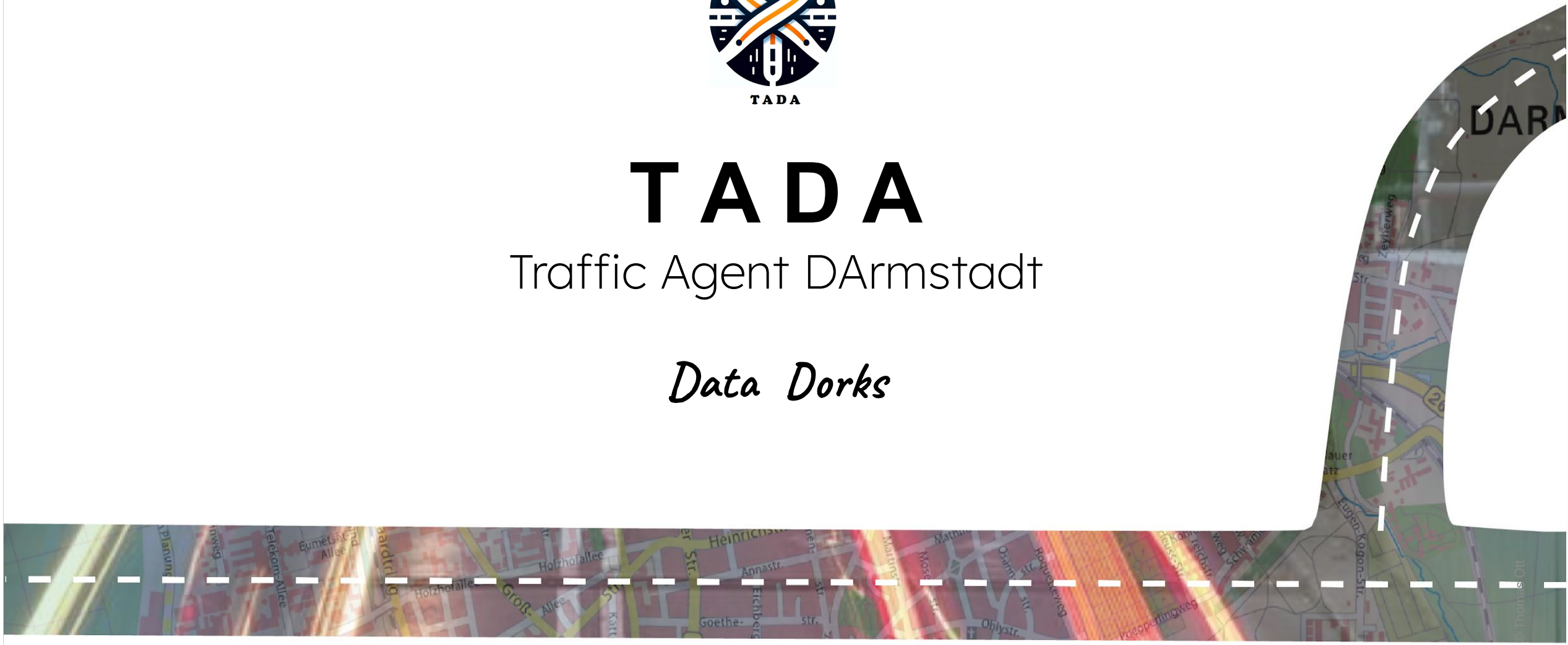
TECHNISCHE UNIVERSITÄT DARMSTADT

TADA

TADA

Traffic Agent DArmstadt

Data Dorks

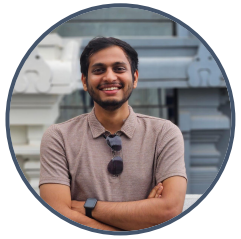


DATA DORKS

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bahar Bayer (Ranjbaran) (Team Member)

- Sudhanva Sridhar Bhat (Team Member)



- Lisabeth Maria George (Team Member)

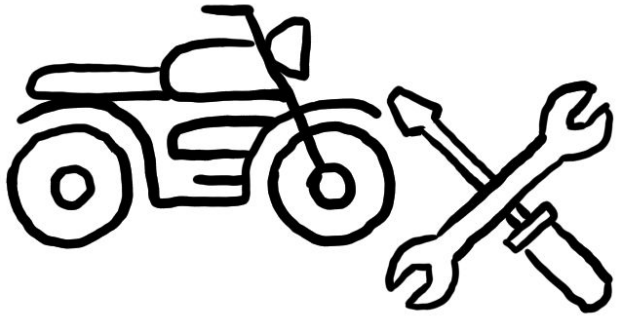
- Tresa Maria Joseph (Team Member)

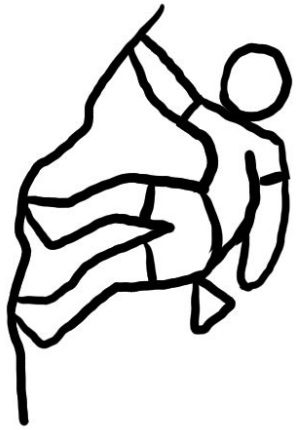
- Nora Simon (Team Member)

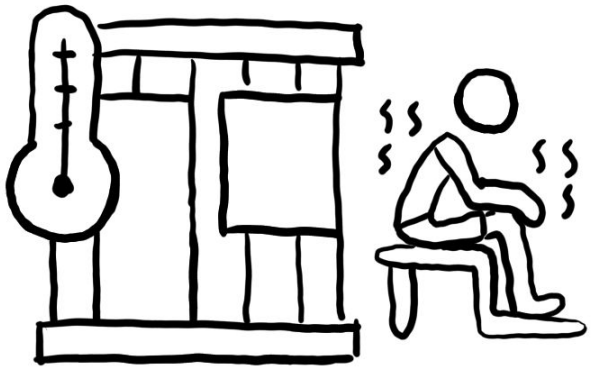
- Dr. Olivier Boisard (Supervisor)

Outline

Problem Statement

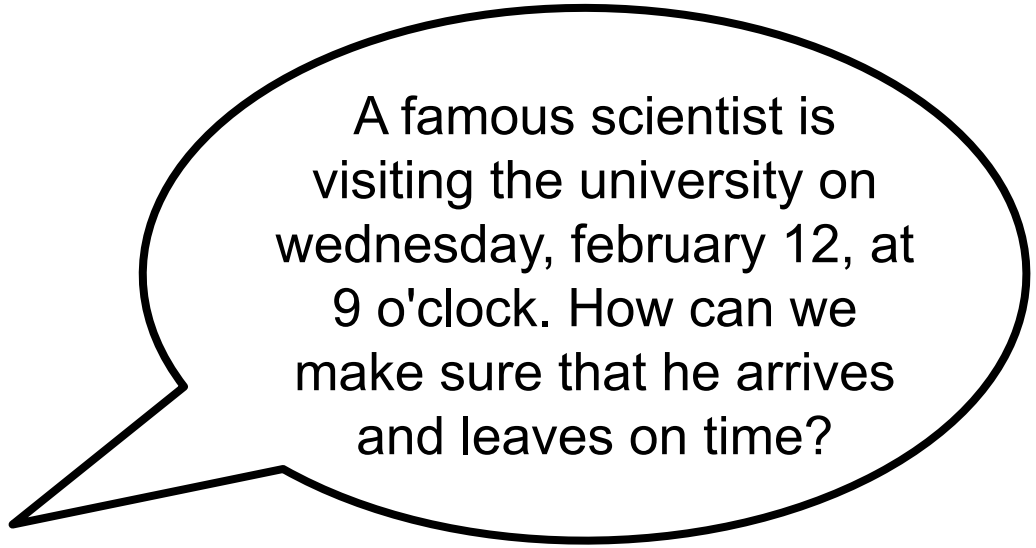






Problem Statement

Problem Statement



Problem Statement

How to make the best decision?

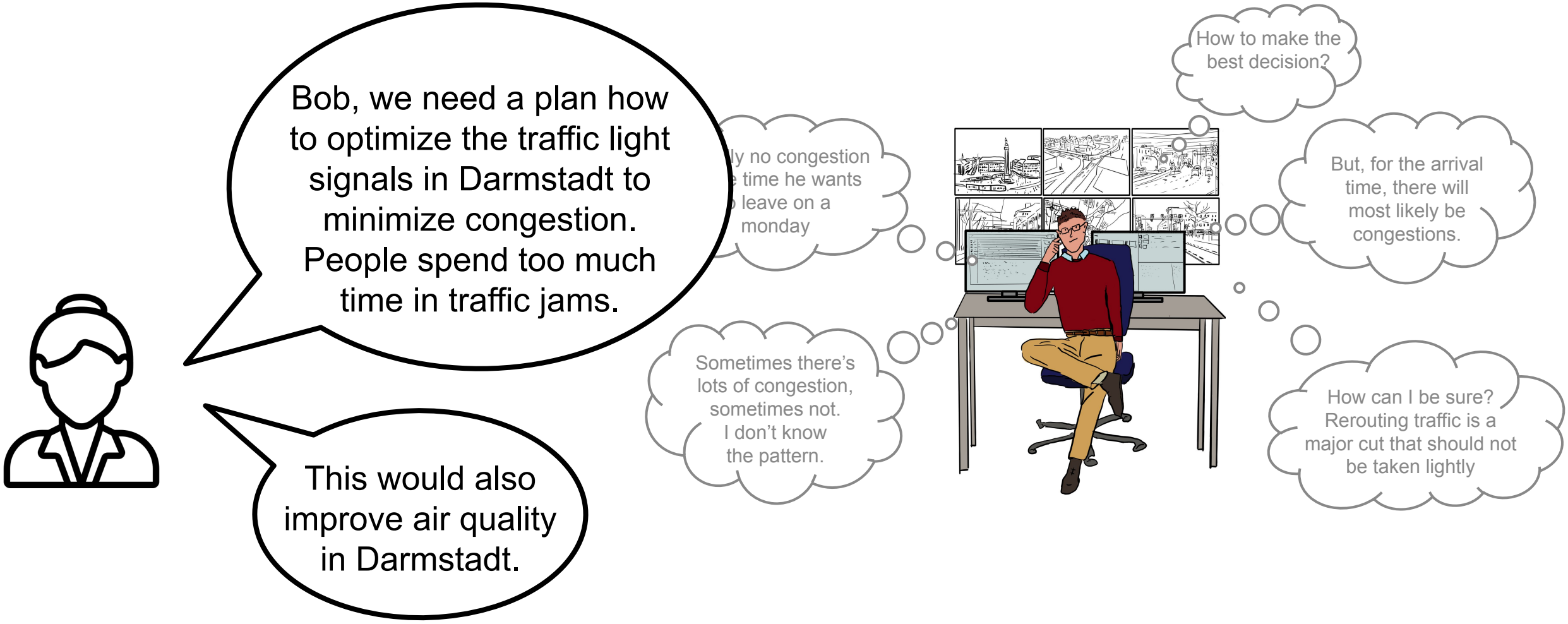
Usually no congestion at the time he wants to leave on a monday

But, for the arrival time, there will most likely be congestions.

Sometimes there's lots of congestion, sometimes not. I don't know the pattern.

How can I be sure? Rerouting traffic is a major cut that should not be taken lightly

Problem Statement



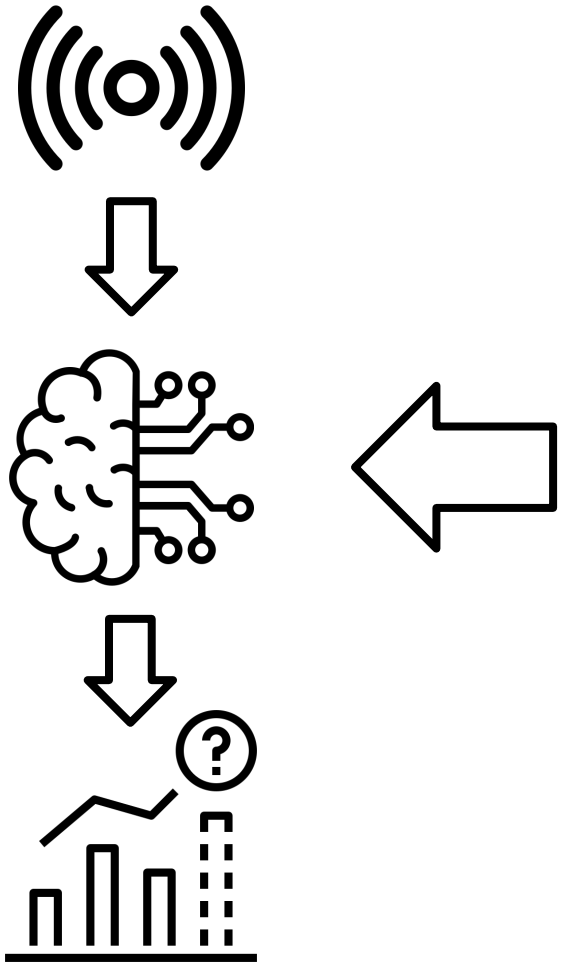
Problem Statement

I just wanted to go from Roßdorf to Groß-Gerau, and it's faster to drive 10km more around Darmstadt to get there.

Problem Statement

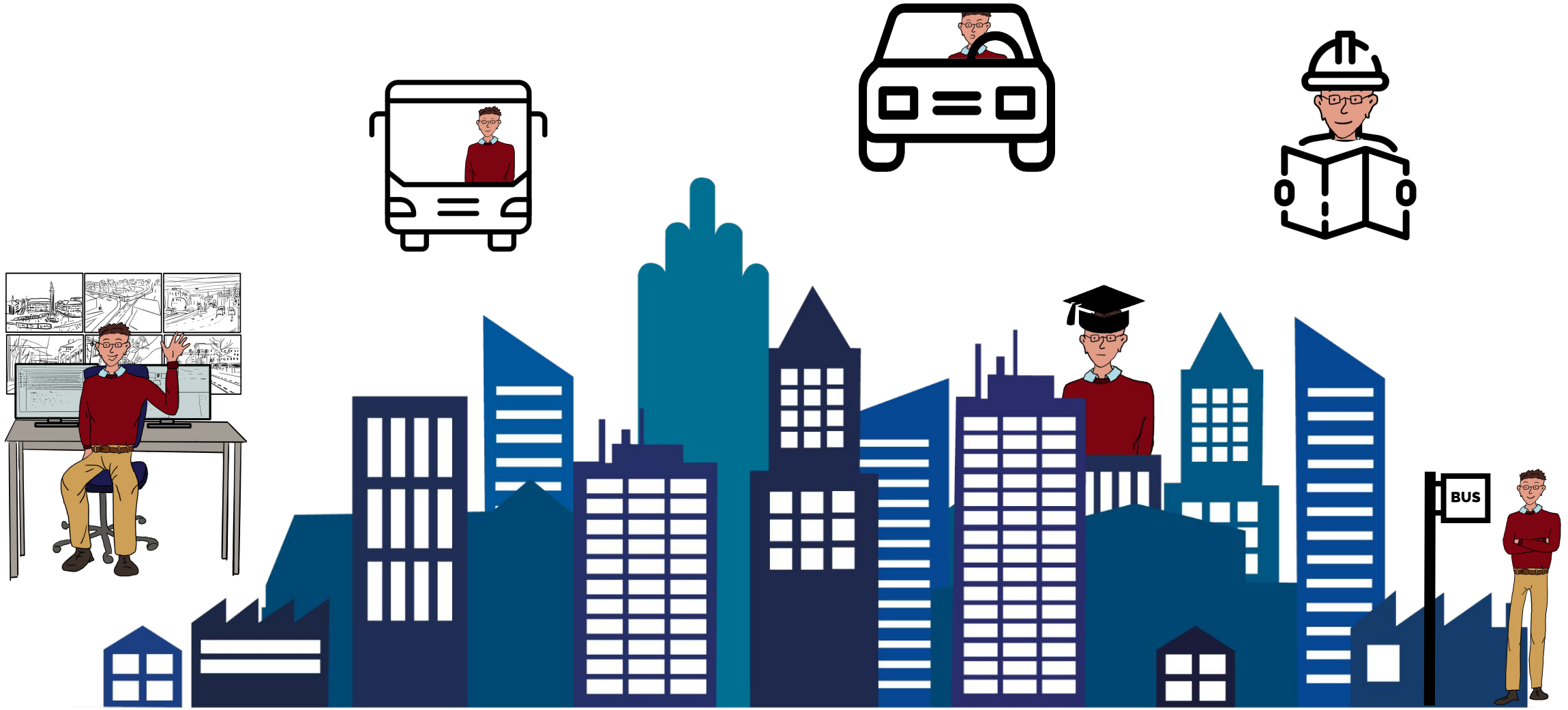
TADA:
Traffic Agent DArmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

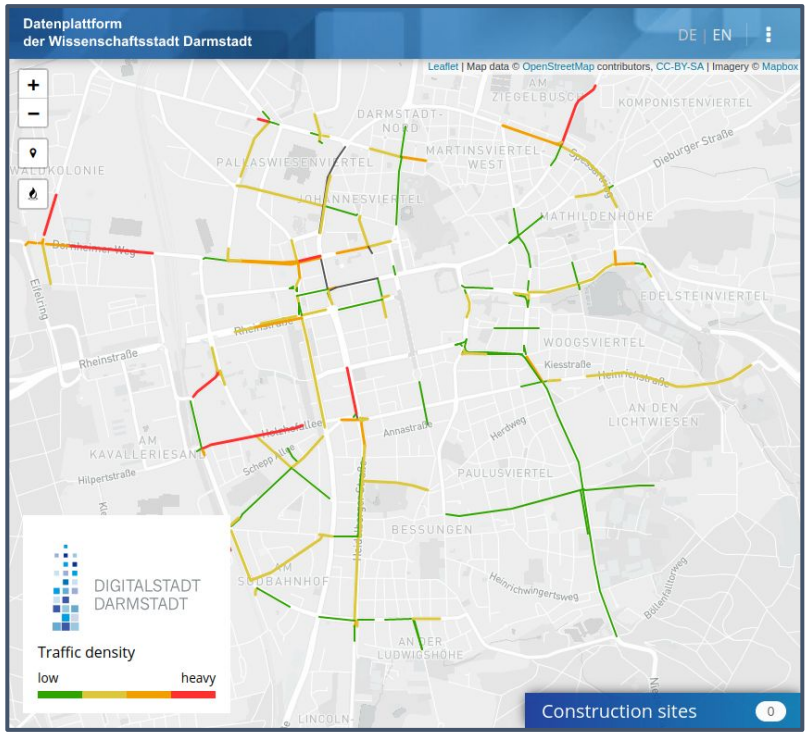


- Identify critical junctions early on before congestion occurs
- Balance traffic by controlling traffic light signals
- Analyse traffic over long term to improve traffic concepts
- Precisely plan traffic at specific times

Potential Users



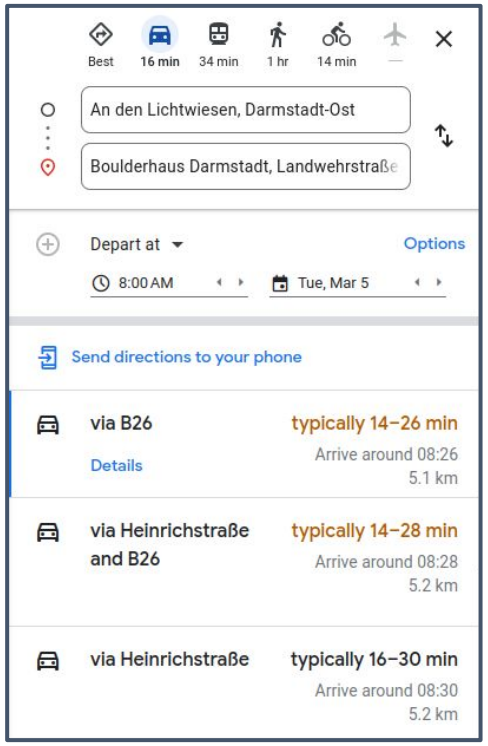
Market Analysis



Traffic user interface of Datenplattform Darmstadt

- Use of Internal sensor data
- No prediction
- No navigation

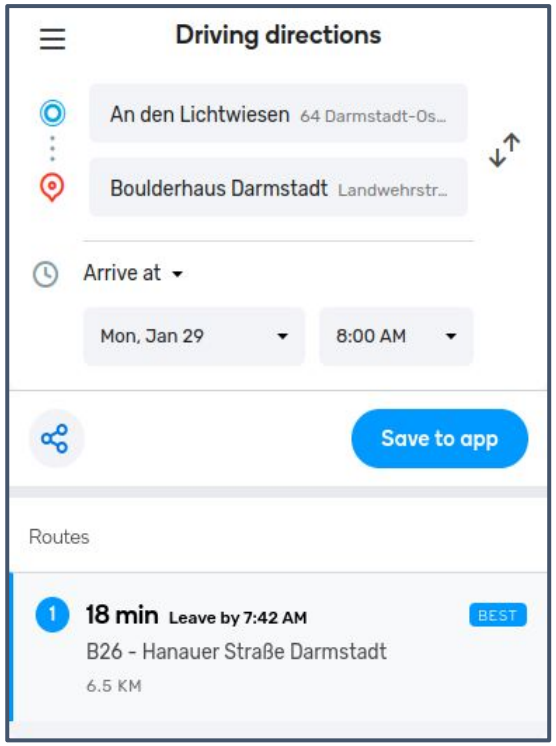
Google Maps



Route prediction with Google Maps

- No Internal sensor data
- Imprecise prediction

waze



Route prediction with Waze

- No Internal sensor data
- Prediction: seven days

Business Application

Data is publicly available

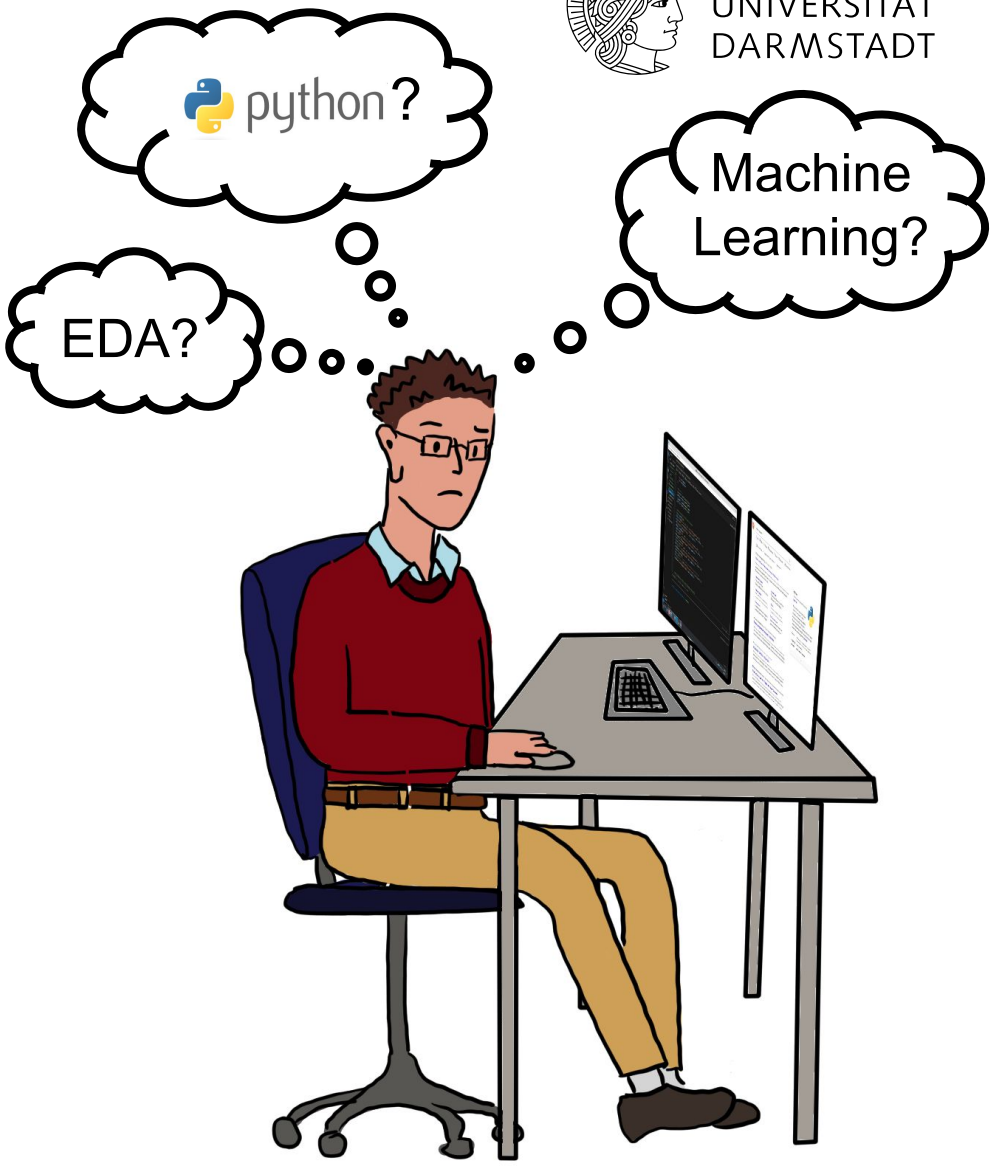
- Bob could read it out himself
- Very complex, high hurdle if in-depth data science and Programming knowledge is lacking

TADA

- Intuitive user interface
- Extendable

indirect positive effects

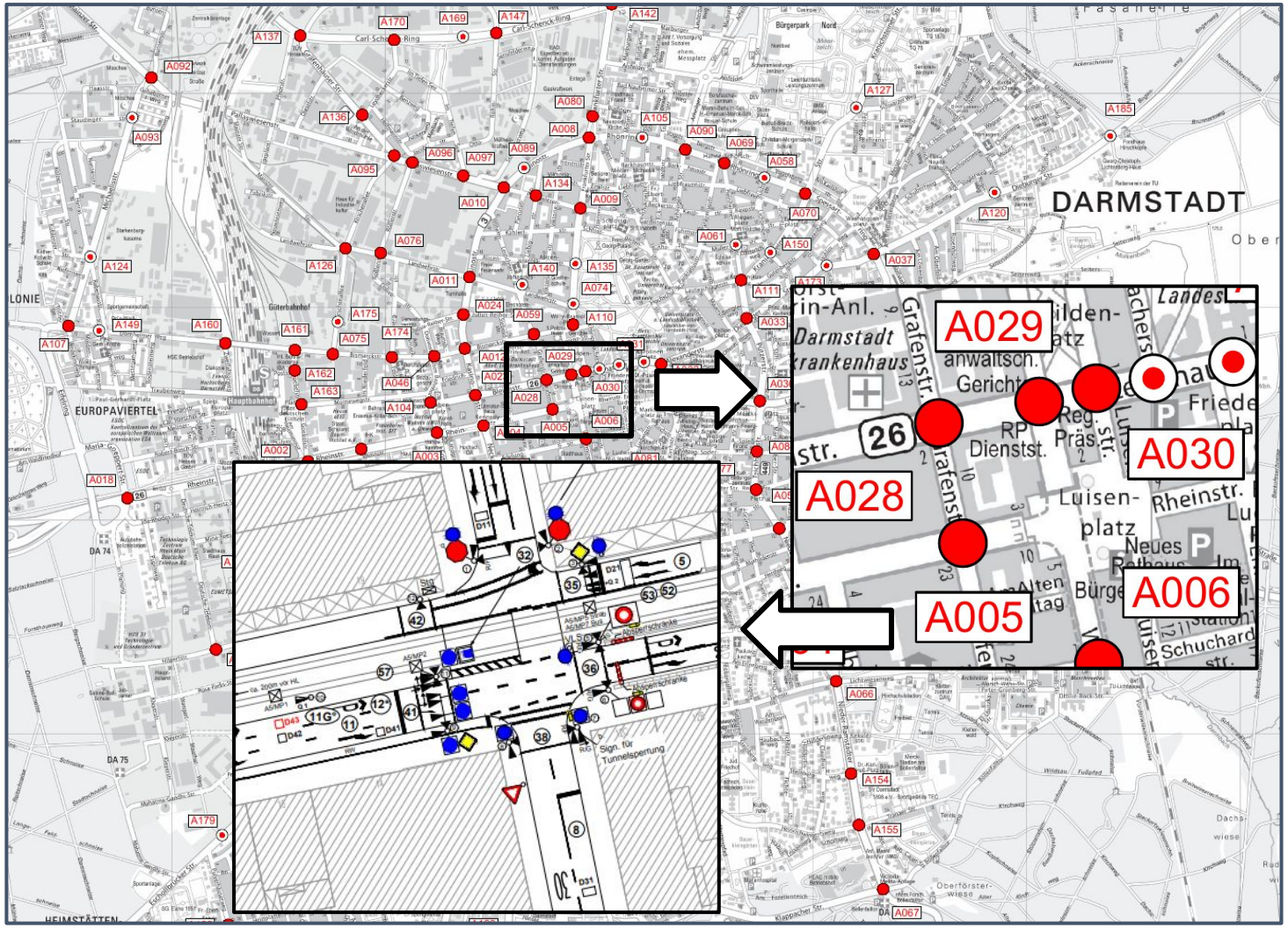
- improvement of air quality
- increased safety



Data Analysis and Interpretation

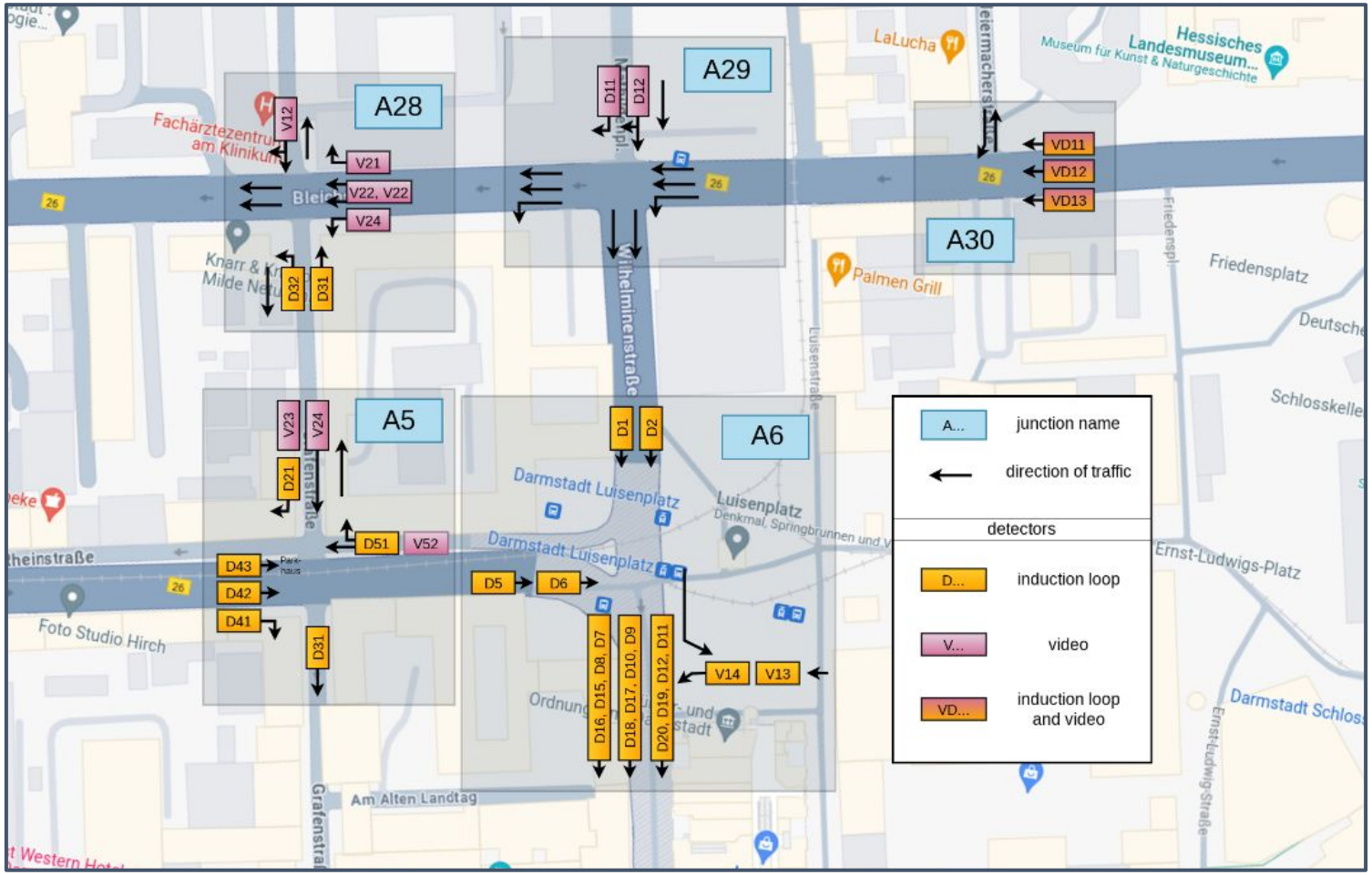
TECHNISCHE UNIVERSITÄT DARMSTADT

- Data acquisition at 152 junctions
- Different sensors for motor vehicles, pedestrians and public transport
  - Induction loops and video detectors most common
  - Traffic counts at specific timestep
- Recording since May 2021 to May 2023
- Minimum Viable Product
  - 5 junctions in the center of Darmstadt
  - Focus on motor traffic



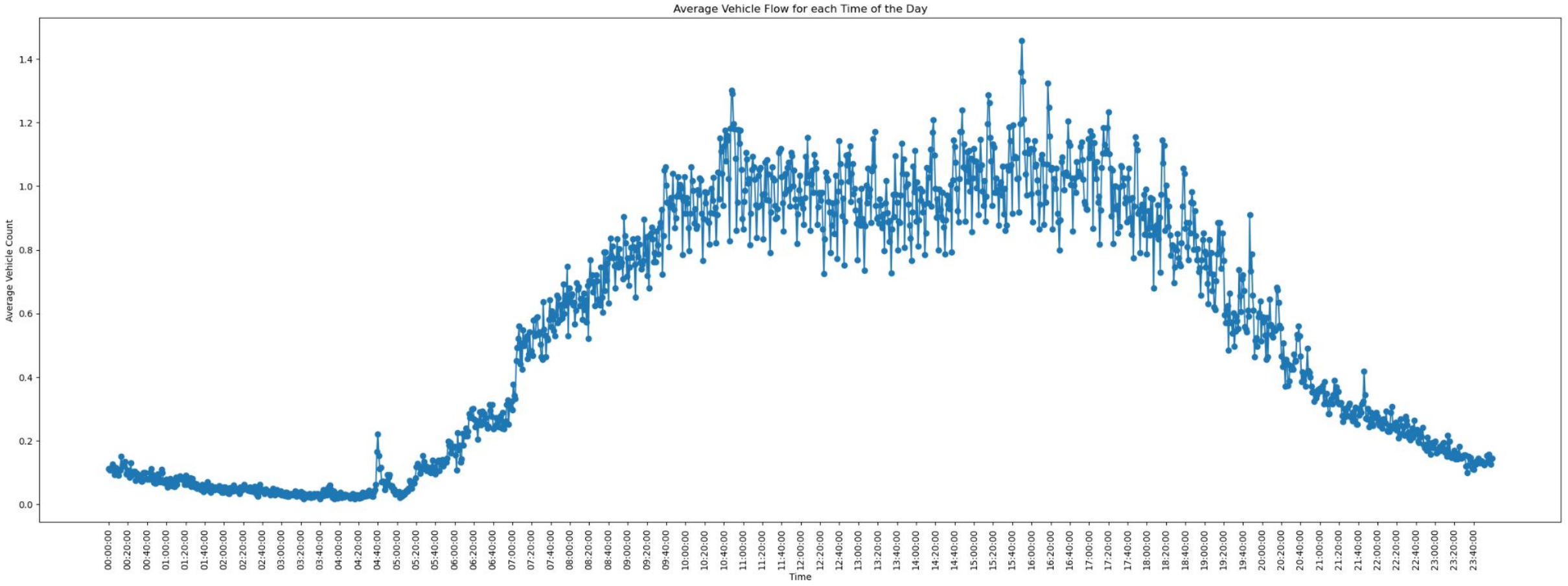
Overview of traffic light systems with detectors from Datenplattform Darmstadt

Data Analysis and Interpretation

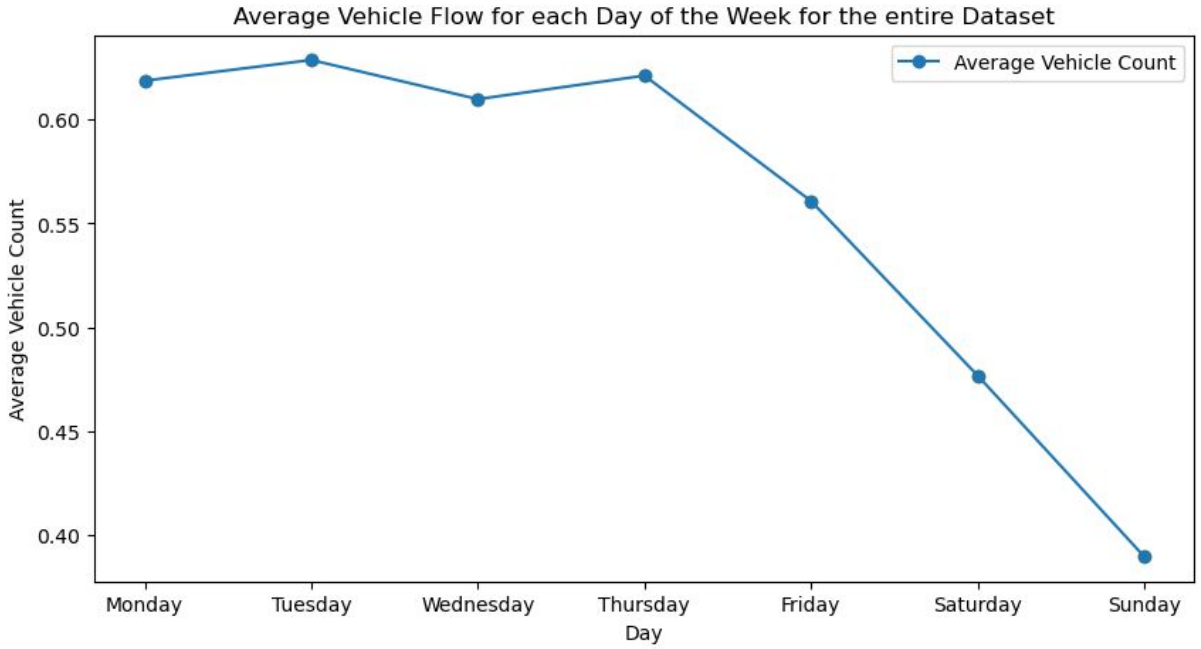


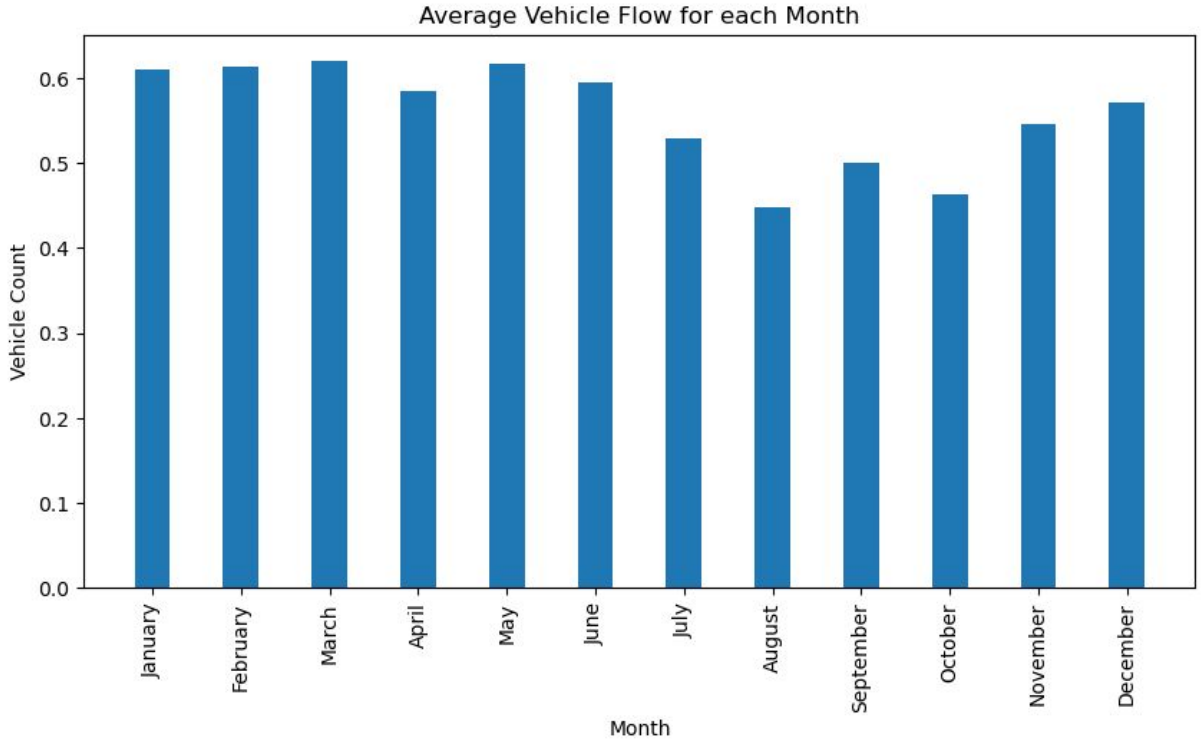
Motor traffic detectors at junctions A5, A6, A28, A29 and A30 (sketch, not to scale)

Data Analysis and Interpretation



Data Analysis and Interpretation





Model

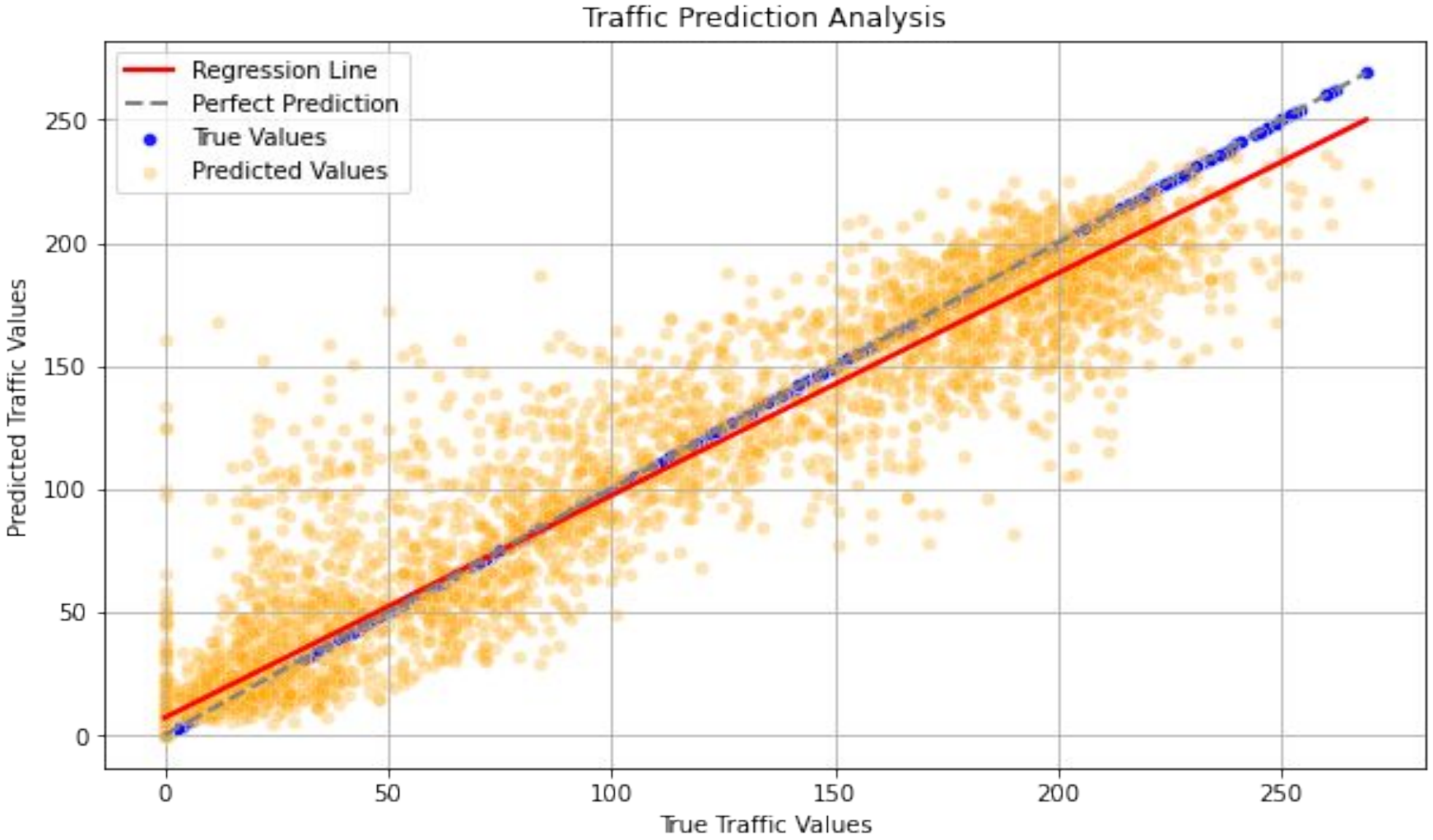
- Time-series traffic data prediction: Predict the number of cars per hour
- Suitable models include LSTM, ARIMA, Prophet, Random Forest, LightGBM and CatBoost

	MAE (Mean Absolute Error)	R^2 (Coefficient of Determination)	SMAPE (Symmetric Mean Absolute Percentage Error)
Prophet	3.611	-	-
Random Forest	3.834	0.94	102.21
LightGBM	4.069	0.938	102.72
CatBoost	4.10	0.94	102.72

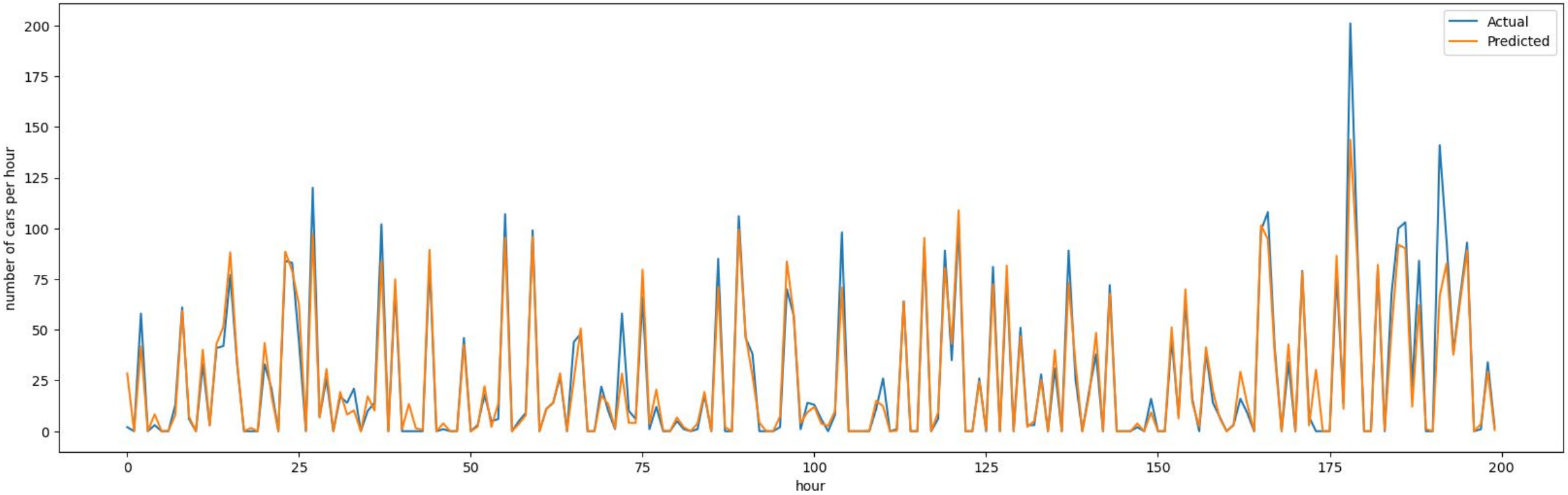
not usable if predicted value is 0

Model comparison metrics

Random Forest



Random Forest



Random Forest: first 200 predicted and actual values

Random Forest Evaluation

How to evaluate the model?

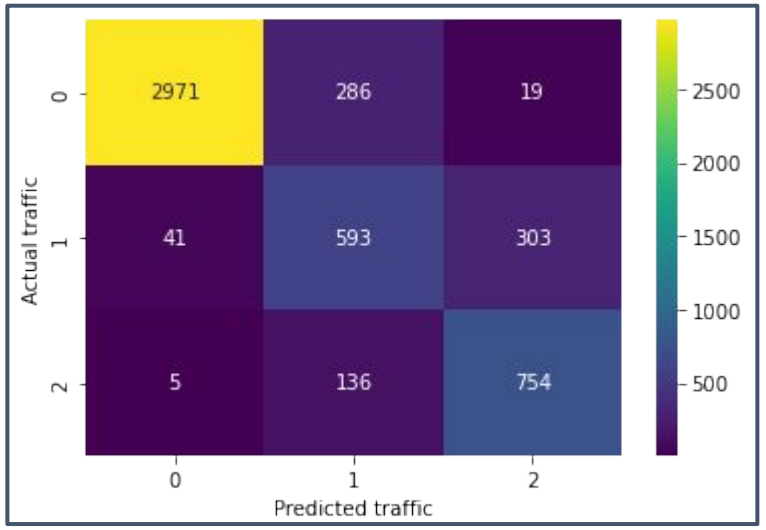
- Use metrics from scikit-learn

Hyperparameter tuning:

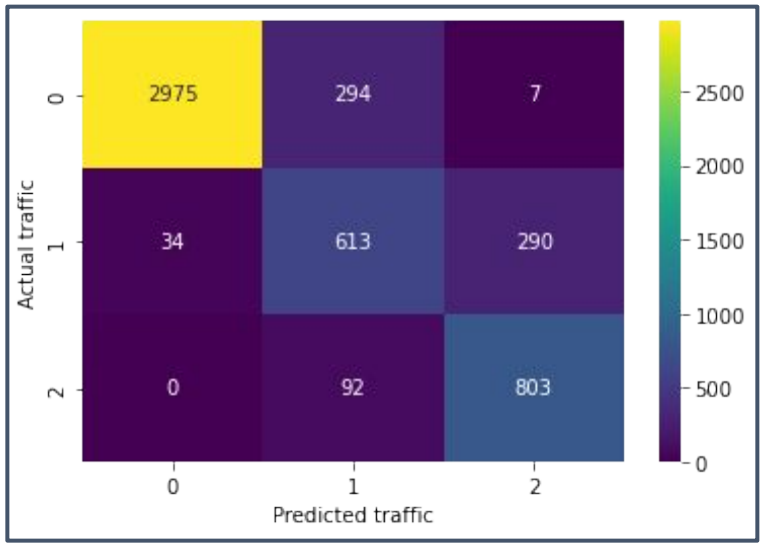
- Basic model performs best
- Comparison using Confusion Matrix

Cross validation tricky with time-series data

- Cannot mix past and future
- Not possible to use cross-validation from libraries



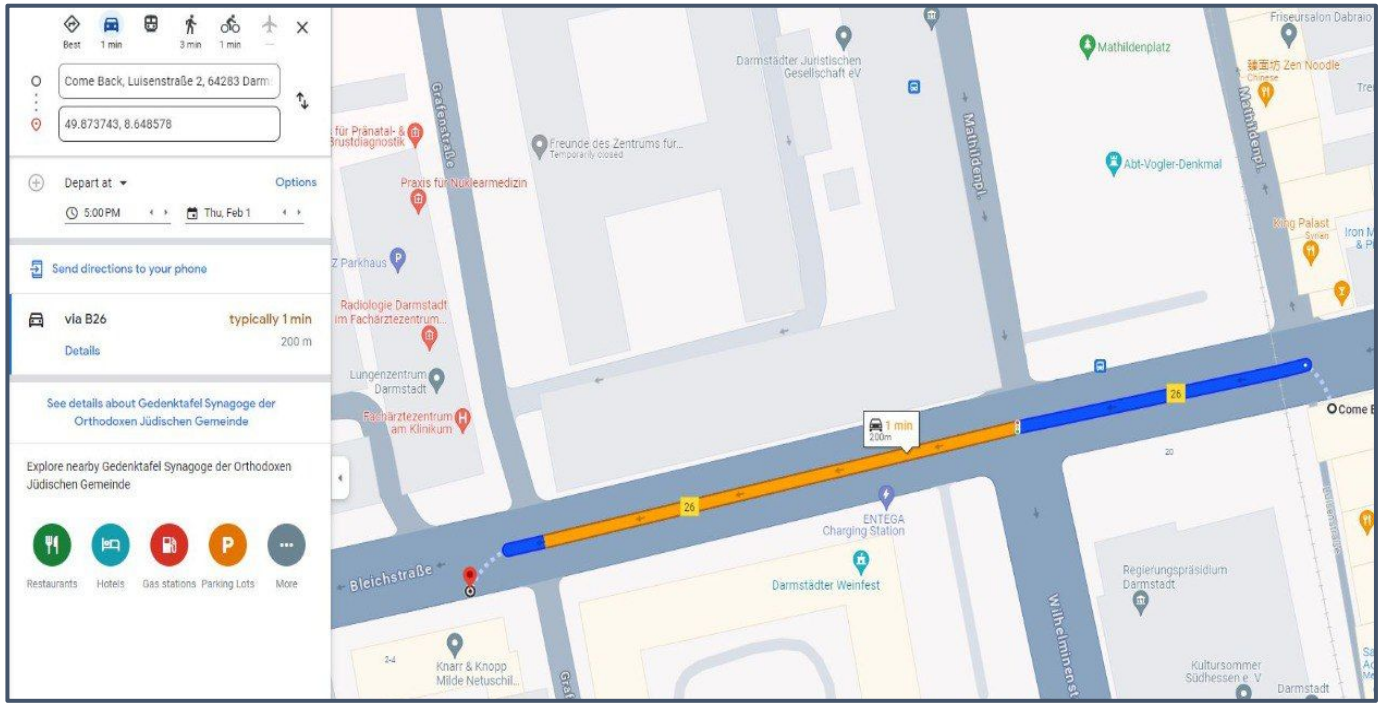
confusion matrix of hyper-parameter tuned model



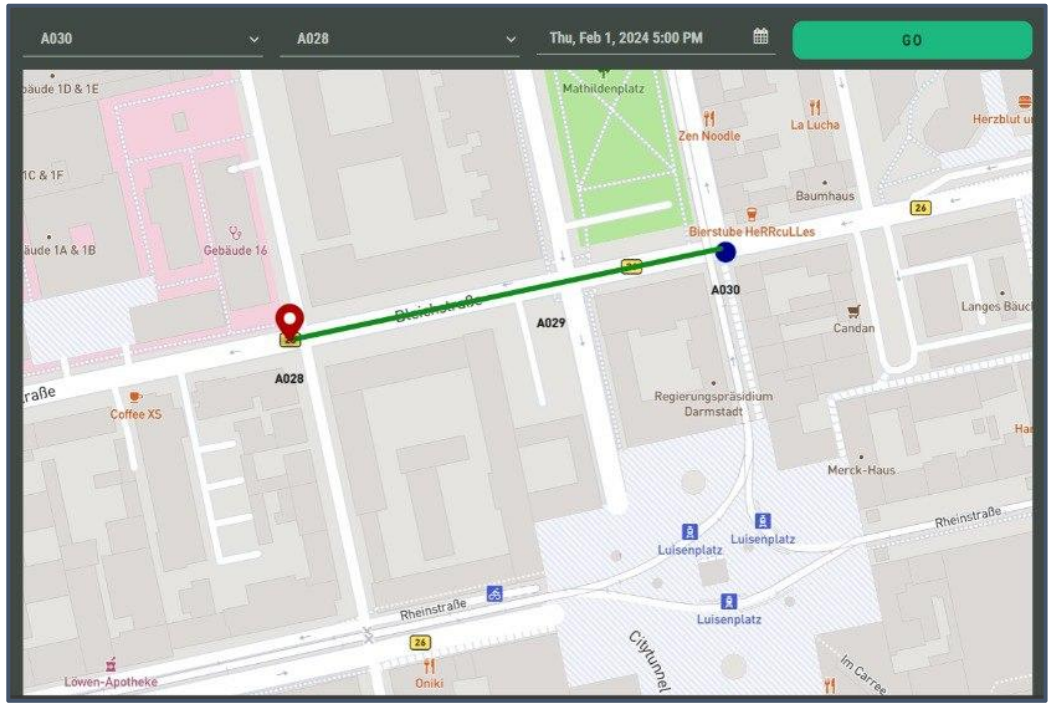
confusion matrix of Random Forest model

Sample: Comparing Google Maps and TADA

Data predicted for 1st February 2024, 17:00 by google and TADA



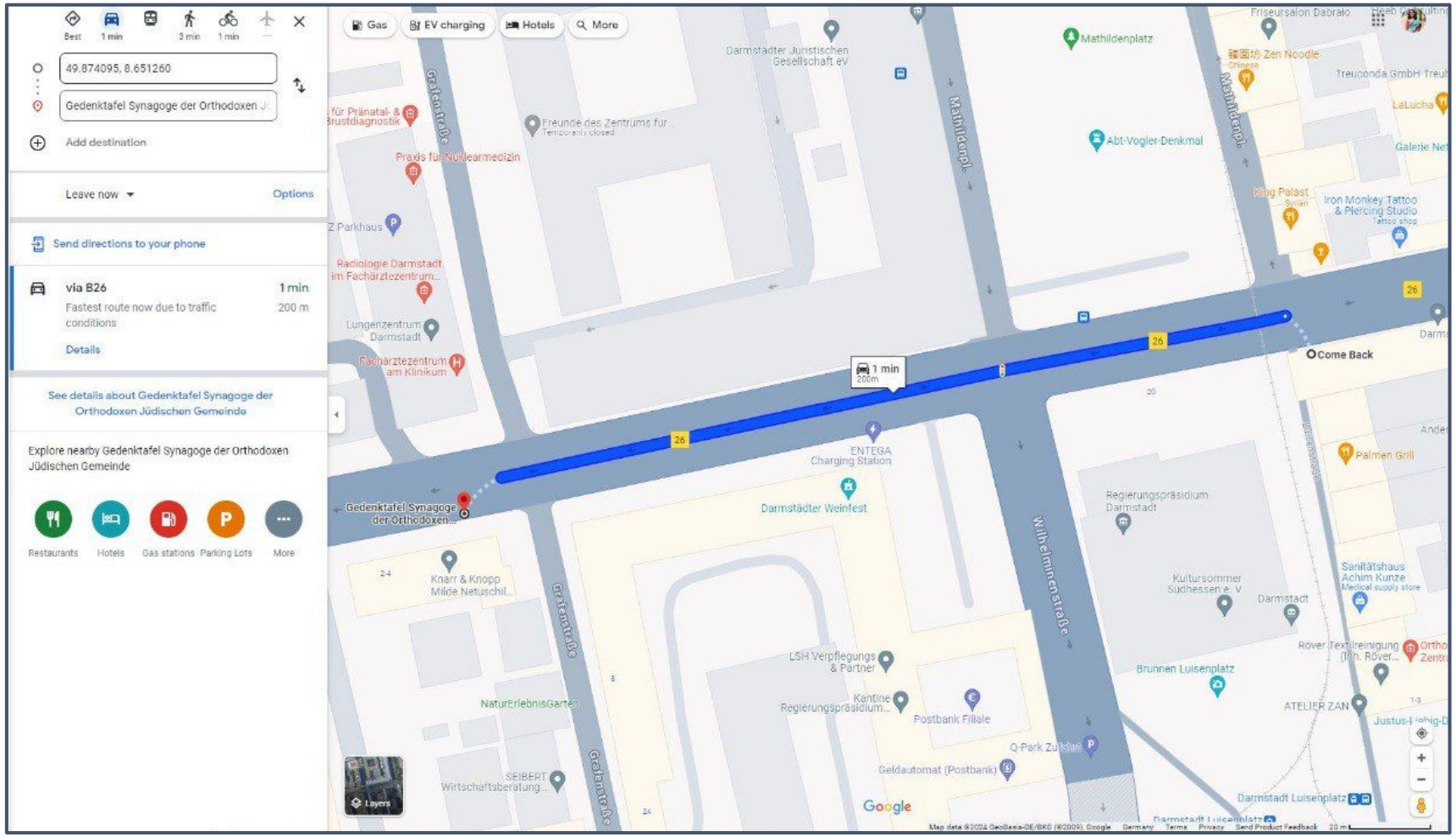
Google Maps congestion prediction for 1st February at 5 pm



TADA congestion prediction for 1st February at 5 pm

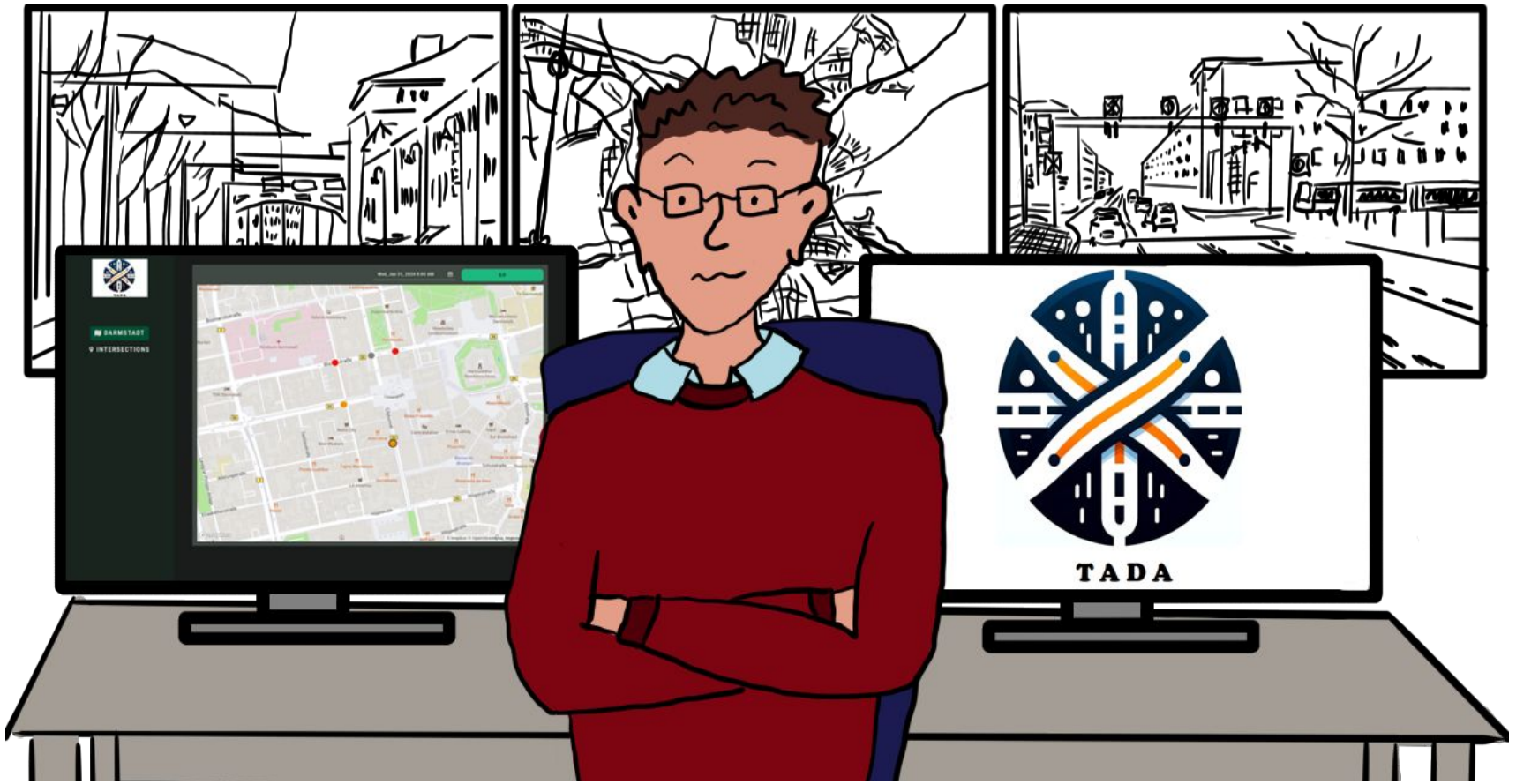
Sample: Result

Actual data on 1st February 2024, 17:00



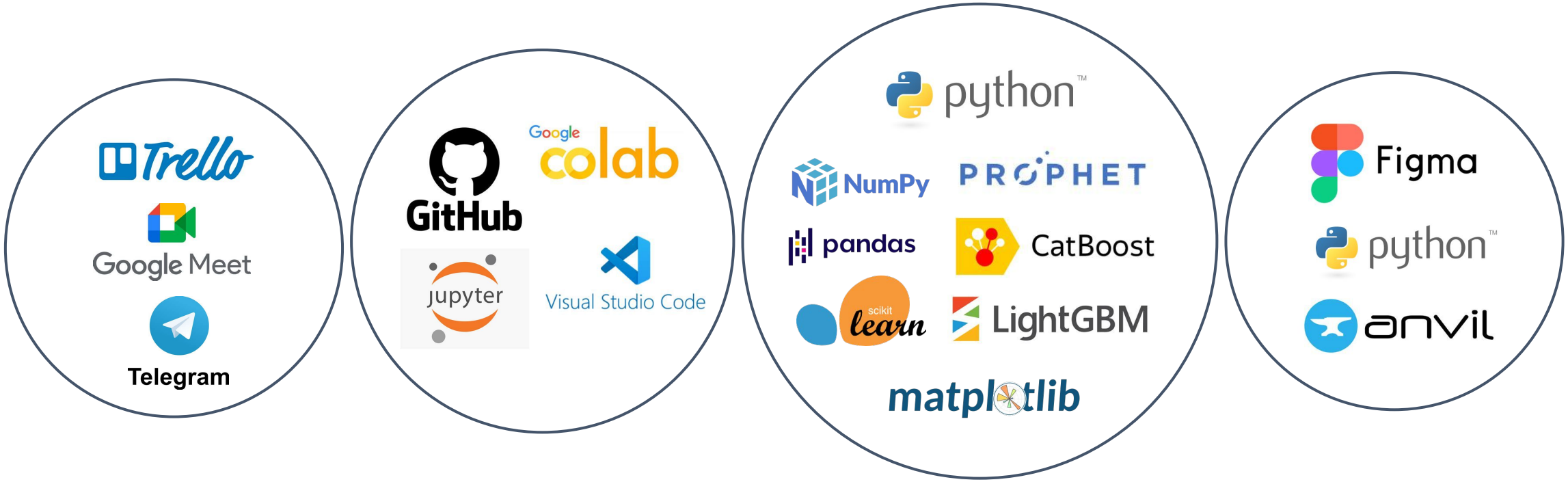
Google Maps congestion at 1st February at 5 pm

From model to application



Technology Stack

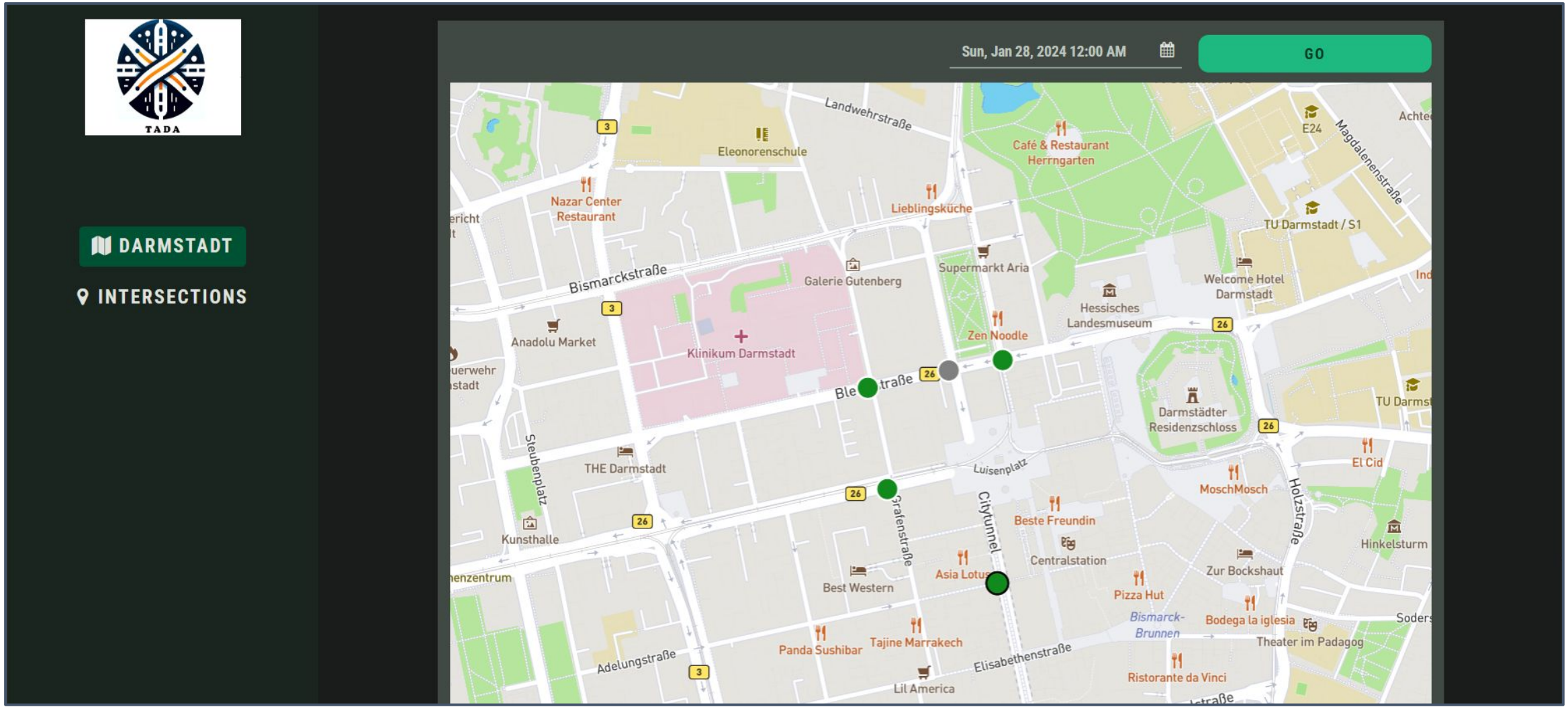
TECHNISCHE UNIVERSITÄT DARMSTADT



Project Management | Development | Backend | Frontend

Demo

TECHNISCHE UNIVERSITÄT DARMSTADT



TECHNISCHE UNIVERSITÄT DARMSTADT

Would you use it?

Limitations

TECHNISCHE UNIVERSITÄT DARMSTADT

- Implementation restricted to selected junctions at the moment.
- Real time data integration.
- No regular pattern in junction design and placement of sensors.
- Assumptions and simplifications made in the predictive models may lead to variations in real-world scenarios.
- Model Limitation: Climate factors are not accounted for in this analysis.
- The model may struggle to account for sudden and unexpected events that were not present in historical data.

Future Work & Improvements

TECHNISCHE UNIVERSITÄT DARMSTADT

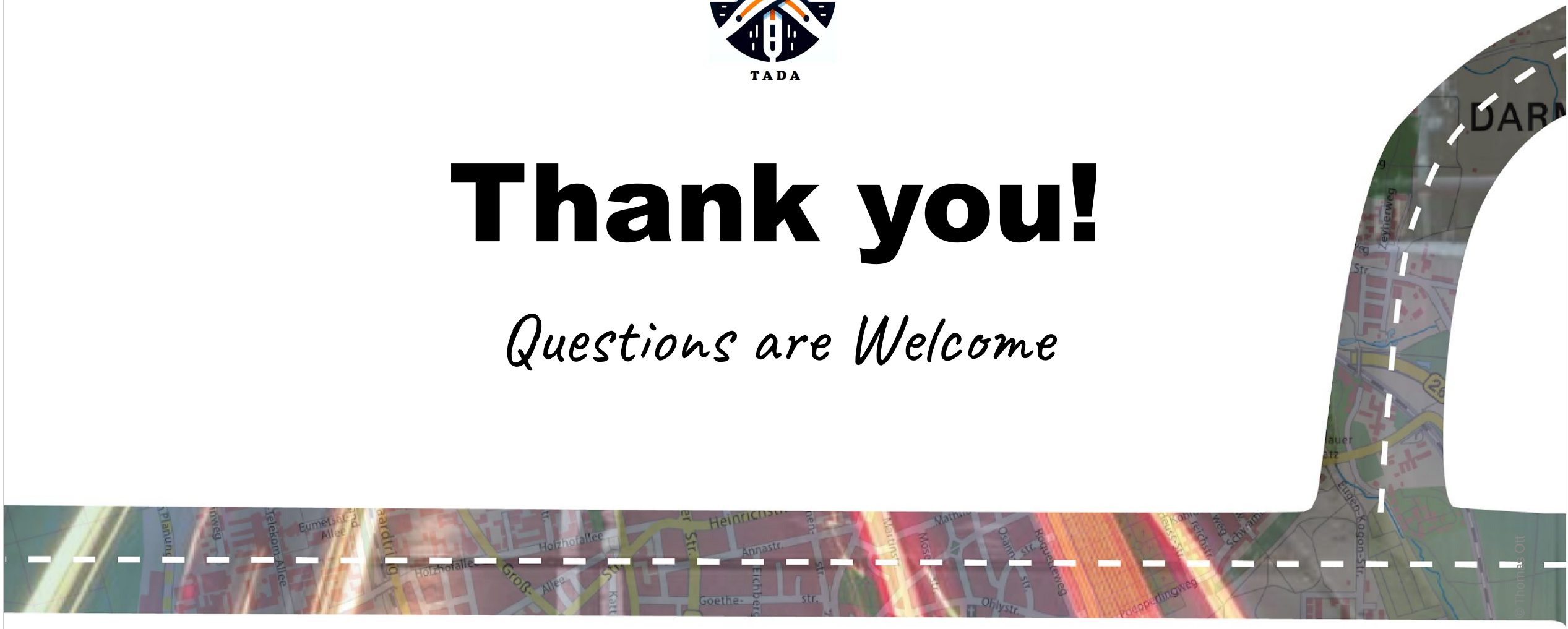
- Integration of wider regions - Eberstadt, Dieburg, Griesheim and Pfungstadt. Even the whole of Germany.
- Predicting alternate route with lesser traffic. Better route to transport from A to B.
- Automating the process - as regions added.
- Real time Data Integration.
- Collaboration with local authorities for further development strategies.
- Utilizing the multiple sensors per junctions.
- Adding prediction confidence level.
- Improve model performance by hyperparameter tuning

TECHNISCHE UNIVERSITÄT DARMSTADT

TADA

Thank you!

Questions are Welcome



DATA SCIENCE CANVAS

Team: Data Dorks | Project name: TADA: Traffic Analysis DArmstadt | Iteration: 2 | Date: 01.02.24

Problem Statement

Congested roads and slow-moving traffic are problems faced by commuters in Darmstadt. The existing challenges necessitate innovative solutions to optimize traffic flow, reduce environmental impacts, and enhance overall transportation efficiency in the city.

Solution

Through our application, historical data can be used to predict future traffic flow. As such, this application can provide insights for traffic department staff and help them improve urban transportation in various ways.
For instance, critical junctions can be identified early on before congestion occurs, traffic can be analyzed over long-term for urban planning and to improve traffic concepts, redirections for construction sites or events can be planned, and traffic light signals can be balanced which leads to less slow-moving traffic.

Users & Use

Our application is aimed at control center agents monitoring and controlling traffic in Darmstadt. Moreover, it offers support in recognizing and planning critical situations. This relieves and supports them in their work.
Indirectly, road users and residents of Darmstadt will benefit the most, as they will be able to move around in the city more quickly and safely and plan routes reliably.

Data Acquisition

"Datenplattform Darmstadt" provides traffic data in CSV format for one month, an overview map of all intersections, a detailed map for each intersection showing the location of sensors, and a JSON file containing sensor configuration data.

An extended dataset of two years was made available after a mail request.

Analytics Formulation

Input: Traffic sensor data from "Datenplattform Darmstadt"
Output: Number of cars per hour (past / present / predicted)
Methodology: Machine learning models: Prophet, Random Forest, LightGBM, CatBoost.
Regression for time-series data was implemented.

Modeling

We performed time-series data prediction for the number of cars per hour. For this purpose, we evaluated the following models: Prophet, LightGBM, CatBoost, and Random Forest. Based on the metrics, Random Forest was deemed as best to use for our application.
After prediction, data was binned to different congestion categories (low / high / medium).

Data Preparation

First, we had to gain deep understanding of the dataset (rudimentary / missing explanation), then data cleansing was performed e.g. intersections that are out of service due to roadworks, merging data from various intersections, connecting which sensors are installed where and at which intersection, and interpolation for missing values.

Evaluation

Evaluation was carried out using a combination of error metrics: MAE, MSE. RMSE, R^2, SMAPE, and additionally, a confusion matrix (for the congestion categories). This approach allows for a nuanced evaluation of the model's accuracy, error magnitude, and predictive performance.

Success Criteria

Objective: the model performs well with respect to the evaluation metrics. It identifies congestion in historical data and provides reliable predictions.
Subjective: users can easily recognize critical moments in the current and future traffic situation. Furthermore, there is a user interface representation that shows the traffic in different colors, based on the amount of congestion, and provides an accurate traffic prediction. The predictions can keep up with and outrun predictions from other tools such as Google Maps and Waze.

Constraints

Updating the model with real time data is a constraint. Unpredictability exists due to regional factors like strikes. Also, there are variations in the sensors. Sensor hardware failure leads to imputation of data which leads to falsification of the prediction. In addition, there is ambiguity in the information about the dataset, sensors, and their positions.

Business Value

A regulated flow of traffic increases the safety and well-being of all road users and residents.
Unlike in other services, the end user is a control center agent from the road authority of Darmstadt. Monitoring and prediction are key in planning the future traffic flow. Agents can determine other ways to organize traffic and difficult traffic conditions beforehand. This can be applied to road planning and instantly adjusting traffic lights.
Utilizing the city sensor data allows precise recording of the traffic situation which results in more accurate future predictions.

MVP

The MVP covers a small major area of Darmstadt: five junctions around the city center and their corresponding traffic flow. It demonstrates the amount of present and future congestion. The traffic situation on routes between these intersections is also displayed.

Key Actors

Internal: traffic sensor data
Customer stakeholders: traffic department of Darmstadt (and other cities), vehicular commuters, general public
External: employees from the mobility office or road planning department of Darmstadt.

Technology stack

Backend: Python and Python libraries (including numpy, pandas, sklearn, prophet, catboost, lightgbm, matplotlib)
Frontend: Figma, Anvil, Python
Project Management: Trello, Google Meet, Telegram
Development: Github, Jupyter Notebook, Google Colab, Visual Studio Code

Image Sources

Datenplattform Darmstadt:

- Traffic user interface: https://datenplattform.darmstadt.de/#!/map/trafficCongestion
- open traffic data (dataset, junction plans from slide 15) : https://datenplattform.darmstadt.de/verkehr/apps/opendata/#/

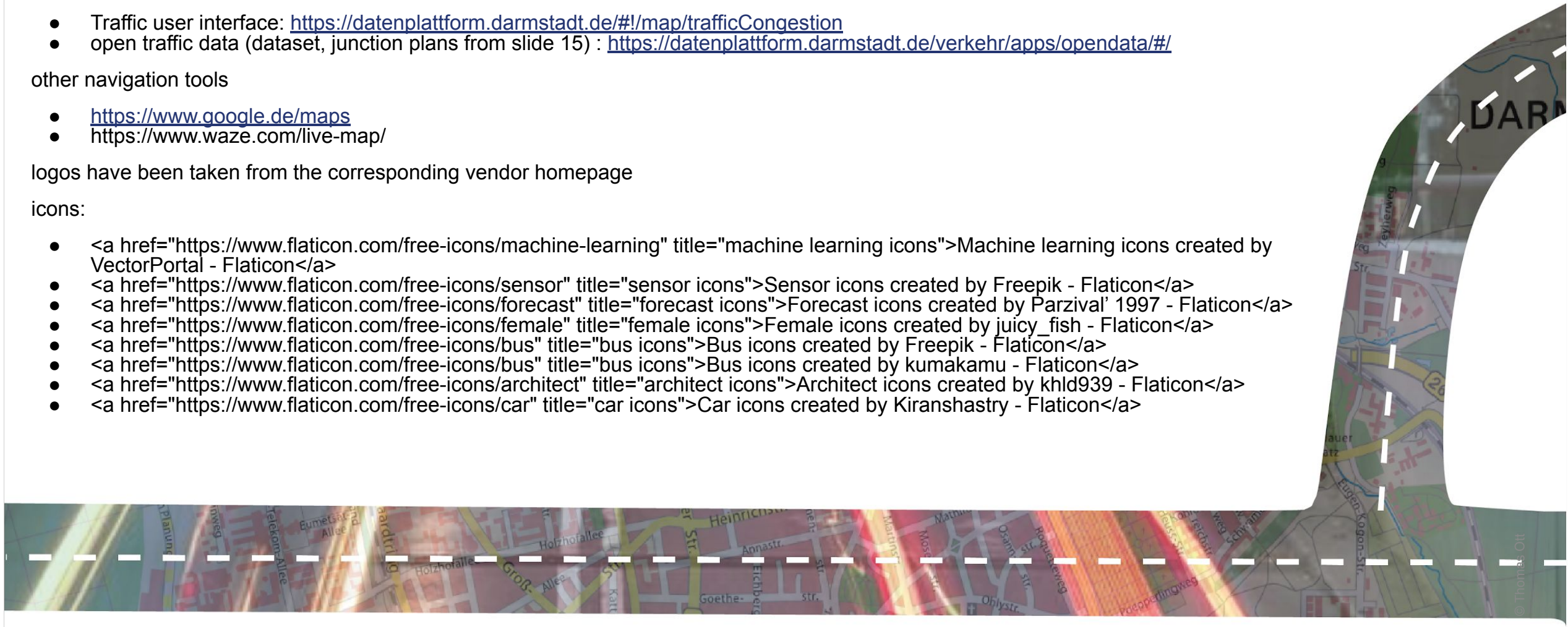
other navigation tools

- https://www.google.de/maps
- https://www.waze.com/live-map/

logos have been taken from the corresponding vendor homepage

icons:

- Machine learning icons created by VectorPortal - Flaticon
- Sensor icons created by Freepik - Flaticon
- Forecast icons created by Parzival' 1997 - Flaticon
- Female icons created by juicy_fish - Flaticon
- Bus icons created by Freepik - Flaticon
- Bus icons created by kumakamu - Flaticon
- Architect icons created by khld939 - Flaticon
- Car icons created by Kiranshastry - Flaticon

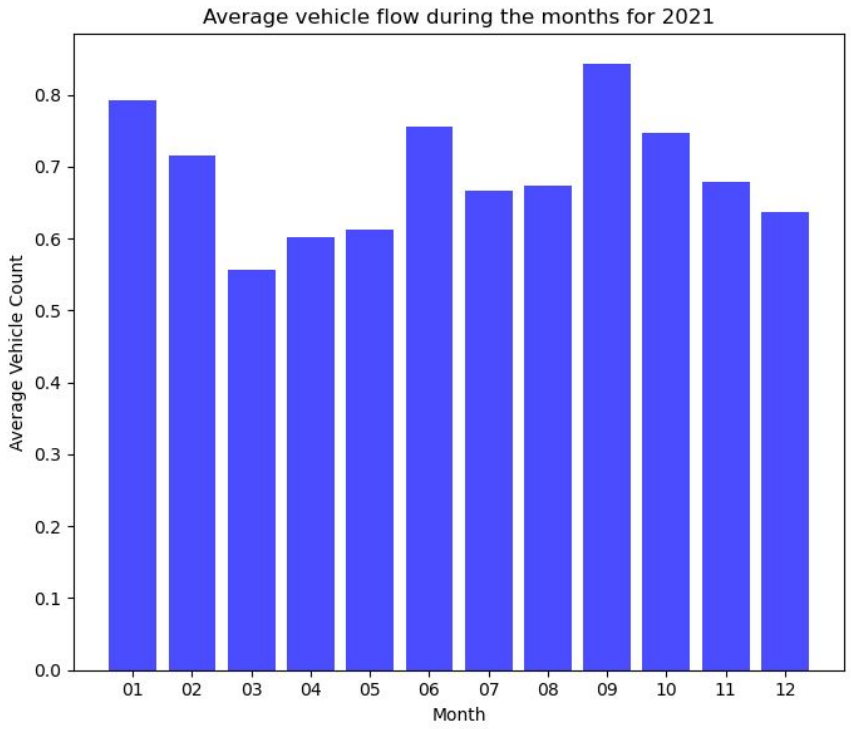


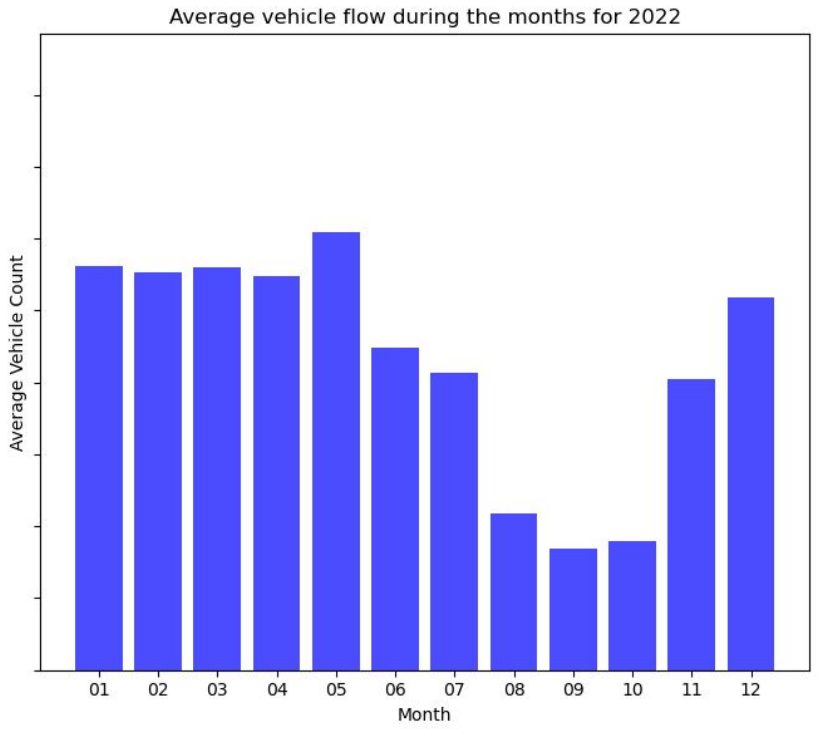
TECHNISCHE UNIVERSITÄT DARMSTADT

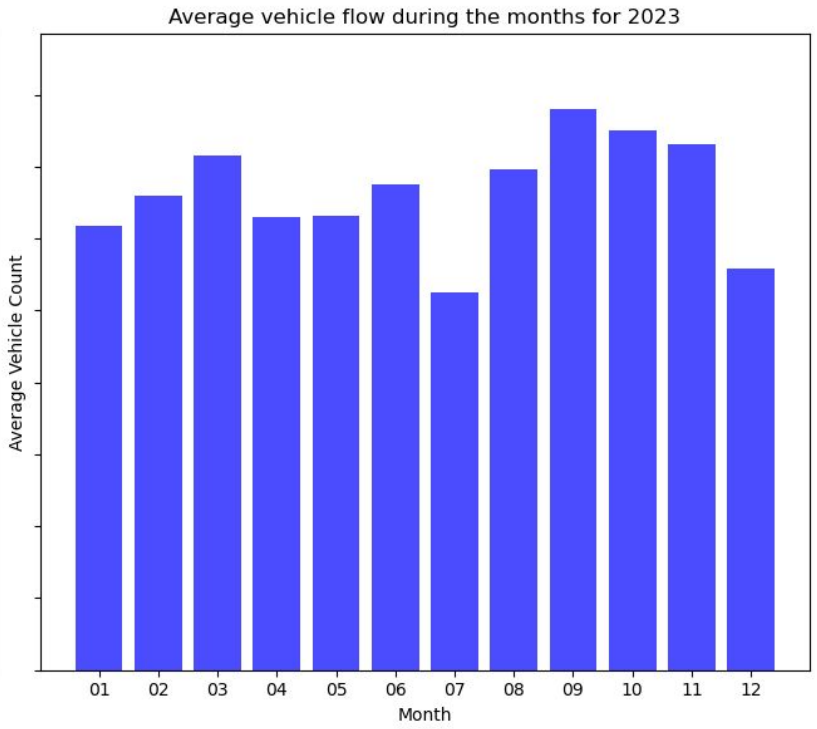
Backup slides

EDA plots

average vehicle flow per year







TECHNISCHE UNIVERSITÄT DARMSTADT

Average vehicle flow for different weekdays and different months

more detailed metrics for RandomForest, CatBoost and LightGBM

	MAE (Mean Absolute Error)	MAE as percentage	MSE (Mean Squared Error)	RMSE (Root Mean Squared Error)	R^2	SMAPE (Symmetric Mean Absolute Percentage Error)
Random Forest	3.834	16.76 %	60.10	7.75	0.94	102.21 %
CatBoost	4.10	17.93 %	60.27	7.76	0.94	102.72 %
LightGBM	4.069	17.98 %	63.28	7.95	0.94	102.72 %

- Mean absolute error (MAE): $\frac{1}{n}\sum_{j=1}^{n}|y^{(j)} - \hat{y}^{(j)}|$.
  same unit, greater errors not penalized more heavily
- Mean Sqaured Error (MSE): $\frac{1}{n}\sum_{j=1}^{n}(y^{(j)} - \hat{y}^{(j)})^2$
  greater errors penalized more, but unit is squared
- Root Mean Squared Error (RMSE)
  greater errors penalized more, same unit as y.
- R-Squared $R^2 = 1 - \frac{RSS}{TSS}$, RSS: sum of residual squares, TSS: total sum of squares.
  $\rightarrow$ how much of data variation can be explained by your model
  the closer RSS to 1, the better the residual error

$$\text{SMAPE} = \frac{100}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$

percentage difference between observed and predicted values.
symmetric (over-and underestimation treated equally), sensitive to extreme outliers, produces infinite values if actual and predicted value are zero

Metrics sources: Data Science 1 course, https://mlpills.dev/time-series/error-metrics-for-time-series-forecasting/, https://sefidian.com/2022/08/18/a-guide-on-regression-error-metrics-with-python-code/

Random Forest Model

TECHNISCHE UNIVERSITÄT DARMSTADT

- 3 Stages :
    - **Training Phase** : Each tree is trained on a random subset of the training data (if bootstrap sampling is used) and at each split, a random subset of features is considered.
    - **Prediction Phase** : Input features are passed through all the trees in the forest. Each tree, based on its learned patterns, makes a prediction about the target variable.
    - **Aggregation** : Final prediction is calculated by averaging the predictions from all individual trees.

Hyperparameters tuned :

- n_estimators : represents the number of trees in the forest
- max_depth : maximum depth of each tree (deeper tree may overfit)
- min_samples_split : minimum number of samples required to split an internal node
- min_samples_leaf : minimum number of samples required to be at a leaf node
- max_features : number of features to consider when looking for the best split (e.g., auto, sqrt, log2)
- bootstrap : True or False (If False, the entire dataset is used to build each tree)

Tuning was done by evaluation on the basis of metrics in slide 19

Prophet, Random Forest, LigthGBM, CatBoost

TECHNISCHE UNIVERSITÄT DARMSTADT

Prophet:
- additive time series forecasting model

Random Forest:
- collection of many decision trees whose predictions are combined for a single prediction
- each tree is trained independently by randomly selecting features and using a random subsampling (bootstrap) part of the dataset
- Randomness helps to make the model more robust than a single decision tree, and less likely to overfit on the training data

Boosting techniques train a single tree using a sequence of underfit trees. There are different numbers of boosting methods.

LightGBM (Light Gradient Boosting Machine):
- implements LightGBM algorithm
- Uses Gradient-based One-Side Sampling (GOSS) that helps find the most influential cuts
  - only use instances with large gradients and cut the small gradients when sampling the dataset
- Exclusive Feature Bundling (EFB). EFB reduces the number of features.
- Grows trees by leaves -> can result in imbalance (asymmetric) trees.

CatBoost: also based on gradient boosting
- optimized for categorical variables
- Uses Ordered Boosting and ordered TS (Target Statistics)
- minimal variance sampling (MVS) helps to maximize the accuracy of splits
- ordering principal stops target leakage
- grows trees by levels and results in balanced trees

https://medium.com/octave-john-keells-group/xgboost-light-gbm-and-catboost-a-comparison-of-decision-tree-algorithms-and-applications-to-a-f1d2d376d89c
https://colab.research.google.com/drive/1uCNP2h8mDb3cmiORc5b8mZ5bILFRIU6-?usp=sharing#scrollTo=pih-9yS6e5ga
https://medium.com/@veribilimi35/machine-learning-part-4-random-forest-gbm-xgboost-lightgbm-catboost-10e4e69eef33
https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/
https://research.facebook.com/blog/2017/02/prophet-forecasting-at-scale/