Do I have a seat?

Predicting PT occupancy

A supervised learning approach for imbalanced data classification

Léonie Heydenrijk-Ottens, Viktoriya Degeler, Ding Luo, Niels van Oort, Hans van Lint

Brisbane, July 25, 2018

TU Delft

Picture: DvhN/Jan Bouwman

Problem

- Smart Card Data = knowledge However
- Not *real time* available

Question

- How well can we predict the regular occupancy, based on historical data?

TU Delft

Self introduction

Background

- PhD candidate (2017-2021)
  Transport and Planning, TU Delft
- Before: Data scientist consultant
  CGI
- MSc Applied Mathematics
  Constrained optimal control of nonlinear systems, TNO, VU University Amsterdam

Delft integrated Traffic & Travel Laboratory

Smart Public Transport Lab

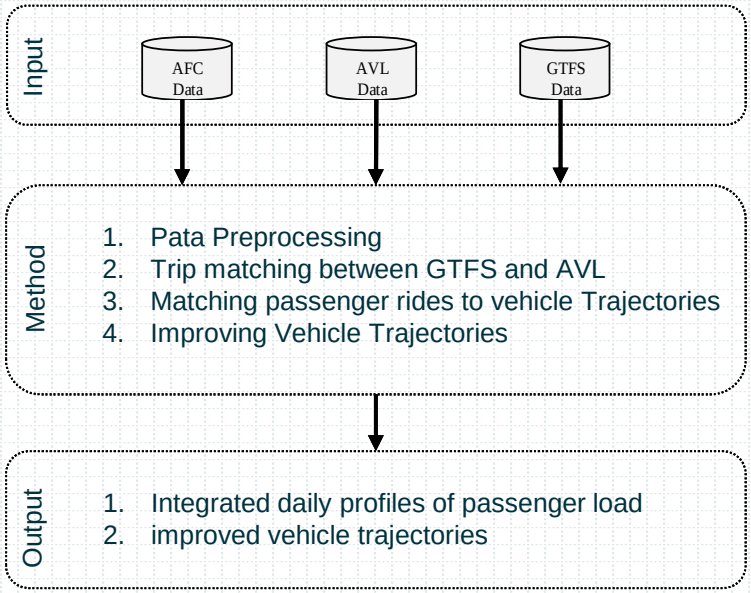
Project: My TRAvel Companion

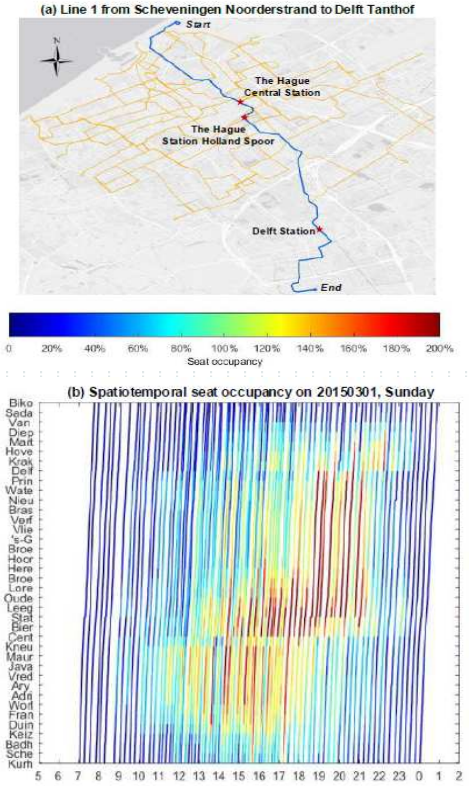
Predicting passenger load – Why?

- Do I have a seat?
- (Especially at night) Are there other people, so I feel save?

Data: Spatiotemporal Load Profiles of the Hague

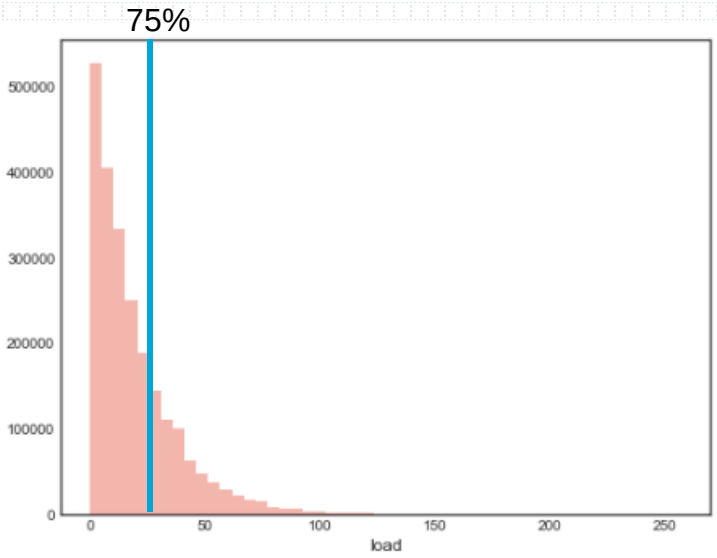
March 2015

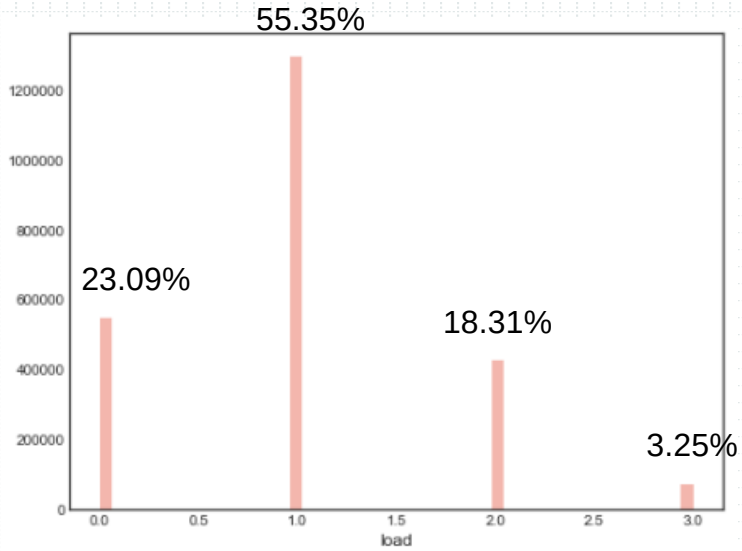




Source: Luo, D., (2018). Constructing Spatiotemporal Load Profiles of Transit Vehicles with Multiple Data Sources (No. 18-02399). Acknowledgement: HTM and Stichting OpenGeo provided the AFC and AVL datasets, resp.

Passenger load – March 2015

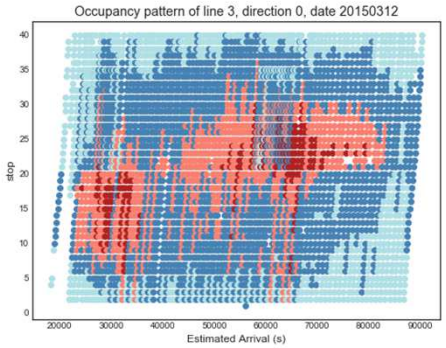


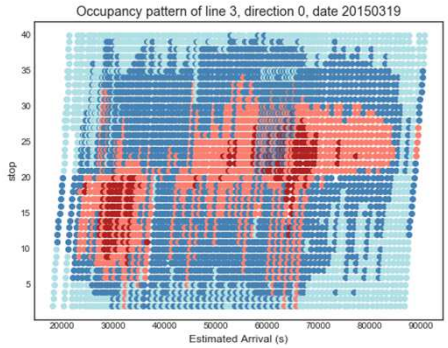


- Passenger load, all lines
- Passenger load classes
  - 0: < 10% seat occupancy
  - 1: < 50% seat occupancy
  - 2: < 100% seat occupancy
  - 3: ≥ 100% seat occupancy

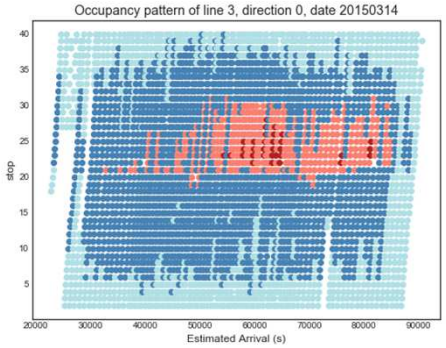
Occupancy patterns

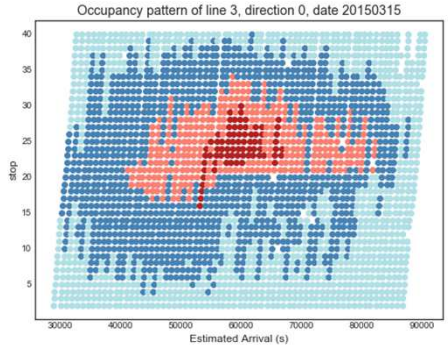
- Workday:





- Saterday/Sunday:





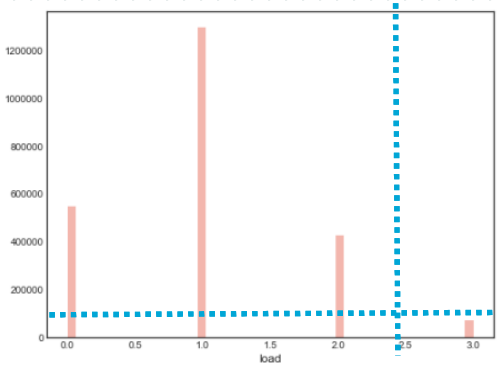
Legend:

- Almost empty
- Sit alone
- Sit next to someone
- All seats occupied

Method - classification

- Classification models
  - Random forest, gradient boosting, k-nearest neighbor, multi-layer perceptron
- Undersample majority classes
  - Randomly remove majority class observations
  - Helps balance the dataset
  - Risk of deleting important information
- Oversample majority classes
  - Synthetic Minority Over-sampling Technique (SMOTE)
  - Creates synthetic observations of the minority class by:
    - Finding the k-nearest-neighbors for minority class observations (finding similar observations)
    - Randomly choosing one of the k-nearest-neighbors and using it to create a similar, but randomly modificated, new observation.
  - No information loss
  - Risk of overfitting

Load classes



Classes:
- Class 0: Almost empty
- Class 1: Sit alone
- Class 2: Sit next to someone
- Class 3: All seats occupied

Method - classification

- Sequential classifier
  - Step 1. ‘translate’ dataset - very under represented class versus rest (one-vs-all):
    - Class 0’: not all seats occupied
    - Class 1’: all seats occupied
  - Step 2. Either undersample class 0’ or oversample class 1’.
  - Step 3. Use this data to train a model M1 to predict class 3
  - Step 4. Train a model M2 to predict the ‘type’ of seat – do not use data from class 3
  - Step 5. M1 overrules M2

- Classes:
  - Class 0: Almost empty
  - Class 1: Sit alone
  - Class 2: Sit next to someone
  - Class 3: All seats occupied

TU Delft

Method - classification

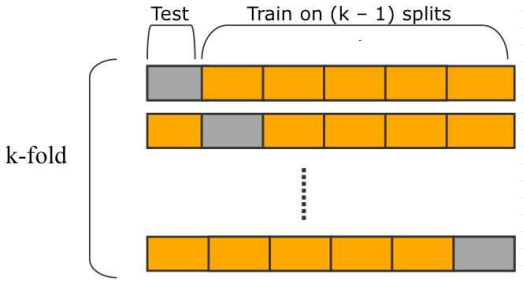
- One-versus-all sequential classifier
  - Train a model for each class in a one-vs-all way
  - Let each model predict
  - Prediction overruling: (class) 3 > 0 > 2 > 1
- Cost sensitive classification with sampling

- Classes:
  - Class 0: Almost empty
  - Class 1: Sit alone
  - Class 2: Sit next to someone
  - Class 3: All seats occupied

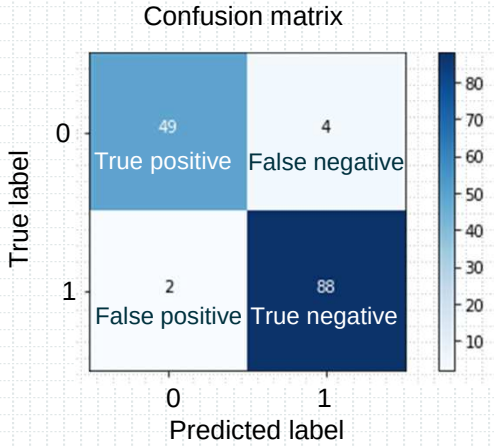
TU Delft

Method - Evaluation

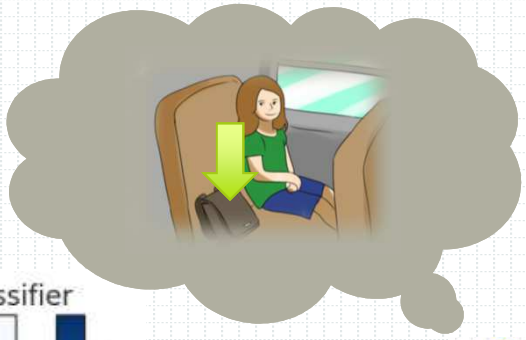
- K-fold Cross validation
  - Randomly select whole days

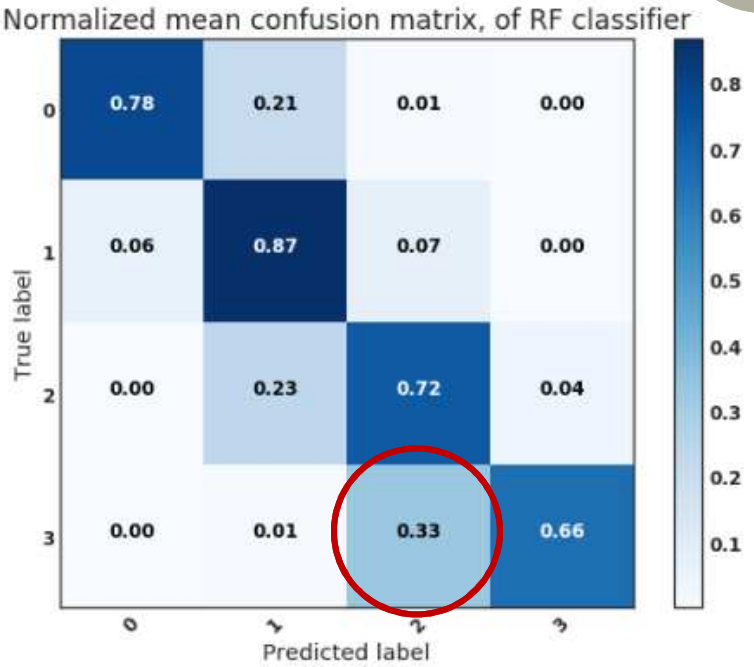


- Evaluation parameters:
  - Confusion matrix
  - punish false negative harder
  - I.e. avoid low recall: use F2-score



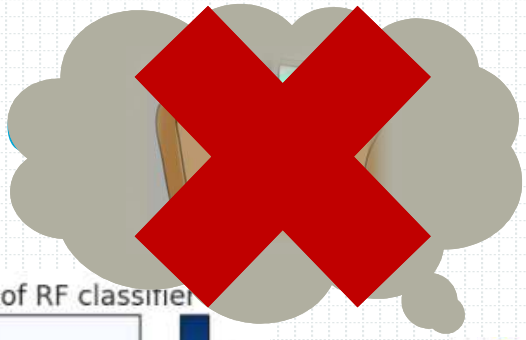
Interpretation

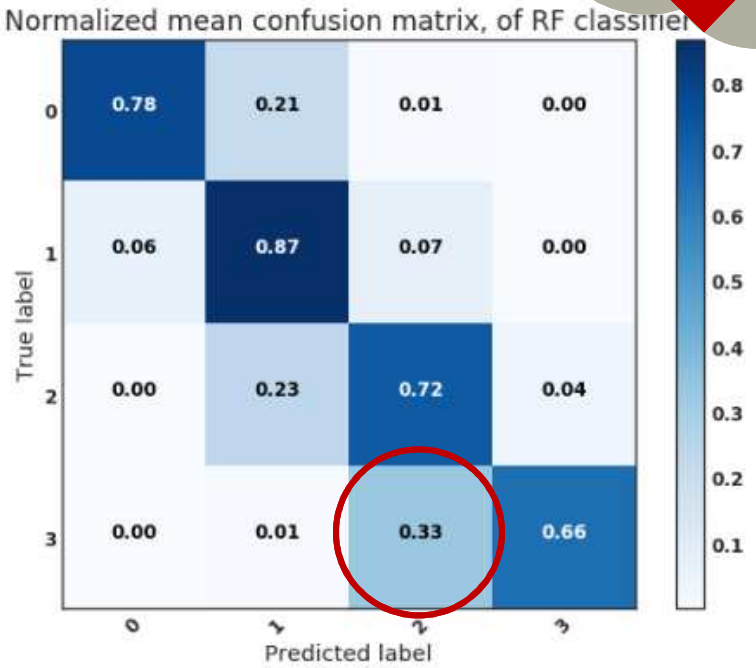




33% of the times where all seats were occupied, model predicts: ‘sit next to someone’

Interpretati





33% of the times where all seats were occupied, model predicts: ‘sit next to someone’

Features

- Day
- Line
- nth stop in sequence
- Direction ID
- Scheduled (GTFS) arrival time
- Scheduled (GTFS) Trip-ID
- Previous Trip-ID

General occupancy prediction

- Departure Delay at previous stop
- Arrival Delay this stop

Real time AVL data – Predict current stop

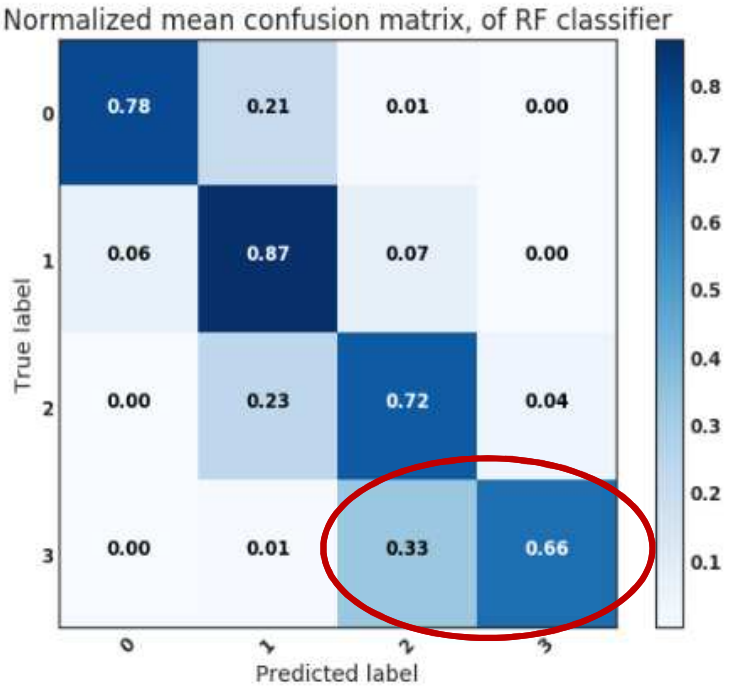
TU Delft

Results: Normal classification

Light rail: classification – no weights, no sampling						
	Static Features			Static+AVL Features		
Classifier	F2 score	Class 0 Recall	Class 3 Recall	F2 score	Class 0 Recall	Class 3 Recall
Random Forest	**0.80***	0.78	**0.65***	**0.81***	**0.78***	**0.66***
Gradient Boosting	0.78	0.73	0.57	0.78	0.73	0.58
Multilayer Perceptron	**0.80***	0.76	0.60	0.80	0.77	**0.66***
K-Nearest Neighbor	0.78	**0.79***	0.62	0.72	0.75	0.60

***** best in column

Results: Normal classification

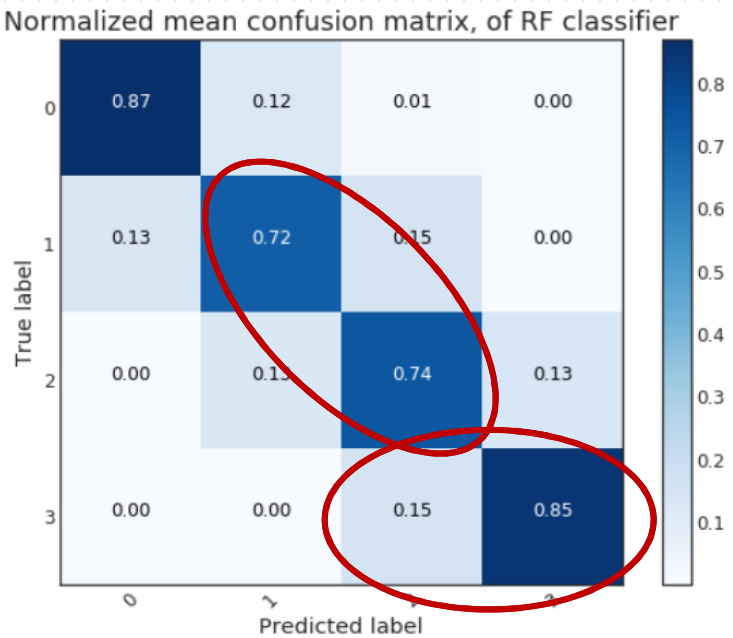


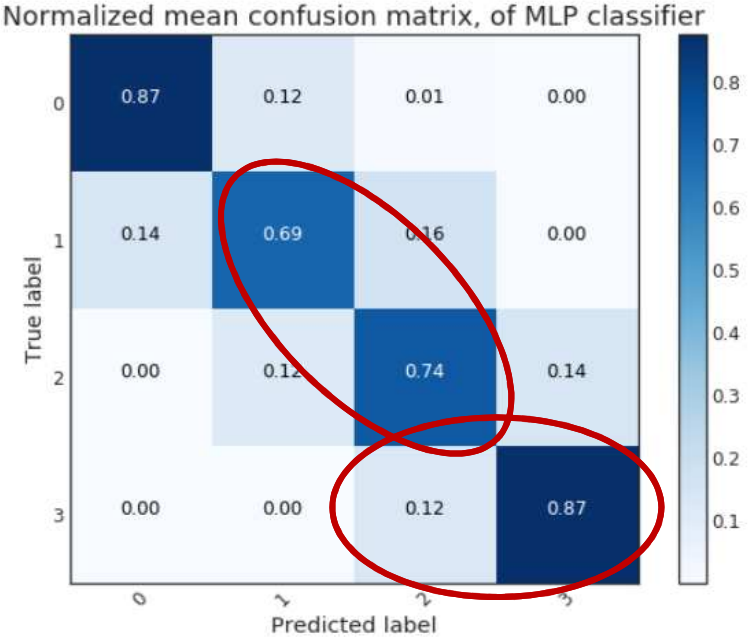
Results: Sampling

Light rail: classification with oversampling**						
	Static Features			Static+AVL Features		
Classifier	F2 score	Class 0 Recall	Class 3 Recall	F2 score	Class 0 Recall	Class 3 Recall
Random Forest	**0.79***	0.81	0.71	**0.80***	0.81	0.72
Gradient Boosting	0.74	0.82	0.78	0.76	0.81	0.78
Multilayer Perceptron	0.76	**0.87***	**0.85***	0.77	**0.87***	**0.86***
K-Nearest Neighbor	0.78	0.79	0.67	0.70	0.79	0.67
Light rail: classification with undersampling***						
Random Forest	**0.76***	**0.87***	0.84	**0.76***	**0.87***	0.85
Gradient Boosting	0.72	0.85	0.83	0.72	0.84	0.84
Multilayer Perceptron	0.74	**0.87***	**0.86***	0.75	**0.87***	**0.88***
K-Nearest Neighbor	0.69	0.85	0.85	0.63	0.82	0.82

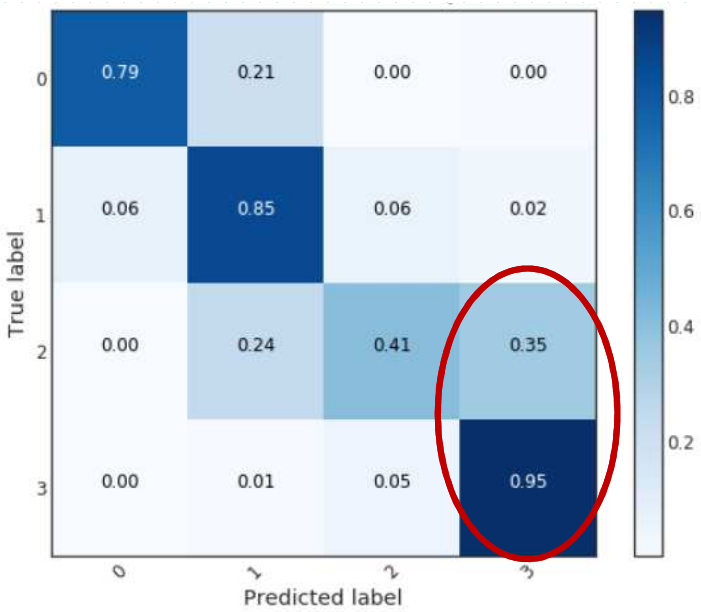
*best in column
** SMOTE - Synthetic Minority Over-sampling Technique
*** Random sampling without replacement

Results: Undersampling





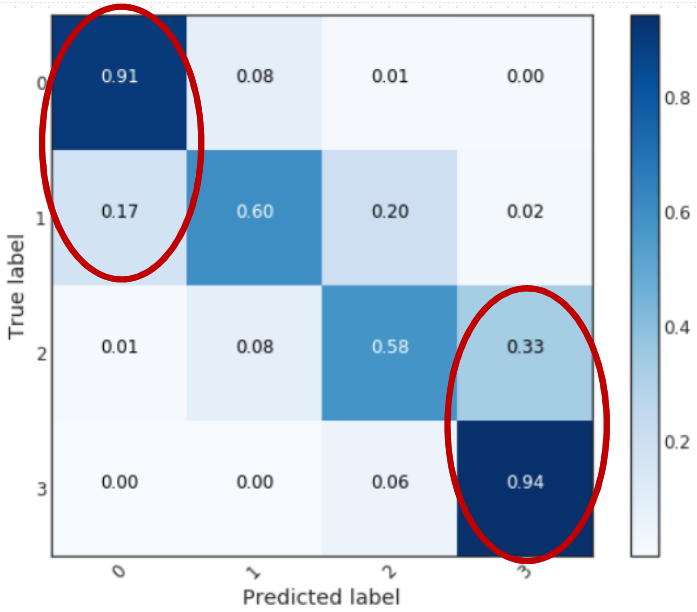
Results - Sequential classification



1. Predict all seats occupied versus rest
   Model: Multilayer Perceptron,
   Sampling

2. Predict the kind of seat
   (circumstances)
   Training: no class 3 data
   Model: Random Forest
   No sampling

Results - One-versus-all sequential classification with sampling



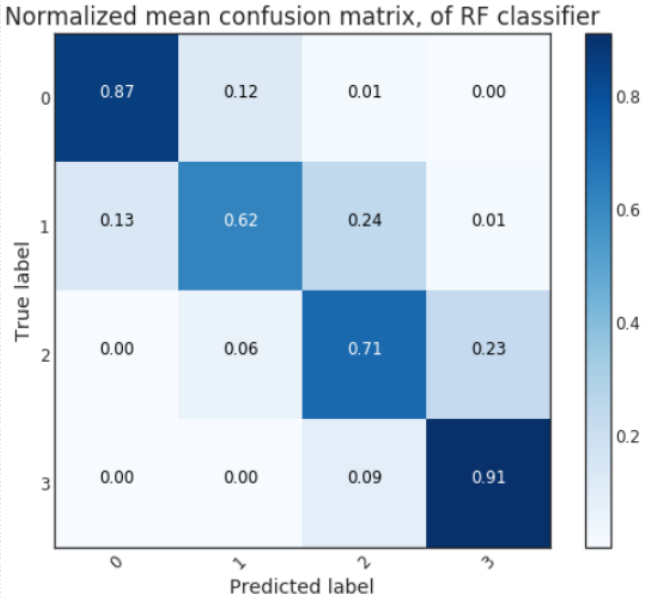
- Training: each class in a one-vs-all way
  Model: Multilayer perceptron
  Sampling: Yes
- Prediction overruling: 3 > 0 > 2 > 1

Results – cost sensitive classification with sampling

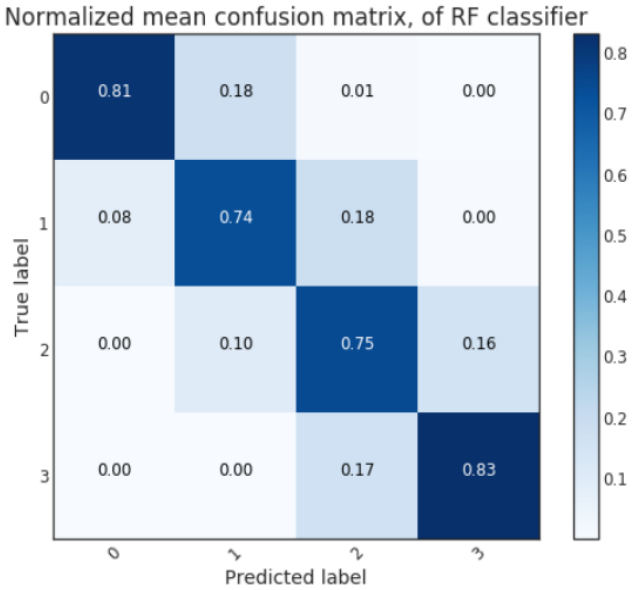
Lightrail: Cost sensitive classification with undersampling								
	Static Features				Static+AVL Features			
Classifier	F2 score	Class 0 Recall	Class 3 Recall	Weights	F2 score	Class 0 Recall	Class 3 Recall	Weights
Random Forest	0.70	0.87	0.91	1,1,5,100	0.71	0.87	0.91	1,1,5,100
Lightrail: Cost sensitive classification with oversampling								
	Static Features				Static+AVL Features			
Classifier	F2 score	Class 0 Recall	Class 3 Recall	Weights	F2 score	Class 0 Recall	Class 3 Recall	Weights
Random Forest	0.76	0.81	0.83	1,1,5,100	0.79	0.81	0.81	1,1,5,100

TU Delft

Results – cost sensitive classification with sampling



Undersampling



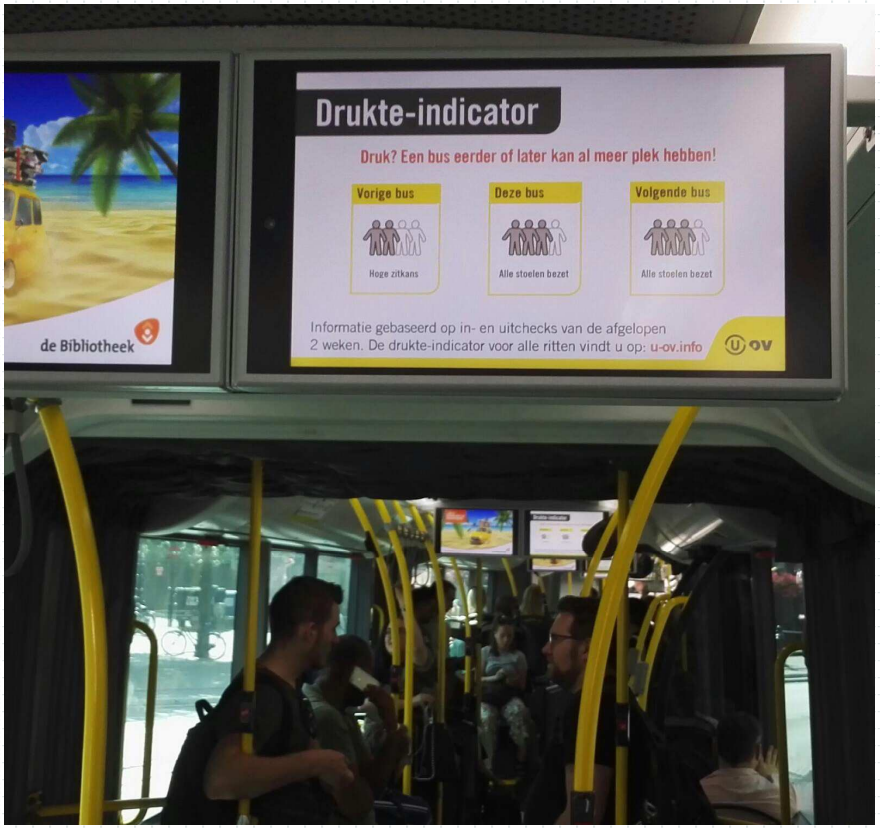
Oversampling

Conclusion

- Predicting general occupancy, using static and ‘real-time’ AVL data
- Tried several approaches:
  - Sampling
  - Sequential, with sampling in first step
  - One-versus-all with sampling
  - Cost sensitive with sampling

TU Delft

Relevance – Utrecht example



Future research

- Test on more data, incl. bus and tram lines
- Add ‘real-time’ passenger load data from 60 min. before
- Investigate: when is real time, real time enough?
- Including other real time data sources like weather

TU Delft

Questions?

L.J.C.heydenrijk-ottens-1@tudelft.nl

Acknowledgements:

- This research was supported by H2020 project My-TRAC (Grant No. 777640).
- HTM and Stichting OpenGeo for providing the AFC and AVL datasets, respectively.

TU Delft