MACHINE LEARNING AS A EXAMPLE FOR OVERRIDE DETECTION

SUPERVISED MACHINE LEARNING WITH MATLAB.
MOTIVATION.

Evaluation of the "Statistics and Machine Learning" Toolbox from MATLAB

Large number of recorded vehicle measurements (unlabeled) available
MACHINE LEARNING.

Algorithms used.

Machine Learning

- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
  - Clustering

Support Vector Machines
- Discriminant Analysis
- Neural Networks
- Nearest Neighbor
- Decision Trees
WORKFLOW.

1. Record the measurements
2. Loading the data
3. Feature extraction
4. Training the model
5. Application of the model to test data set
6. Validation of prediction
GENERATION OF THE MEASURED DATA. TRACK

Train a model

- Training data set:
  - Handling course Miramas
  - 259,000 data points
  - ≈ 43 minutes

Test the trained model

- Test data set:
  - Handling course Aschheim
  - 150,000 data points
  - ≈ 25 minutes
GENERATION OF THE MEASURED DATA.

Insert a trigger signal.

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Signal 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Signal 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Signal 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Signal 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Override
GENERATION OF THE MEASURED DATA.

Insert a trigger signal.

Signal 1
Signal 2
Signal 3
Signal 4
FEATURE EXTRACTION.

Filter.

Suppress signal noise
Training and test data sets have to filter in the same way
FEATURE EXTRACTION.

Peak Analysis.

- Use the FindPeaks function
- Minimum distance between the peaks
FEATURE EXTRACTION.

Principal Component Analysis (PCA).

Transformation in the directions of Principal Components
MODEL SELECTION.

K-Fold Crossvalidation.

Source: Machine Learning for Evolution Strategies, Kramer, 2016, S.39
MODEL SELECTION.

Confusion matrix.

Goal: 100% on the green diagonal

Receiver-Operating-Characteristic-Curve.

Goal: $AUC = 1$
VIDEO: PROCEDURE.

```matlab
function [daten] = Daten_laden_zusammenfuegte_Daten('C:\Nico\Desktop\brauchbare_Messungen\zusammenfuegte_Daten\daten=[Daten_zusammenfuegte_Daten, Miranas_gut_gefiltert, Miranas_komplett, Miranas_komplett].

% training data
trainingsdaten = load('Daten_laden_zusammenfuegte_Daten(trainingdaten)

% training data load
trainingsdaten = load('Daten_laden_zusammenfuegte_Daten

trainingsset = trainingset, neuesdatenset;

% Alle relevanten Signale in eine Matrix zusammenlegen

daten = arraywin(Daten);

daten.Properties.VariableNames = {'Radwinkel', 'Gierrate', 'Laengenbeschleunigung', 'Querbeschleunigung', 'Trigger'};

end
```
RESULTS.

confusion matrix: K-Nearest Neighbor & PCA Feature Extraktion

Performance of model:

<table>
<thead>
<tr>
<th>Predicted</th>
<th>True Class 0</th>
<th>True Class 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted 0</td>
<td>90.4% (82943)</td>
<td>9.6% (8856)</td>
</tr>
<tr>
<td>Actual 0</td>
<td>90.35% (82943)</td>
<td>9.65% (8856)</td>
</tr>
<tr>
<td>Actual 1</td>
<td>5.26% (3162)</td>
<td>94.74% (57004)</td>
</tr>
</tbody>
</table>
RESULTS.

confusion matrix: Support Vector Machine

![Confusion Matrix for SVM](image)

Performance of model:

<table>
<thead>
<tr>
<th></th>
<th>Predicted 0</th>
<th>Predicted 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual 0</td>
<td>73.07% (67078)</td>
<td>26.93% (24721)</td>
</tr>
<tr>
<td>Actual 1</td>
<td>1.08% (650)</td>
<td>98.92% (59516)</td>
</tr>
</tbody>
</table>

![ROC Curve](image)
RESULTS.

confusion matrix: Quadratic Discriminant analysis model

Performance of model:

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted 0</th>
<th>Predicted 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual 0</td>
<td>82.73% (75946)</td>
<td>17.27% (15853)</td>
</tr>
<tr>
<td>Actual 1</td>
<td>1.17% (701)</td>
<td>98.83% (59465)</td>
</tr>
</tbody>
</table>
RESULTS.

confusion matrix: Complex Decision Trees

Performance of model:

<table>
<thead>
<tr>
<th></th>
<th>Predicted 0</th>
<th>Predicted 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual 0</td>
<td>95.86% (87996)</td>
<td>4.14% (3803)</td>
</tr>
<tr>
<td>Actual 1</td>
<td>1.84% (1109)</td>
<td>98.16% (59057)</td>
</tr>
</tbody>
</table>

ROC Curve
Learning must always be carried out from the beginning of the measurements, no adaptive learning

Generate C code from the learned algorithm possible

Fast results with little previous knowledge