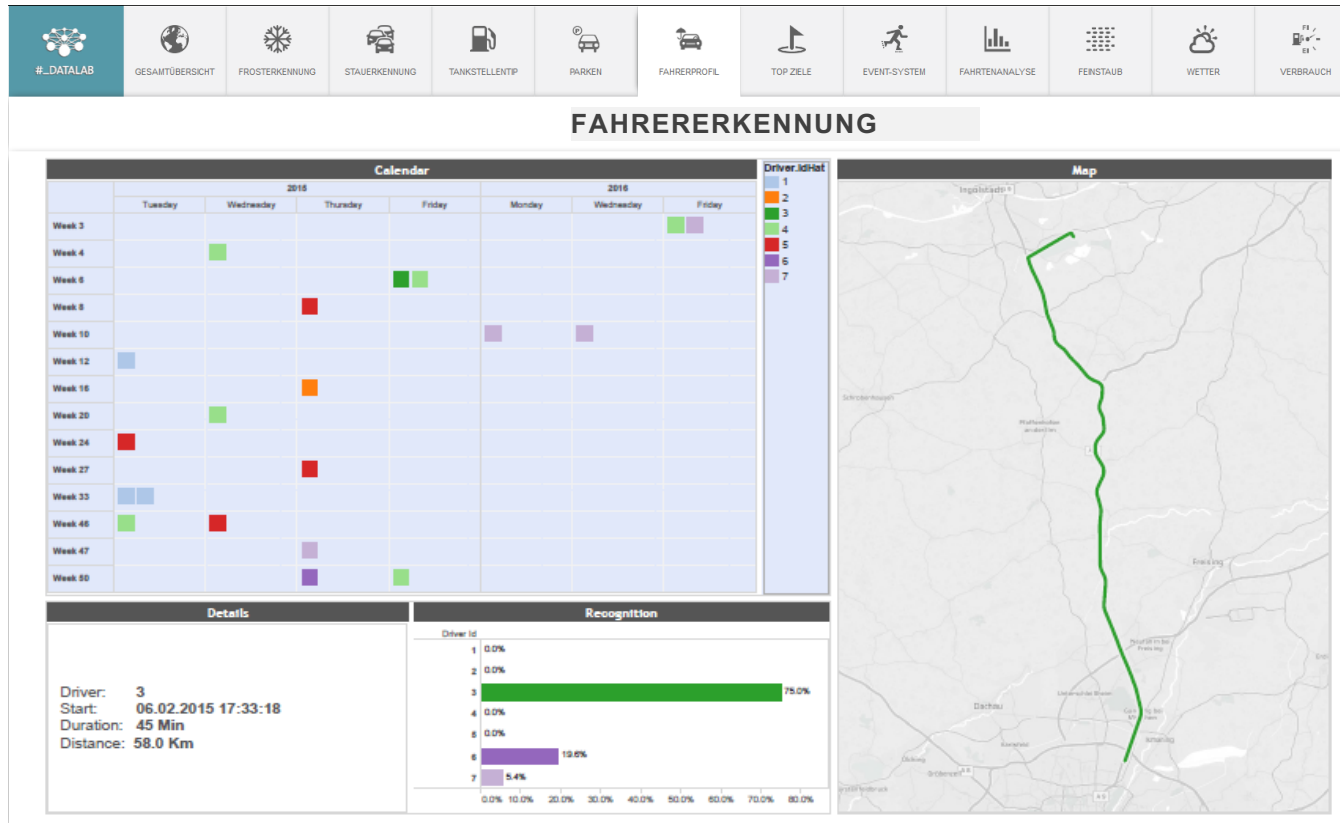
```
C = confusionmat(featMatTest(:,end),yHat);  
C = C./repmat(sum(C,2),1,size(C,2));
```

```
confMat(C,driver)
```

1	0.73	0.09	0.00	0.09	0.00	0.00	0.09
2	0.00	0.13	0.00	0.50	0.25	0.13	0.00
3	0.00	0.00	0.60	0.10	0.00	0.30	0.00
4	0.04	0.08	0.04	0.42	0.13	0.17	0.13
5	0.08	0.23	0.00	0.23	0.38	0.00	0.08
6	0.00	0.00	0.00	0.18	0.09	0.36	0.36
7	0.00	0.00	0.04	0.20	0.08	0.20	0.48
	1	2	3	4	5	6	7

Data Science – Fahrererkennung durch Fahrverhalten

Automatisierte Fahrererkennung durch Maschinelles Lernen



- Berechnung von Features basierend auf geloggtten CAN-Bus Daten, die das Fahrerverhalten charakterisieren
- Lernen individueller Fahrermodelle („**telematic fingerprint**“)
- Statistische Erkennung des Fahrers
- Ermöglicht ein **automatisiertes Fahrtenbuch**
- Grundlage für das „**pay-as-you-drive**“ Konzept
- PoC, noch keine Integration