Michael Jantscher, BSc

Big Data Analysis for Road Accident Risk Prediction in Graz

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Information and Computer Engineering

submitted to

Graz University of Technology

Supervisor

Dipl.-Ing. Dr.techn. Roman Kern

Institute for Interactive Systems and Data Science
Head: Univ.-Prof. Dipl-Inf. Dr. Stefanie Lindstaedt

Graz, September 2019

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Abstract

Traffic accident prediction has been a hot research topic in the last decades. With the rise of Big Data, Machine Learning, Deep Learning and the real-time availability of traffic flow data, this research field becomes more and more interesting. In this thesis different data sources as traffic flow, weather, population and the crash data set from the city of Graz are collected over 3 years between 01.01.2015 and 31.12.2017. In this period 5416 accidents, which were recored by Austrian police officers, happened. Further these data sets are matched to two different spatial road networks. Beside feature engineering and the crash likelihood prediction also different imputation strategies are applied for missing values in the data sets. Especially missing value prediction for traffic flow measurements is a big topic. To tackle the imbalance class problem of crash and no-crash samples, an informative sampling strategy is applied. Once the inference model is trained, the crash likelihood for a given street link at a certain hour of the day can be estimated. Experiment results reveal the efficiency of the Gradient Boosting approach by incorporating with these data sources. Especially the different districts of Graz and street graph related features like centrality measurements and the number of road lanes play an important role. Against that, including traffic flow measurements as pointwise explanatory variables can not lead to a more accurate output accuracy.

Contents

List of Figures

1 Introduction

Every year, road accidents lead to a dramatic loss of numerous lives resulting from motorized vehicles. In the last decades researchers have paid increasing attention to determine factors that affect these, often deadly, incidents on the streets worldwide.

As stated in the work of F. L. Mannering and Bhat, 2014 and Lord and F. Mannering, 2010, many factors like environmental conditions, roadway geometrics or traffic volume seem to have a great impact on the frequency and also on the severity of crashes. Many of these data sources, like traffic volume measurements, are just available country-wide or region-wide because they are maintained by several institutions separately.

Driver specific data which would affect the crash likelihood as mentioned in the study of Howard et al., 2004, are at all hardly to figure out. Personal factors like driver distraction, sleepiness and health status can not be tackled but would have a great influence in the sense of crash risk prediction.

Recently, because of the rise of personal GPS trackers as smartphones, fitness bracelets and so on, human mobility data becomes more and more available also over national boarders. Also other data sources like weather data, and road link related features are provided and can be accessed in real-time. This access to different data sources in real-time will generate a high added-value in therms of accident risk prediction. Just think of an application on mobile devices for accident risk estimation which alerts an driver about high risky road segments on his planned route. Like a traffic jam map, they will be able to avoid traffic incidents more easily. Not only the data must be provided in real-time but also the inference model must handle the different explanatory variables extracted from the according sources and generate the prediction output in real-time. With the rise of machine learning techniques and Big Data analytic tools this use-case can be tackled more accurately today. F. L. Mannering and Bhat, 2014 summarizes the most common methodological approaches not only in terms of crash-frequency but also in the sense of crash-injury severity prediction. Despite the availability of different data sources in real-time, the problem of missing values has generate issues for researches for years. Especially missing information in dynamic traffic flow data, mainly due to detector failure and lossy communication systems (Asif et al., 2016), is still a big topic as mentioned in the work of Qu et al., 2009.

Problem Statement

To summarize, this thesis tackles the following issues:

- Collect and analyze heterogeneous data sets for the city of Graz including accident, traffic flow, weather and population data for crash likelihood estimation.
- Overcome missing values in the data sets by applying different imputation strategies. Especially imputation for the traffic flow data is a big topic.
- Apply different machine learning models for road accident prediction not only on the whole data set but also on subsets. Evaluate and compare these models on different evaluation metrics.

2 Related Work

2.1 Preliminary Work

2.1.1 Introduction

In a previous work a road accident severity prediction model based on the Austrian traffic data set was deployed. Each accident in this data set consists of several features which had been documented by Austrian police officers. The use-case of this work was to apply a model which can estimate the severity class of unlabeled samples based on this features.
Important features of the classification task therefore were figured out and went through a data preprocessing pipeline. Afterwards different classification algorithms like Neural Networks, Decision Trees and ensemble methods like Adaptive Boosting and Random Forest got applied and different accuracy measurements where tracked and compared.

2.1.2 Data analysis

For deeper insides histograms of features cited in Table 2.1 were plotted. To overcome features with missing values, the whole entry got deleted or, if their are too many unknown values, the whole feature was left out.
Features like the color of the vehicle, the speed limit and the first registration of a vehicle were completely left out because of too many missing values.
Others, like a certain type of a car trailer, does not seem to have a great impact because more than 99% of accident participants had not carried a trailer at all.

In case of the ″VerkehrsartGruppe″ feature only the categories: ″Bus″, ″Lkw 3,5-12t″, ″Lkw < 3,5t″, ″Lkw > 12t″, ″Pkw″, ″Fussgaenger″, ″Fahrrad″, ″Moped″ and ″Motorrad″ were considered.
For the target feature ″Verletzungsgrad″ the categories: ″Todeseintritt an der Unfallstelle″, ″Todeseintritt innerhalb von 30 Tagen″ and ″Todeseintritt nach mehr als 30 Tagen″ were combined to the class ″toedlich verletzt″ as shown in Figure 2.2. Entries classified as ″Suizid/Versuch″, ″Tod/Verletzung durch ploetzliche Erkrankung nicht durch Unfall″ and ″o.A.″ where completely left out.
One can see that the classes ″toedlich verletzt″ and ″schwer verletzt″ are, thank god, underrepresented which might have an negative impact for the classification task itself.
Also entries with unknown values for the feature ″Alter″ and ″Geschlecht″ were removed from the data set.
For other features like ″Airbag″, ″Helm″ and ″Gurt″ some classes were combined to a new one.

Feature	categorical	number categories	missing values
Jahr	yes	5	no
Monat	yes	12	no
Wochentag	yes	7	no
Stunde	yes	24	no
Tag	yes	31	no
Tempolimit	yes	16	yes
Gebiet	yes	2	no
UnfalltypOG10	yes	10	no
Bundesland	yes	9	no
StrZustand	yes	5	no
Niederschlag	yes	5	no
Nebel	yes	2	no
Wind	yes	2	no
Lichtverhaeltnisse	yes	4	no
Fahrbahndecke	yes	6	no
StrassenartGruppe1	yes	3	no
VerkehrsartGruppe	yes	14	yes
Anhaenger	yes	8	yes
Farbe	yes	19	yes
Erstzulassung	yes	11	yes
Leistung	yes	1	yes
Beteiligung	yes	2	no
Alter	no	-	yes
Geschlecht	yes	3	yes
Nationalitaet	yes	163	yes
Schulweg	yes	2	no
Alkoholisiert	yes	2	no
ErstzulassungPerson	yes	11	no
ErstzulassungFahrzeug	yes	11	yes
Airbag	yes	5	yes
Gurt	yes	5	yes
Helm	yes	5	yes
Verletzungsgrad	yes	6	yes
Koordinaten0	no	-	no
Koordinaten1	no	-	no

Table 2.1: Extracted Features for accident entry

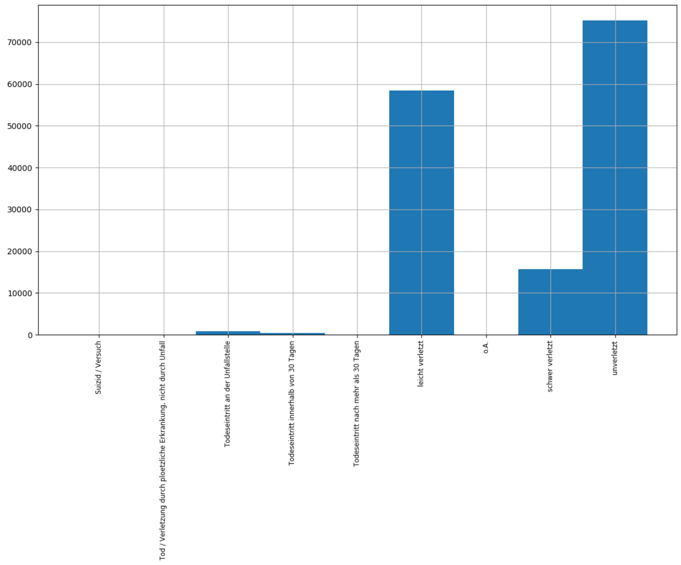


Figure 2.1: ″Verletzungsgrad″ before preprocessing.

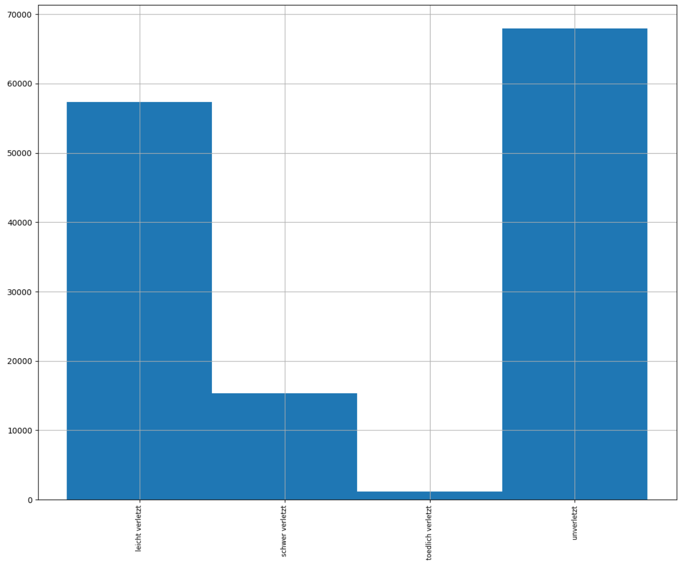


Figure 2.2: ″Verletzungsgrad″ after preprocessing.

Feature "Airbag"

Regarding the feature ″Airbag″, the classes ″Airbag nicht vorhanden″ and ″Airbag vorhanden″ were combined to the class ″Airbag nicht ausgeloest″ but only for four wheel vehicles. All other types where set to ″Airbag nicht vorhanden″.

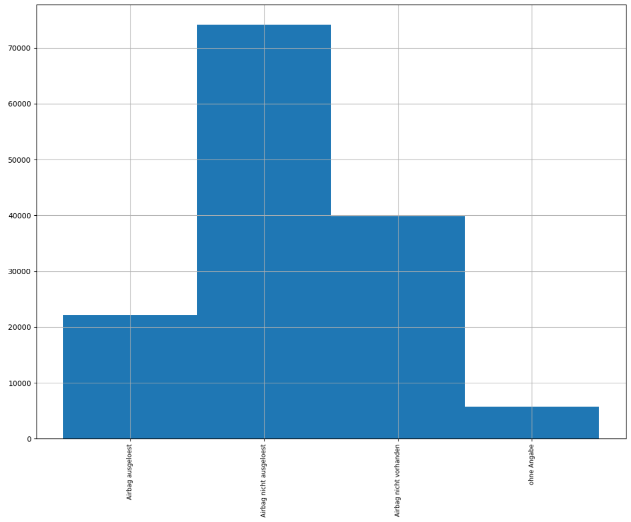


Figure 2.3: ″Airbag″ feature

Feature "Helm"

In case of the feature ″Helm″, for all four wheel vehicles the class ″Sturzhelm nicht vorhanden″ was chosen. For two wheel vehicles two different semantic versions of the word ″Helm″ namely ″Radhelm″ and ″Sturzhelm″ are used. To generalize that, all classes with ″Radhelm″ were casted to ″Sturzhelm″.

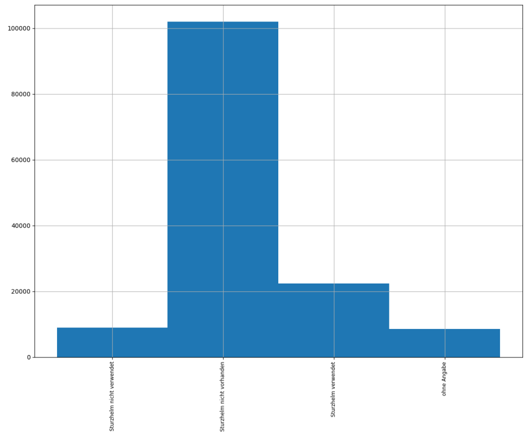


Figure 2.4: ”Helm” feature

Feature ”Gurt”

For the feature ”Gurt”, the class ”Sicherheitsgurt nicht vorhanden” was casted to the class ”Sicherheitsgurt nicht verwendet” for all four wheel vehicles. For all other types this feature was set to ”Sicherheitsgurt nicht vorhanden”

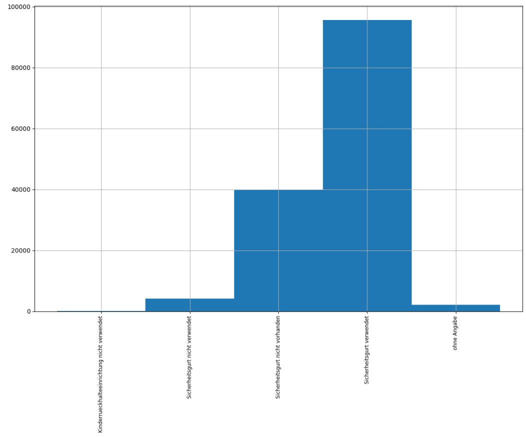


Figure 2.5: ”Gurt” feature

2.1.3 Feature transformation

Because most machine learning algorithms can not handle text and categorical features directly, all categorical features were one-hot encoded. This can be done via the pandas DataFrame function *get_dummies*. After preprocessing all categorical attributes, one entry in the data set now consists of 166 training features.
The only numerical feature is the age of the accident participants. To overcome bad performance of the different machine learning algorithms, one has to scale this feature to a given range.

2.1.4 Training Validation and Test Set

The whole preprocessed data set was split into a training (80%) and a test (20%) set according to the distribution of the target classes. Afterwards several experiments were done on the remaining samples in the training set and evaluated on the test set. Because of under represented target classes, new samples were generated for these classes. This can be done by oversampling the minority classes during the training process via resampling with replacement. Now the target classes are equally distributed and also the training set size increased.
For some of the following Machine Leaning models also a extra validation set has to be considered. The split for this set is done like the one for the test set, where the validation set takes 33% of the training set.

2.1.5 Classification Models

In this section the used classification models are going to be described concerning:

- reason of choice
- hyper parameter evaluation
- evaluation

Neural Network model - Multilayer Perceptron (MLP)

Because MLPs are good in learning nonlinearities given a specific data set it is one of the state of the art classifiers. For the classification task itself the *MLPClassifier* [1] from the scikit library was used. This class implements a multilayer perceptron algorithm that trains using Backpropagation. Currently, the MLPClassifier supports only the Cross-Entropy loss function which is a common choice for multi-class classification tasks.
The *RandomizedSearchCV* [2] class from the scikit library was used for hyper parameter evaluation. For the original data set the estimated hyper parameter values represented in Table 2.2 were evaluated and the accuracy of the cross-validation was 74.2%

hyper parameter	**chosen value**
activation	logistic
hidden_layer_sizes	(100, 200)
alpha	1e-4
early_stopping	True
solver	lbfgs
max_iter	50
learning_rate	constant
tol	1e-5

Table 2.2: List of hyper parameters for randomized grid search for the MLP

For the up-sampled data set the hyper parameter values represented in Table 2.3 were evaluated and the mean score of the cross-validation was 86.3%

[1] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html (Accessed on: 2019-12-15)

[2] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html?highlight=randomizedsearchcv#sklearn.model_selection.RandomizedSearchCV (Accessed on: 2019-12-15)

hyper parameter	chosen value
activation	relu
hidden_layer_sizes	(250, 500)
alpha	1e-4
early_stopping	True
solver	adam
max_iter	50
learning_rate	constant
tol	1e-3

Table 2.3: List of hyper parameters for randomized grid search for the MLP

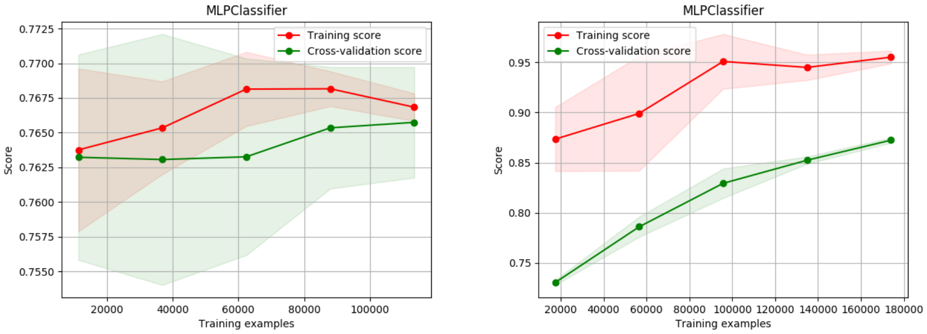


Figure 2.6: left: learning curve on the original data set, right: learning curve on the upsampled data set

Ordinal Classification

For the classification of accident severities their exists a natural order of the classes. Starting with 0- ”unverletzt” ending with 3-”toedlich verletzt”.
For this ordinal classification the class ”LogisticIT” [3] provided by the mord library was used. Again *RandomizedSearchCV* was used for hyper parameter evaluation. The following parameters were evaluated for the original training data set:

[3] `https://pythonhosted.org/mord/` (Accessed on: 2019-12-15)

hyper parameter	chosen value
alpha	1.0
max_iter	664

Table 2.4: List of hyper parameters for randomized grid search for the LogisticIT

And for the up-sampled data set the evaluated parameters are:

hyper parameter	chosen value
alpha	1e-3
max_iter	407

Table 2.5: List of hyper parameters for randomized grid search for the LogisticIT

The accuracy for the original data set is 75.5% whereas for the up-sampled set is is 55.7%

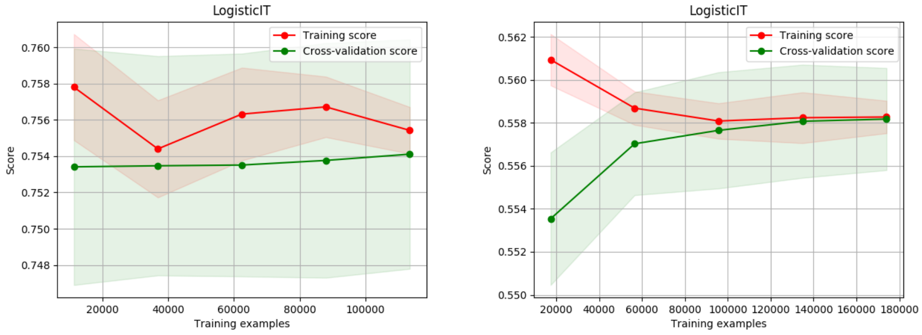


Figure 2.7: left: learning curve on the original data set, right: learning curve on the up-sampled data set

Random Forest Classifier

Decision Trees are great classifiers on imbalanced data sets because their hierarchical structure allows them to learn signals/decisions from different classes. Because ensemble methods like Random Forests almost always outperforms single Decision Trees, only the Random Forest was considered.

For this task, the *RandomForestClassifier* [4] from the scikit library was used. The hyper parameter search offered the following results for the original data set with an accuracy of 76.8%:

hyper parameter	**chosen value**
bootstrap	True
min_samples_leaf	253
n_estimators	20
min_samples_split	246
criterion	entropy
max_features	50
max_depth	None

Table 2.6: List of hyper parameters for randomized grid search for the Random Forest

And for the up-sampled data set with an accuracy of 62.7%:

hyper parameter	**chosen value**
bootstrap	True
min_samples_leaf	186
n_estimators	50
min_samples_split	2866
criterion	gini
max_features	100
max_depth	30

Table 2.7: List of hyper parameters for randomized grid search for the Random Forest

[4] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html?highlight=randomforestclassifier#sklearn.ensemble.RandomForestClassifier` (Accessed on: 2019-12-15)

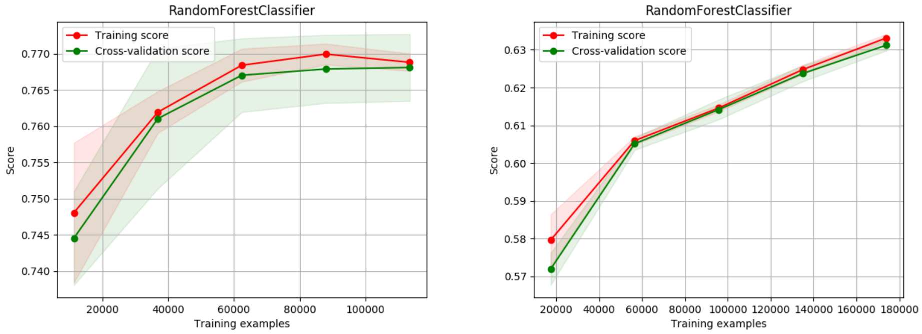


Figure 2.8: left: learning curve on the original data set, right: learning curve on the up-sampled data set

AdaBoost

Another ensemble method which was evaluated was the so called AdaBoost Classifier [5], again from the scikit library. It is a meta-estimator that begins by fitting a classifier on the original data set and then fits additional copies of the classifier on the same data set but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. The resulting parameters of the hyper parameter search based on the original data set (mean CV score: 76.8%) and for the up-sampled data set (mean CV score: 61%) offers the same settings:

[5] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html?highlight=adaboost#sklearn.ensemble.AdaBoostClassifier` (Accessed on: 2019-12-15)

hyper parameter	**chosen value**
learning_rate	0.1
n_estimators	500
base_estimator	default Decision Tree with gini criterion and a *max_depth* of 1
algorithm	SAMME.R
max_features	100

Table 2.8: List of hyper parameters for randomized grid search for the AdaBoost classifier

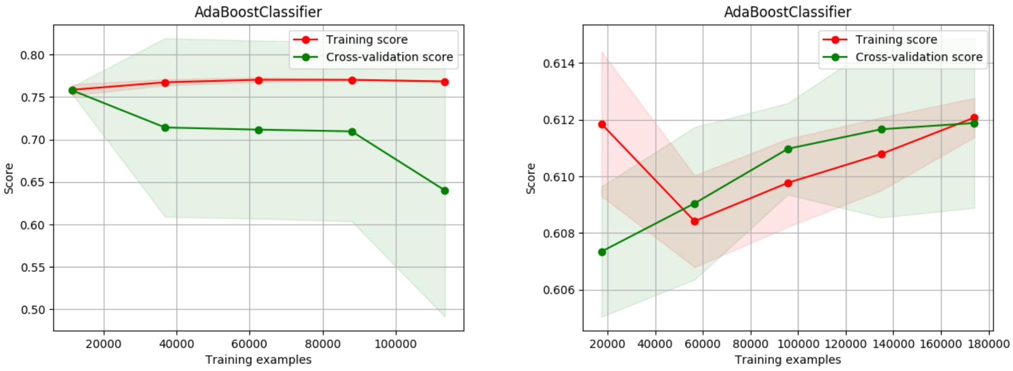


Figure 2.9: left: learning curve on the original data set, right: learning curve on the upsampled data set

GradientBoosting

The Gradient Boosting Classifier builds an additive model and allows the optimization of arbitrary differentiable loss functions. In each stage *n_classes* regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function. The validation score of the training based on the original data set is 77% and on the up-sampled one 87.7% Hyper parameters for original data set:

hyper parameter	**chosen value**
warm_start	True
learning_rate	0.9
n_estimators	500
max_leaf_nodes	None
criterion	entropy
max_features	None
max_depth	1

Table 2.9: List of hyper parameters for randomized grid search for the Gradient Boosting

And for the up-sampled set:

hyper parameter	**chosen value**
warm_start	False
learning_rate	0.9
n_estimators	1000
max_leaf_nodes	20
criterion	entropy
max_features	100
max_depth	6

Table 2.10: List of hyper parameters for randomized grid search for the Gradient Boosting

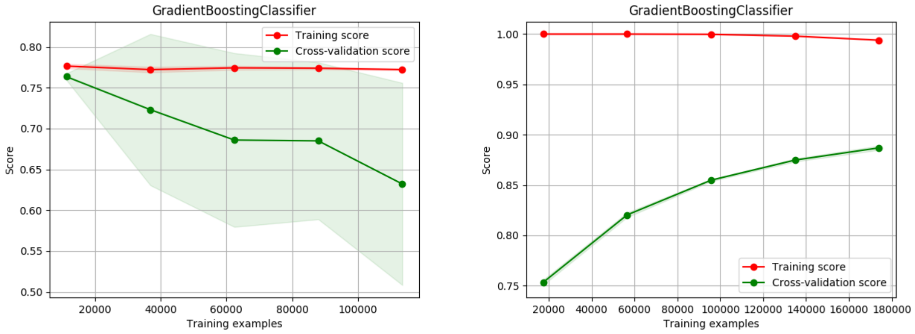


Figure 2.10: left: learning curve on the original data set, right: learning curve on the upsampled data set

2.1.6 Model evaluation on the test set

For evaluation purpose, these five models, as described in Chapter 2.1.5, got trained on the training and validation set and afterwards tested on the test set. For deeper understanding not just the accuracy on the test set but also the confusion matrices on the training data (training and validation set) and test data were plotted for each trained model.

MLP evaluation

Figure 2.11 and 2.12 show the according confusion matrices. One can see that the MLP overfits the training data so that this model don't generalize well. Especially the accuracy for the two under represented classes "schwer verletzt" and "toedlich verletzt" is very bad. So maybe the *RandomOverSampler* does not really work well and therefore a different sampling strategy would be preferable.

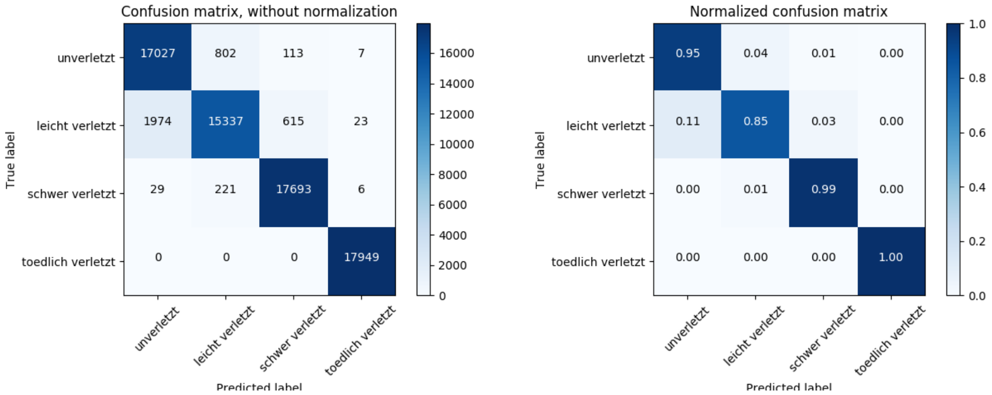


Figure 2.11: Confusion Matrix training set

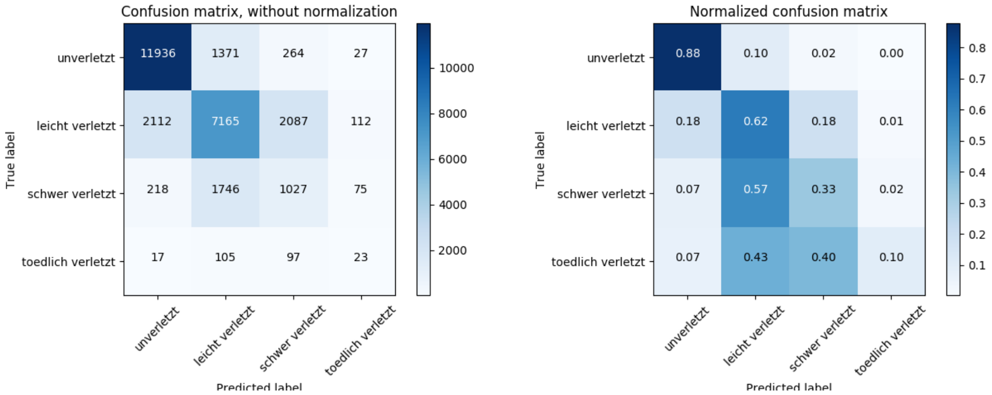


Figure 2.12: Confusion Matrix test set and an accuracy of 70.9%

Ordinal Classification evaluation

Compared to the MLP model this model seems to be more biased. But it reaches a better accuracy for the two under represented classes as visualized in Figure 2.13 and 2.14.

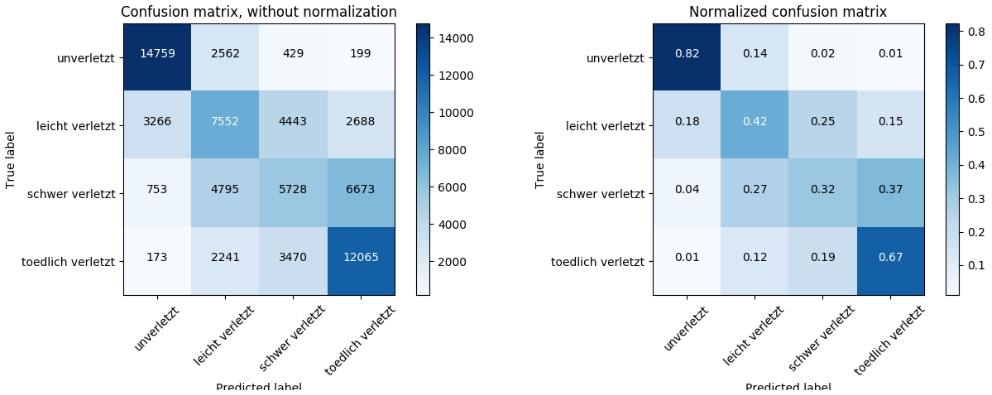


Figure 2.13: Confusion Matrix training set

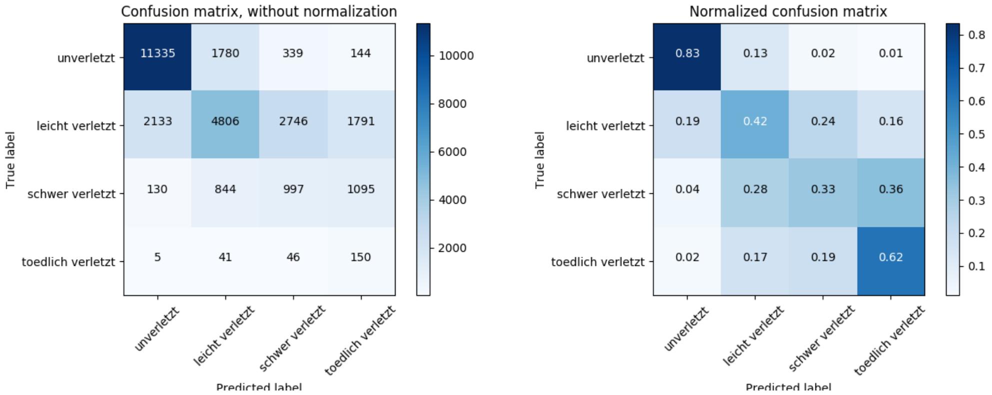


Figure 2.14: Confusion Matrix test set and an accuracy of 60.9%

Random Forest evaluation

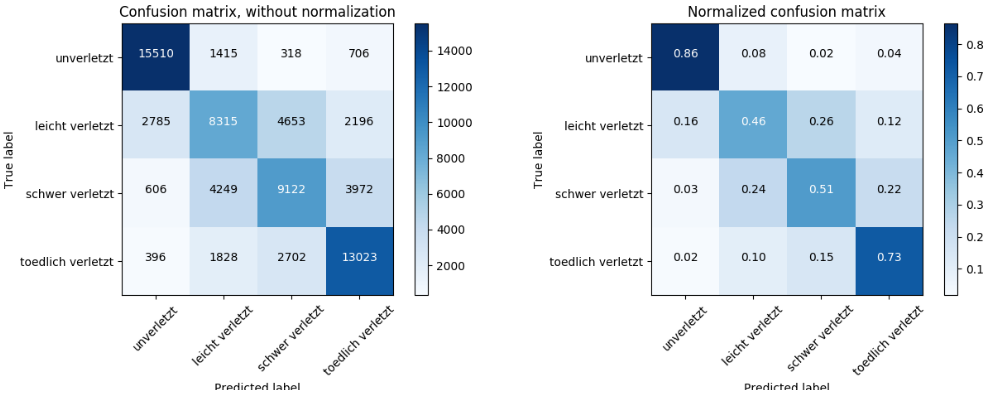


Figure 2.15: Confusion Matrix training set

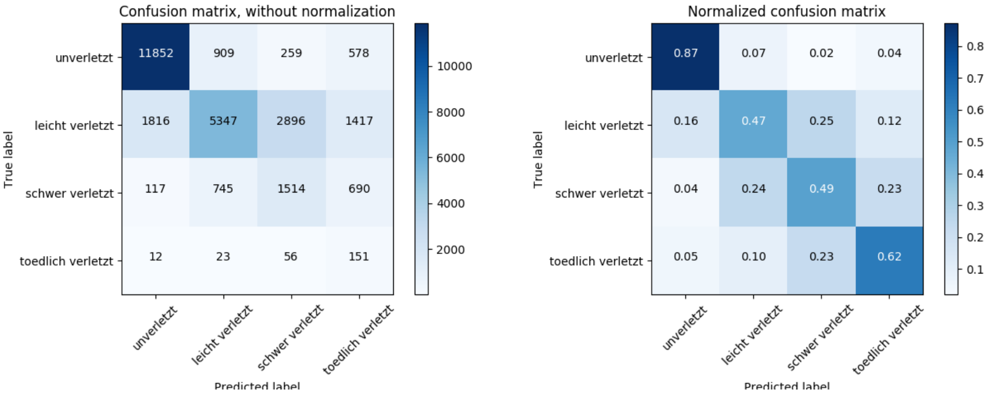
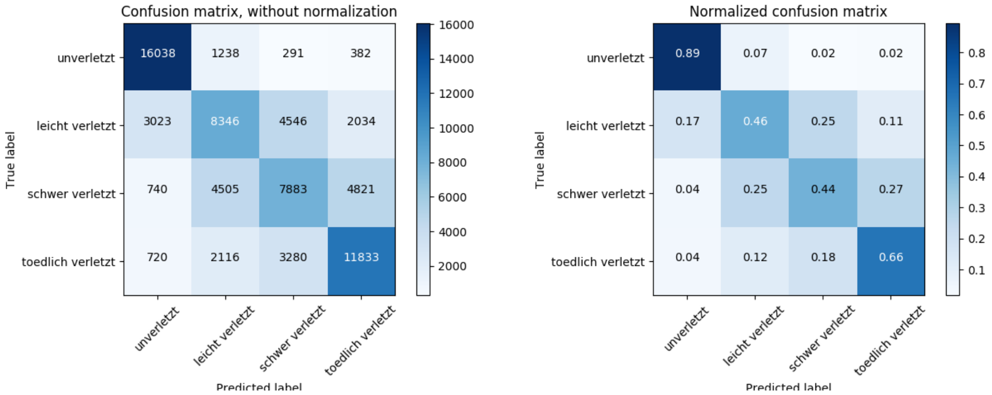


Figure 2.16: Confusion Matrix test set and an accuracy of 66.4%

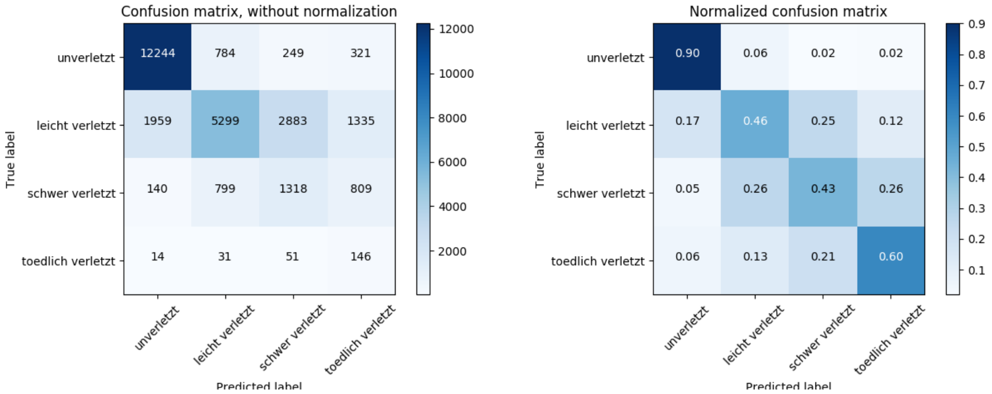
AdaBoost evaluation



	unverletzt	leicht verletzt	schwer verletzt	toedlich verletzt
unverletzt	16038	1238	291	382
leicht verletzt	3023	8346	4546	2034
schwer verletzt	740	4505	7883	4821
toedlich verletzt	720	2116	3280	11833

	unverletzt	leicht verletzt	schwer verletzt	toedlich verletzt
unverletzt	0.89	0.07	0.02	0.02
leicht verletzt	0.17	0.46	0.25	0.11
schwer verletzt	0.04	0.25	0.44	0.27
toedlich verletzt	0.04	0.12	0.18	0.66

Figure 2.17: Confusion Matrix training set



	unverletzt	leicht verletzt	schwer verletzt	toedlich verletzt
unverletzt	12244	784	249	321
leicht verletzt	1959	5299	2883	1335
schwer verletzt	140	799	1318	809
toedlich verletzt	14	31	51	146

	unverletzt	leicht verletzt	schwer verletzt	toedlich verletzt
unverletzt	0.90	0.06	0.02	0.02
leicht verletzt	0.17	0.46	0.25	0.12
schwer verletzt	0.05	0.26	0.43	0.26
toedlich verletzt	0.06	0.13	0.21	0.60

Figure 2.18: Confusion Matrix test set and an accuracy of 66.9%

GradientBoosting evaluation

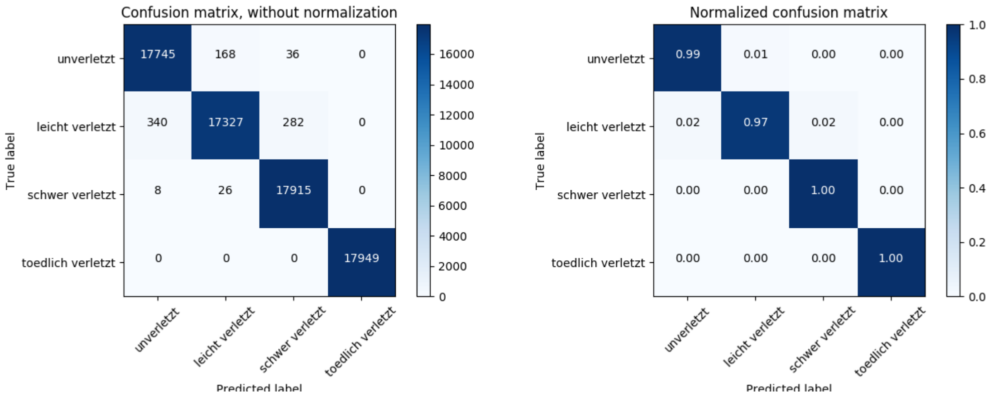


Figure 2.19: Confusion Matrix training set

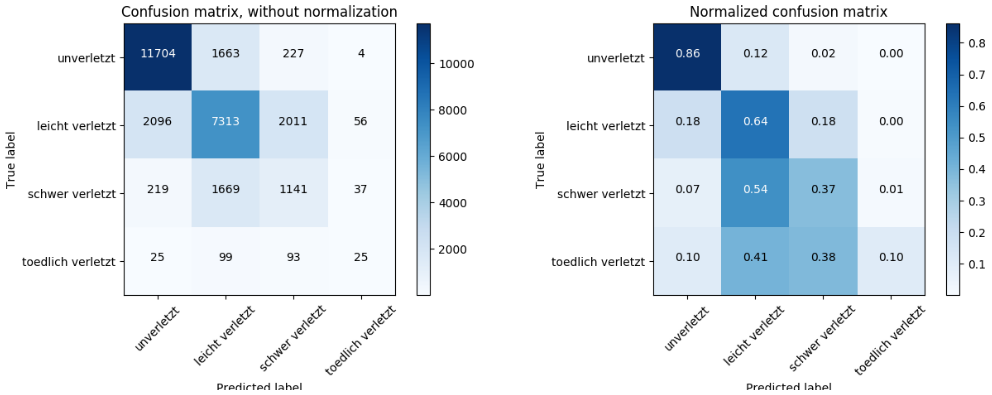


Figure 2.20: Confusion Matrix test set and an accuracy of 71.1%

Stacked Classifier

Because of the fact that different models perform better on different classes, a stacked classifier was trained. Therefore all the previous described models were trained only on the training set.
Afterwards the predictions from the validation set of the different models got stacked and feet into a meta learner which is, in this case, a MLP with the parameters mentioned in Table 2.11

hyper parameter	chosen value
activation	relu
hidden_layer_sizes	(100, 200)
alpha	0.01
early_stopping	True
solver	lbfgs
max_iter	200
learning_rate	constant
tol	1e-05

Table 2.11: List of hyper parameters for randomized grid search for the MLP meta learner

The hyper parameters where evaluated again with the *RandomizedSearchCV* but in this case only on the validation data set.
The base estimators are:

- MLP
- Gradient Boosting
- Random Forest

The according confusion matrices on the validation and test set look like:

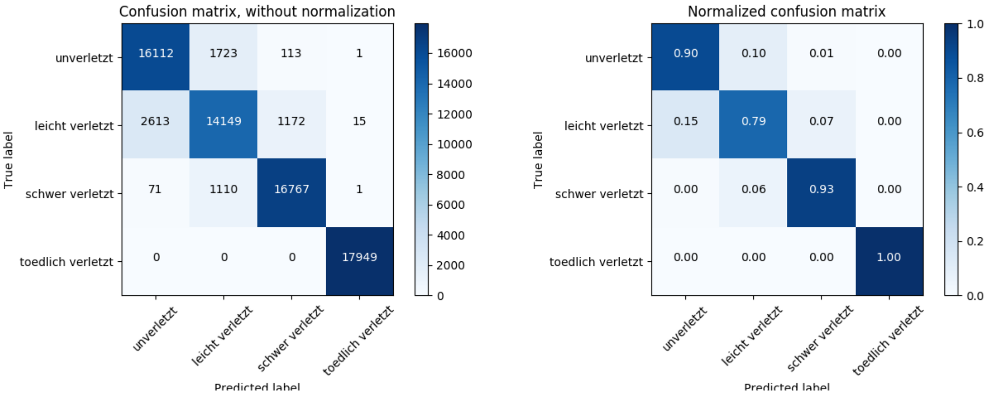


Figure 2.21: Confusion Matrix validation set and an accuracy of 90.5%

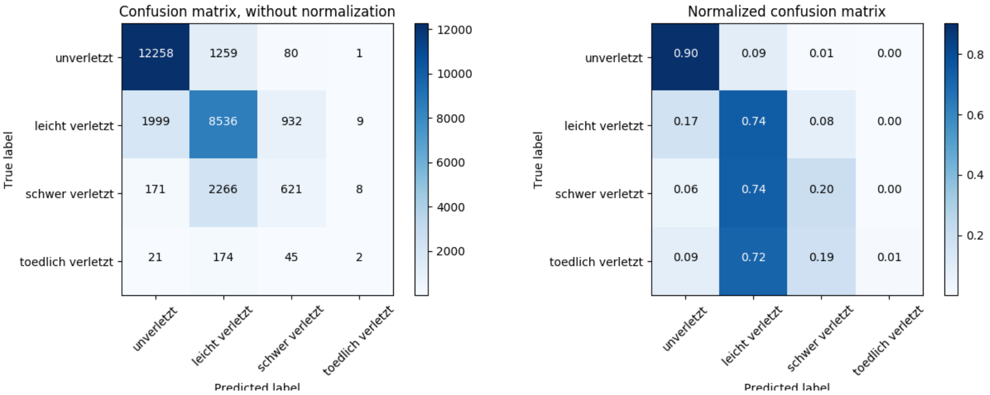


Figure 2.22: Confusion Matrix test set and an accuracy of 76%

2.1.7 Feature Importance

One can also evaluate which features from the data set are relevant for the classification task. Therefore the Random Forest classifier provides the member variable $feature\_importances\_$ which lists all features with their importances. The sum over all values is equal to 1. The top results are shown in Table 2.12

feature	importance value
Airbag nicht ausgeloest	0.229
Unfaelle mit nur einem Beteiligten	0.161
Sicherheitsgurt verwendet	0.083
Alter	0.064
Ortsgebiet	0.06
Freiland	0.057
Airbag ausgeloest	0.054
Unfaelle im Begegnungsverkehr	0.04
Sicherheitsgurt nicht verwendet	0.034
Airbag nicht vorhanden	0.032
ausgeloest	0.01
weiblich	0.01

Table 2.12: Feature importances

The feature "Airbag nicht ausgeloest" seems to have the greatest impact for the classification. Also the age and the accident type are important features, whereas features like weather conditions (rainy or foggy weather) do not have an impact at all. The problem by this estimation is that only the importance of a single feature is tracked but maybe there are also combinations of multiples which influences the classification too.

2.1.8 Conclusion

The result shows that the underrepresented target classes "schwer verletzt" and "toedlich verletzt" do not get predicted well with the different trained classifiers. One should think about an better up-sampling strategy for these

underrepresented classes. For training imbalanced classes, Random Forest and also the Ordinal Classification model seem to be the best by predicting these classes.
Another problem is that the features may not be very representative. So features like the speed of a vehicle at the moment of the crash would have a much greater impact for the different classes of injury severities.

2.2 State of the Art

Because of the urbanization around the world, the number of traffic accidents and especially the number of serious accidents has rapidly grown in the last decades. The ability to analyze and further to predict traffic accidents become a great research field. Since the rise of Big Data, Deep Learning and the rapid development of data collection techniques this field becomes more and more interesting not only for traffic safety institutions but also for data scientists. Further, because of the access to urban specific data sets like accident records, rainfall and traffic flow data the prediction of traffic accidents becomes more realistic.
The related work is split into several research fields as discussed in the following sections.

2.2.1 Road accident prediction

To gain some basic knowledge about the key factors and explanatory variables which affects the likelihood of crash accidents, Lord and F. Mannering, 2010 reviews common concepts and methodological approaches. It is stated that researchers have framed their analytical approaches in a way that spatial and temporal elements associated with crashes are handled as explanatory variables in their prediction models. This is because of the fact that detailed driving data (acceleration, braking and steering information but also driver related response) and also near-crash events are hardly to figure out or are typically not available due to privacy issues. This is a great drawback when framing this problem. Also many minor crashes might not be recorded which again leads to a loss of important information. For crashes with car

damage only, participants which are involved in the accident in Austria, do not have to inform the police [6]. So therefore no accident record will be generated.
Another issue comes along with time-varying explanatory variables and their right aggregations. For example, one is modeling the number of crashes per month and precipitation is one of the explanatory variables. The distribution of this variable over the month highly influences the number of crashes but aggregated precipitation values in an large interval like one month results in a large loss of information. Because of this information loss in time-varying explanatory variables, data are often considered in smaller time intervals.
Not only time-varying but also spatial explanatory variables have an great impact on the cause of traffic accidents. Therefore F. L. Mannering and Bhat, 2014 cites that all data (temporal and spatial) might have unobserved factors that might influence the risk of crashes. Ignoring these spatial and temporal correlations will result in inefficient and inconsistent parameter estimates. F. L. Mannering and Bhat, 2014 also summarizes, that many factors affecting the frequency and severity of crashes are not observable or impossible to collect. These unobserved explanatory variables, also referred as unobserved heterogeneity, might be correlated with observed factors. As a result biased parameters will be estimated and incorrect inference results will be drawn. For example, a statistical model is considered with age as an explanatory variable. This variable correlates, in the sense of an crash-injury severity prediction model, with many unobserved factors that affects the crash-injury severity such as physical health, susceptibility of bones to breakage, body positioning at the time of crash, reaction times that may mitigate the severity of the crash, and so on. Considering only the age as an explanatory variable, age acts as a proxy variable for many underlying factors which might vary significantly across the same age value or same age category.

The main work in this thesis is to predict the accident probability for a given street segment under specific conditions. Therefore the most interesting papers and previous works are in the field of classification models for traffic accidents. Because of the fact that the availability and accessibility of data sources strongly differs between countries the most scientific papers are in

[6] https://www.help.gv.at/Portal.Node/hlpd/public/content/99/Seite.992438.html (Accessed on: 2019-10-28)

the form of case studies.

Ren et al., 2018 and Q. Chen et al., 2016 come with similar approaches for real-time traffic accident prediction. Both discretize traffic accident data in space and time. Ren et al., 2018 processes the crash data in one hour interval and a spatial resolution dimension of 1000m x 1000m uniform grid cells. The problem which arises by spatial discretization is that additional data sets like a Road Network Graph seems to loose effectiveness because the direct influence of e.g. speed value of a specific road lane, lane width or road radius is not given anymore.

Ren et al., 2018 main approach is to find a correlation pattern between the accident frequency and the discretized spatial grid cells. So additional data sources such as traffic flow, weather data and road characteristic data sets, which maybe significant for crash risk prediction, are not considered. Q. Chen et al., 2016 uses traffic flow data as an additional explanatory variable for its Stacked denoise Autoencoder (SdAE) to find correlation between this flow data set and the spatial and temporal information of the crash records. The inference model outputs the accident risk level in a given grid cell at a specific hour of the day. But as a conclusion Q. Chen et al., 2016 also states that because of the complexity of traffic accidents, mobility data are not enough to construct a satisfactory model for the prediction of risks.

Yuan et al., 2017 comes with a case study in the state of Iowa where they analyzed traffic accidents between 2006 and 2013 for traffic accident forecasting predictions. They incorporated spatial structure of the road network as well as hourly weather data including high resolution rainfall data, Annual Average Daily Traffic (AADT) and demographic data into the predictive models. For construction of negative samples a informative sampling approach was proposed. The overall data set contains about 3 times negative samples than positive samples where negative samples are generated very close and far away from positive samples. For classification, they applied Support Vector Machines (SVM), Random Forests (RF) and Deep Neural Networks (DNN). Regarding performance they stated that the DNN and RF approach achieved an accuracy and AUC of about 0.95. Compared to several other works like Chong, Abraham, and Paprzycki, 2005 or Jie Sun and Jian Sun, 2015 where they achieved an accuracy between 60% and 70% this increase just because of including the spatial road network graph seems a bit strange. This might be because of the fact that the sampling of negative examples has a great impact of the accuracy and the precision/recall metric.

So when sampling negative values far from the positive, the results seem far better than sampling from distributions very close to positive samples. This is a problem in this research field at all, because there is no general rule how to handle this imbalance issue regarding crash and no-crash events. Therefore accuracy analysis between different studies are hardly to compare. The most common approach at the data level is a synthetic minority oversampling strategy (SMOTE) as applied in Yuan et al., 2017 or Ke et al., 2019. Ke et al., 2019 matches 10 non-crash samples for each crash samples with predefined matching rules:

- The location of non-crash events should be the same as crash events.
- The within day time of a non-crash event should be the same as a crash event but in a different day.
- The non-crash event should be of the same day type (weekday or weekend) as the crash event.

This matching rule seems to be much stricter as the matching rule as stated in Yuan et al., 2017. Ke et al., 2019 also comes up with a solution at the algorithmic level called cost-sensitive learning. This learning technique simply adjusts the objective function by penalizing the misclassification of a minority class sample (crash events in this case) much stronger than a majority class sample. Another scientific works like Anderson, 2009 focus on traffic accidents hot spot detection. It can be seen a bit like the preprocessing step as stated in Q. Chen et al., 2016 and Ren et al., 2018. In this special case the kernel of the hot spot detection process divides the entire study area into a given number of cells whereas in Q. Chen et al., 2016 and Ren et al., 2018 these cell discretization is done manually by fixed values. But again multiple street segments will be merged and road specific data get lost.

2.2.2 Missing value imputation techniques

Real-time traffic records from loop detectors are not ready to be processed because they contain a lot of missing data samples. Because missing values of a specific detection loop often occur in whole series one has to overcome this missing value problem by some imputation process. These missing values can be lead back due to temporary power or communication failures.

Many research efforts have been undertaken in therms of estimating missing traffic flow values and therefore many imputation methods have been proposed. Ke et al., 2019 offers some review of state of the art imputation techniques not only for missing traffic volume data but also for other related areas like road network data sets and in the area of traffic safety. The chosen imputation technique strongly depends on the pattern or distribution of missing values in the specific data set. Therefore Little and Rubin, 2019 comes up with a more general classification of missing patterns:

- Missing Completely At Random (MCAR)
- Missing At Random (MAR)
- Not Missing At Random (NMAR)

MCAR and MAR problems can be addressed by some universal imputation strategies while the NMAR problem can hardly be solved because of countless possibilities in the distribution of missing values. Tan et al., 2013 and Qu et al., 2009 comes up with a categorization of the different methods of imputation:

- Prediction based methods
- Interpolation based methods
- Statistical learning based methods

Shang et al., 2018 summarizes these main concepts and cites different algorithms and solving strategies for each of these imputation categories. Imputing missing values by an interpolation based method aims to estimate these values by including spatial and temporal relations. Therefore several transportation studies paid attention on approaches where missing values are estimated by choosing an multivariate approach in terms of spatial relationships. Considering that loop detectors have spatial and temporal dependencies, this method has an advantage over the other imputation methods because it takes observed neighbouring data, in space and time, correlated with these missing values into account.
Bae et al., 2018 aims to extend the spatial kriging approach to a multivariate imputation framework by including secondary data sources. Haworth and Cheng, 2012 developed a non-parametric imputation approach based on K-nearest neighbour which has been trained and tested on 5 minute travel time data in the London metropolitan area. Henrickson, Zou, and Wang,

2015 comes up with a multiple imputation approach called Multiple Imputation by Chained Equations (MICE) introduced by Rubin, 2004. MICE is a multiple imputation approach where a multivariate inference model is defined separately for each variable with missing data. In the sense of traffic flow data a variable represents a specific loop detector. Henrickson, Zou, and Wang, 2015 applied this imputation strategy to a loop detector volume data set collected on Interstate 5 in Washington State. The outcome of this study is that MICE outperforms elementary pairwise regression and offers reliable estimates even when a larger time period of missing values appear.

3 Background

In this chapter the most important tools used for prediction, imputation and visualization in this work are covered.

3.1 Machine Learning Preliminaries

3.1.1 Decision Trees and Random Forests

Decision Trees (DT) and further Random Forests (RF) are versatile machine learning algorithms which can handle classification and also regression tasks (Géron, 2017). Decision Trees are the basic components of Random Forests which are still one of the most powerful algorithms today. One of the advantages of decision trees is that, against to other machine learning algorithms, no feature scaling is necessary and continuous and categorical variable can be handled at the same time.
A Random Forest is a ensemble of Decision Trees for the purpose of introducing some extra randomness when growing trees. As a result the tree diversity rises, which results in a decreasing variance and an increasing accuracy especially on the validation set.
Another great advantage of Random Forests is the ability to measure the relative importance for each input feature. The Scikit library keeps track of these measurement by observing how much the impurity decreases on average for a specific feature over all trees in the ensemble.

3.1.2 XGBoost

XGBoost is a scalable end-to-end tree boosting algorithm introduced by a research project at the University of Washington (T. Chen and Guestrin, 2016). One main difference between Random Forests and the XGBoost approach is that this boosting algorithm trains each DT sequentially, each trying to correct its predecessor. Again there are several boosting strategies, however XGBoost follows the Gradient Tree Boosting approach. The tree boosting algorithm follows in general the existing literature in gradient boosting as stated in Friedman, 2001 only with small adaption in the regularization therm. A deeper insight in this strategy can be viewed in T. Chen and Guestrin, 2016.

3.2 Graph Theory

In this section all relevant graph definitions will be explained inspired by a road link network. Also different applied graph algorithms and measurements are stated. At the end required python packages which are able to create, manipulate and study structures of graphs are mentioned.

Directed Multigraph

A Directed Multigraph is a graph with directed links where multiple links can have the same start and end node. In therms of a street network graph, street links with a practicability in both direction are represented in this way.

Line graph

A line graph $L(G)$ of a given graph G consists of nodes generated from each edge in G and edges iff two edges in G share a node. For a directed graph

[1] `https://wntr.readthedocs.io/en/latest/networkxgraph.html` (Accessed on: 2019-11-29)

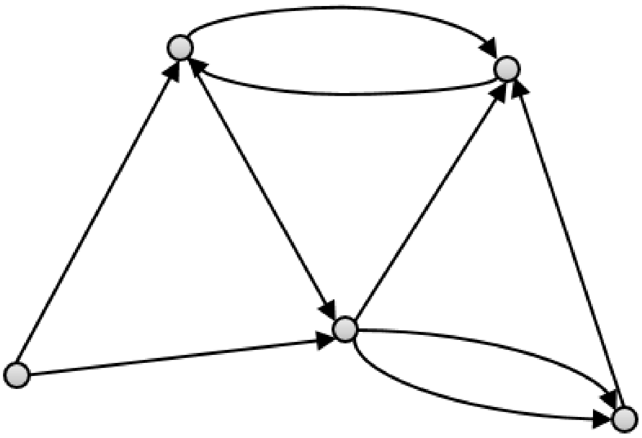


Figure 3.1: Example of a directed multigraph[1]

G, the directed line graph $L(G)$ consists of nodes generated from the set of edges of G. Edges are generated in a way that if a and b are two nodes in $L(G)$, then (a, b) is an edge in $L(G)$ iff the tail of a matches the head of b (Gross and Yellen, 2005).
This conversion is further necessary for analysis purpose like PageRank and centrality calculation of street links.

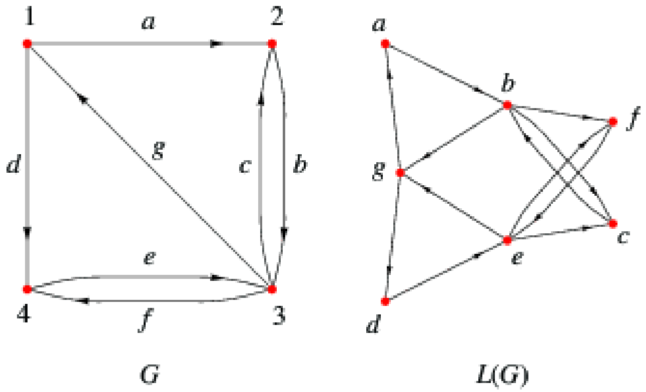


Figure 3.2: Example of a line graph generated from a directed graph[2]

[2] `http://mathworld.wolfram.com/LineGraph.html` (Accessed on: 2019-11-29)

PageRank

The PageRank algorithm (Page et al., 1999 and Langville and Meyer, 2005) generates a ranking of nodes in a graph based on on the structure of incoming links. Originally this algorithm was introduced to rank web pages. But it can be applied to all kind of graph related applications, like ranking street links in a given street network graph. The idea, in therms of a street network graph, is that each street link gets a numerical weighting assigned with the purpose of measuring the relative importance within the whole set. To get a high PageRank, a street link has to be linked by also high ranked links and/or by a high amount of other links. Because only graph nodes can be weighted, the street network graph first has to be converted to a line graph so that the street links become nodes and nodes (street junctions) become edges.

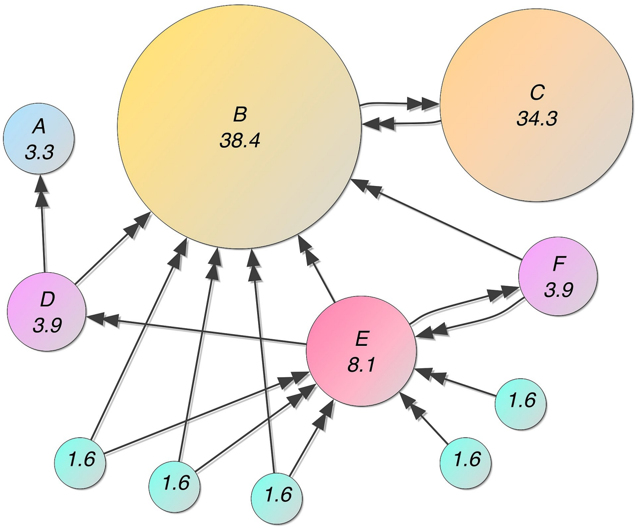


Figure 3.3: Example PageRank

[2] https://en.wikipedia.org/wiki/PageRank#/media/File:PageRanks-Example.jpg (Accessed on: 2019-11-29)

NetworkX

NetworkX (Hagberg, Schult, and Swart, 2008) is a python package for creation, manipulation and analyzing complex graphs. NetworkX provides not only all necessary analysis functions used in this thesis, but also comes with the functionality to export and visualize graphs in different formats like shape files.

OSMNX

OSMNX (Boeing, 2017) is a python library for downloading street networks based on the OpenStreetMap (OSM) graph. It allows one to download a whole street network for a given place, city, country name or by bounding box and coordinates. One can also specify the road network type to download as e.g. get all drivable roads or just all streets with private-access etc. The downloaded street network graph comes with all meta data for street links and junctions provided by OSM. Also different graph analysis tools, based on NetworkX [3] and Shapely [4] are available.

3.2.1 Inverse distance weighting (IDW)

This interpolation method, based on the work of Shepard, 1968, is usually applied on high variable data sets. The basic idea of this strategy is to estimate local measurements by taking the moving average of nearby given data points. Data points lying farther away from the interpolated value do have a much lower impact (weighting) than data points lying closer as one can see in equation 3.1:

$$u(x) = \sum_{k=1}^{N} \frac{w(x, x_k)}{\sum_{j=1}^{N} w(x, x_j)} u_k \tag{3.1}$$

[3] `https://networkx.github.io/documentation/stable/` (Accessed on: 2019-11-29)

[4] `https://shapely.readthedocs.io/en/latest/manual.html` (Accessed on: 2019-11-29)

Where $u(x)$ is the interpolation value of a point x based on known samples $u_k = u(x_k)$ for $k = 1, 2, ..., N$.
$w(x, x_k)$ represents the weighting function

$$w(x, x_k) = \frac{1}{d(x, x_k)^p} \tag{3.2}$$

Where d stands for a given distance metric, typically the Euclidean distance, between x and x_k.

3.2.2 Visualization tools

In this section the two main visualization tools are mentioned. In case of geospatial data, the QGIS Desktop application is used. For tabular data and data analysis visualization the python graphing library and plotly dash is used.

QGIS

QGIS is a Geographic Information System licensed under the GNU General Public License. It runs on Linux, Unix, Mac OSX and Windows and supports viewing, editing, and analysis of geospatial data.

Vector Layer

QGIS supports raster and vector layers. For this thesis, the main visual representation of the different geospatial data sources is to export these data as shape files [5]. Therefore vector data is stored as either point, line, or polygon features.

[5] `https://en.wikipedia.org/wiki/Shapefile` (Accessed on: 2019-11-30)

3.2.3 Plotly

Plotly.py

The plotly python package[6] is an open-source visualization library that supports all relevant chart types for statistical, geographical and scientific use cases. Because this python library is build on the top of the JavaScript library, it enables the creation of interactive web based visualizations out of Jupyter Notebooks[7] or IDEs like Pycharm[8].

3.2.4 Dash

Plotly dash [9] is a Python open-source library for creating reactive web applications. Dash is written on the top of Flask, Plotly.js and React.js and is ideal for building custom user interfaces in python. Applications are rendered in the web browser and therefore comes with a core set of typical HTML elements like inputs, buttons drop downs and so on but also with more complex components as interactive tables and graphs.

[6] `https://plot.ly/python/` (Accessed on: 2019-11-30)

[7] `https://jupyter.org/` (Accessed on: 2019-11-30)

[8] `https://www.jetbrains.com/de-de/pycharm/` (Accessed on: 2019-11-30)

[9] `https://dash.plot.ly/` (Accessed on: 2019-11-30)

4 Data Sources

In this chapter all required data sets and their sources are cited. Also the according data management strategies regarding data collection and storage will be stated.

4.1 Vehicle Crash Data

The vehicle crash data set from Austria was obtained from the Kuratorium für Verkehrssicherheit [1] (KFV). It is the same data set which was used in the preliminary work chapter although only crash records from the city of Graz are used. Each crash entry has been documented by an Austrian police officer. The crash entries are not unique by accidents but unique by accident participants. So if a crash with *n*-vehicles occurred than *n*-crash entries for this particular accident exist. Identified with the same Accident id, called *U_ID* but with their participant specific meta data.
As mentioned before only Graz related crash entries are used. So therefore samples with the feature *Gemeinde_ID* = 60101 are kept for further processing. The *Gemeinde_ID* is a unique identifier for municipalities in Austria. Further, only entries between 01.01.2015 and 31.12.2017 are used. This is because older crash entries hardly have any GPS related information so that they can not be linked with other geospatial data sets later on.
At all, there are 11.788 accident entries with 5.416 different *U_ID*s. Therefore 5.416 different recorded accidents occurred in this 3 years in Graz.
Each categorical feature of this data set is referenced to a look up table via an ID.

[1] `https://www.kfv.at/` (Accessed on: 2019-08-28)

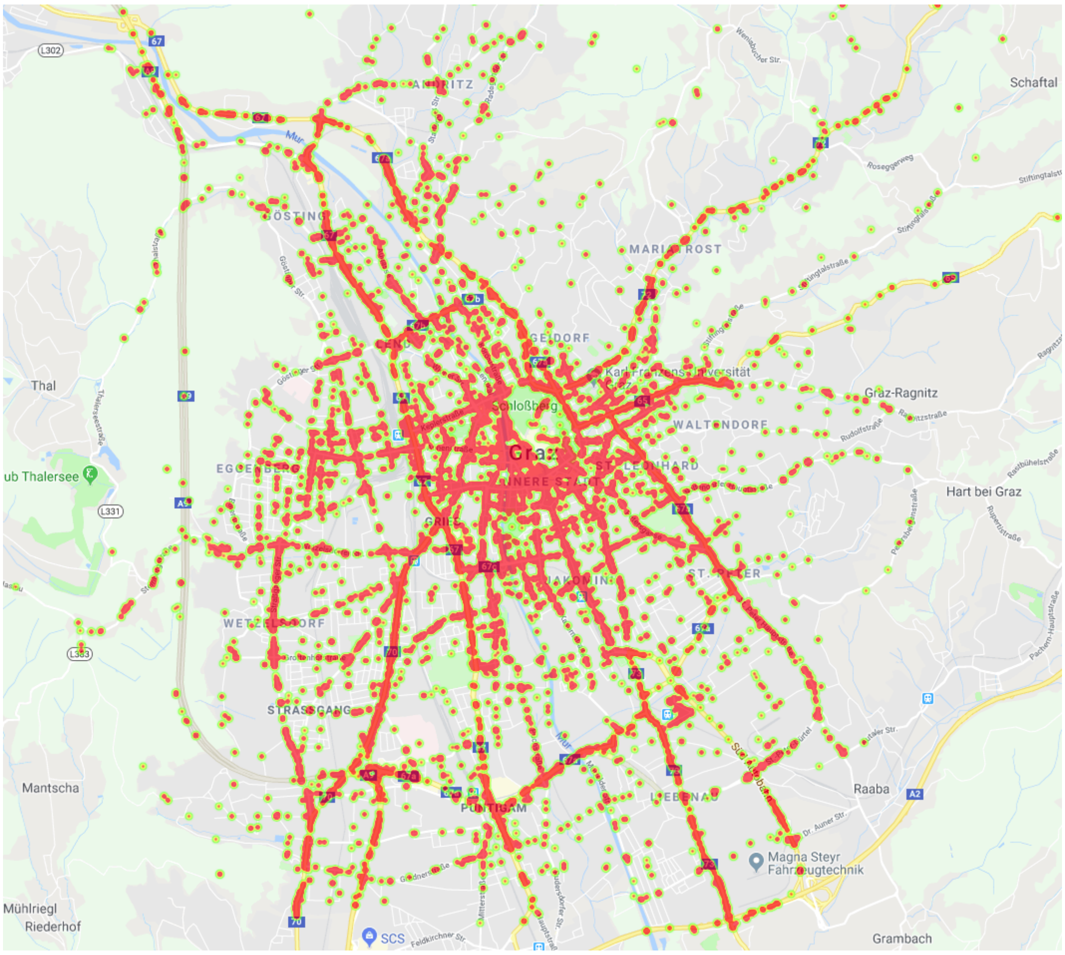


Figure 4.1: Accident heatmap Graz with crash entries between 2015 and 2017

4.2 Open Data Graz

4.2.1 Population

Open Government Data Austria [2] offers public, state specific data for free. For this thesis the data set ″population for Graz by district and age″[3] is used for additional information. To overcome the age attribute, the whole data set is aggregated by district and date time. Each entry represents the whole population of a specific district in the according month between 01.01.2015 and 31.12.2017.
This aggregated data set is stored as an MySQL Table for further processing steps.

4.2.2 Districts as Geo Data Export

For coming calculations and visualization purpose, the districts and their boarders are needed as Geo Data. This geo data set can be exported from the GIS Steiermark [4].
The export represents the 17 districts of Graz as separate polygons in one shape file in WGS84 coordinates. Against the population data set, the export remains as shape file because of the fact that MySQL 5.6 hardly supports spatial data types and the conversion effort would have been too high.

[2] https://www.data.gv.at (Accessed on: 2019-08-28)

[3] https://www.data.gv.at/katalog/dataset/stadt-graz_die-grazer-bev-lkerung-nach-bezirk-und-alter (Accessed on: 2019-08-28)

[4] www.gis.steiermark.at (Accessed on: 2019-08-28)

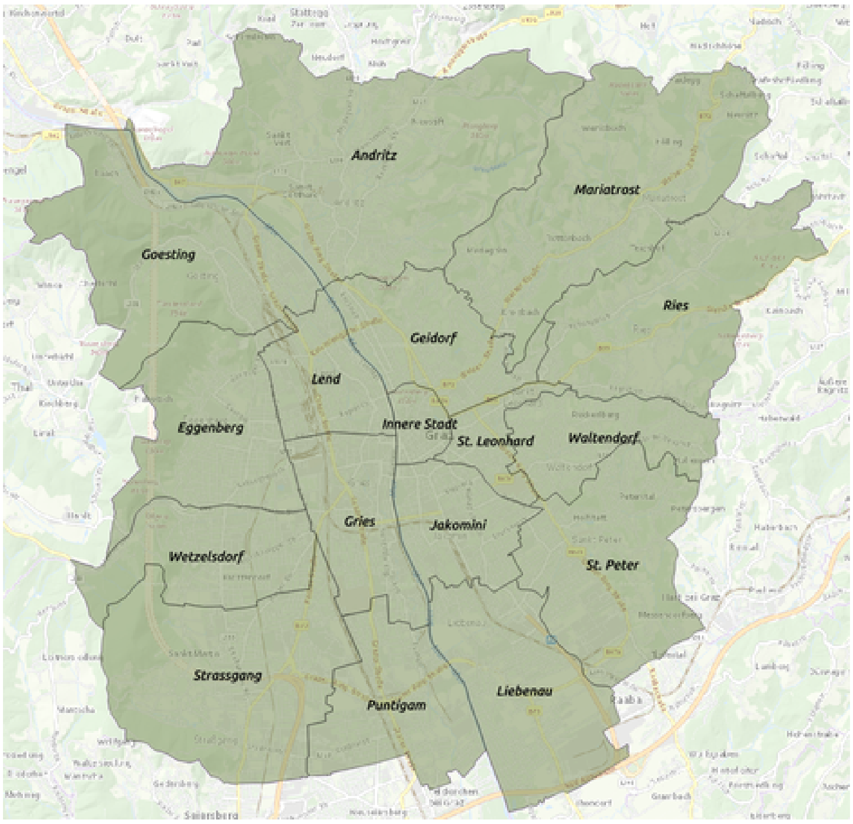


Figure 4.2: Shape file export from the districts of Graz

4.3 Road Networks

4.3.1 OpenStreetMap (OSM)

OpenStreetMap [5] is a collaborative project which offers a free editable map of the whole world. It provides spatial and routable geometries like street links with additional map features.
To download the street network for Graz, the python library OSMNX is used. OSMNX downloads the Graz related street network as a NetworkX graph given a place name or the bounding box (bbox) specific coordinates using the OverPass API[6] of OSM. Also the type of the street network, in this case all drivable roads, can be specified.
The downloaded graph, where each vertex represents a street junction or the end of a street and each edge stands for a street segment between two junctions, is stored as a graphml file. This simplification of graph nodes is done under the hood by the OSMNX framework [7]. Therefore all graph related operations can be applied.
As stated in Table 4.1, OSMNX also downloads the specific meta information for street links and junctions.

Feature	Description
osmid	unique street vertex identifier
lanes	representing number of lanes
length	representing the street segments length
maxspeed	maximum speed value
name	street name
oneway	True if there is an access restriction for this street segment

Table 4.1: Required OSM street meta information

[5] `https://wiki.openstreetmap.org` (Accessed on: 2019-08-28)

[6] `https://wiki.openstreetmap.org/wiki/Overpass_API` (Accessed on: 2019-08-29)

[7] `https://geoffboeing.com/2016/11/osmnx-python-street-networks/` (Accessed on: 2019-08-29)

Because of the internal street merging functionality it can happen that OSMNX combines streets with different street attributes. E.g. two or more street segments with different osmids or maxspeed values. Then these attributes also get combined and saved as different array values for these features. An example of this is represented in Table 4.2 and Figure 4.3. One can see that in this special case three different road links with osmids 205401689, 388770482, 47884494 get combined with different maxspeed values.

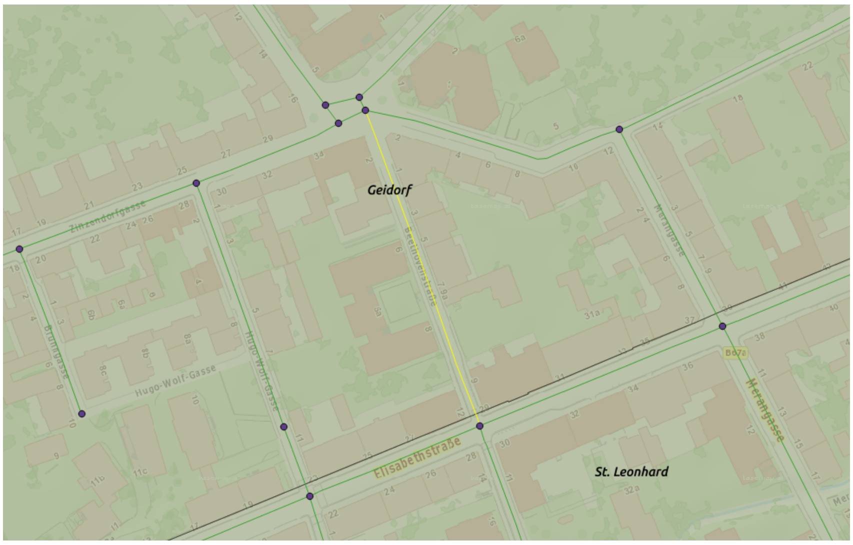


Figure 4.3: OSM street network with vertices and links

Feature	Value
osmid	[205401689, 388770482, 47884494]
lanes	NULL
length	160.492
maxspeed	[30, 20]
name	Beethovenstraße
oneway	True

Table 4.2: Merged street links for Beethovenstraße

4.3.2 Graphenintegrations-Plattform (GIP)

In contrast to OSM, GIP [8] is a joint Project of the Austrian federal states, ASFINAG, ÖBB Infrastruktur, the Austrian Federal Ministry of Transport, Innovation and Technology and ITS Vienna Region. Also the Austrian Association of Cities and Towns is a partner of this project.

GIP Structure

GIP itself consists of several databases, where each of them are maintained by their respective GIP partner. These databases get synchronized every two months and merged to an global, Austrian Street network graph with according meta data[9].
Regarding the road network graph, GIP defines the roads central axes as graph edges. Vertices are more complicated, because not only junctions are represented as a vertex but also changes in street attributes like the street name. Also when an low-level road network joins a higher-level network, a so called virtual vertex is generated.
OSMNX overcomes this issue internally when using OSM as road network, for GIP there is at all no framework which provides a simpler access to the GIP graph.
GIP comes with different graph sources, where the standard routing export is used in this thesis.
The export itself includes multiple CSV tables as one can see in Figure 4.4. For this thesis the relevant tables are:

- Link
- LinkUse
- LinkCoordinate
- Node
- TurnUse
- TurnEdge

[8] `http://gip.gv.at` (Accessed on: 2019-08-31)

[9] `http://open.gip.gv.at/ogd/0_dokumentation_gipat_ogd.pdf` (Accessed on: 2019-08-31)

Link

The Link Table includes all edges of the GIP Graph. The main attributes are stated in Table 4.3

Attribute	**Details**
Link_ID	unique link id
NAME1	the main name of the link
NAME2	all additional link namings seperated with slash
FROM_NODE	node id of the start vertex
TO_NODE	node id of the target vertex
SPEED_TOW_CAR	average speed in km/h in digitization direction
SPEED_BKW_CAR	average speed in km/h against digitization direction
MAXSPEED_TOW_CAR	max speed in km/h in digitization direction
MAXSPEED_BKW_CAR	max speed in km/h against digitization direction
ACCESS_TOW	bitmask of practicability in digitization direction
ACCESS_BKW	bitmask of practicability against digitization direction
LENGTH	length of the link in meters
LANES_TOW	number of lanes in digitization direction
LANES_BKW	number of lanes against digitization direction
WIDTH	width of the link in meters
ONEWAY	allowed driving direction: 1 stands for in digitization direction, 0 against digitization direction, 2 in both directions, -1 unknown

Table 4.3: Link table attributes

LinkUse

This table includes all the different link usages per street link. All relevant attributes for this thesis are mentioned in Table 4.4.

Attribute	**Details**
USE_ID	Use ID
LINK_ID	GIP link id where the linkuse lies
Offset	horizontal distance between linkuse central axis and link central axis in meter. (negative values represent distance to the left regarding digitization direction)
Width	average width of linkuse in meter
MINWIDTH	minimal width of linkuse in meter
SURFACE	road surface ID according to the LUT_SURFACE table

Table 4.4: Linkuse attributes

LinkCoordinate

The LinkCoordinate table contains all points between the start and the end point of a link. The coordinate points of the FROM and TO nodes are also included.

Attribute	**Details**
LINK_ID	according link id
COUNT	ongoing numeration of all between points of the link starting at 1
X	x-coordinate of the between point
Y	y-coordinate of the between point
Z	z-coordinate of the between point

Table 4.5: LinkCoordinate attributes

Node

The Node table includes all vertices of the GIP graph.
The main attributes are the Node_ID and the coordinates X, Y and Z in WGS84 CRS format.

TurnEdge

The TurnEdge table includes the turn permissions on link basis. In Table 4.6 all relevant attributes are stated.

Attribute	**Details**
TURN_ID	ID of the turn relation
FROM_LINK	link ID of the FROM edge
TO_LINK	link ID of the TO edge
VEHICLE_TYPE	vehicle type for which the turn is permitted

Table 4.6: TurnEdge attributes

TurnUse

The TurnUse table includes the turn permissions on linkuse basis. In Table 4.7 all relevant attributes are mentioned.

Attribute	**Details**
TURN_ID	ID of the turn relation
FROM_USE	linkuse ID of the FROM edge
TO_USE	linkuse ID of the TO edge
VEHICLE_TYPE	vehicle type for which the turn is permitted

Table 4.7: TurnUse attributes

LUT_Surface

This lookup table matches the surface id, which is stated in the LinkUse tabel, to the specific road surface type.

ID	Name
-1	unknown
3	unattached (like gravel)
4	paved road (like asphalt)
6	pavement
8	off road

Table 4.8: Lookup table surface attributes

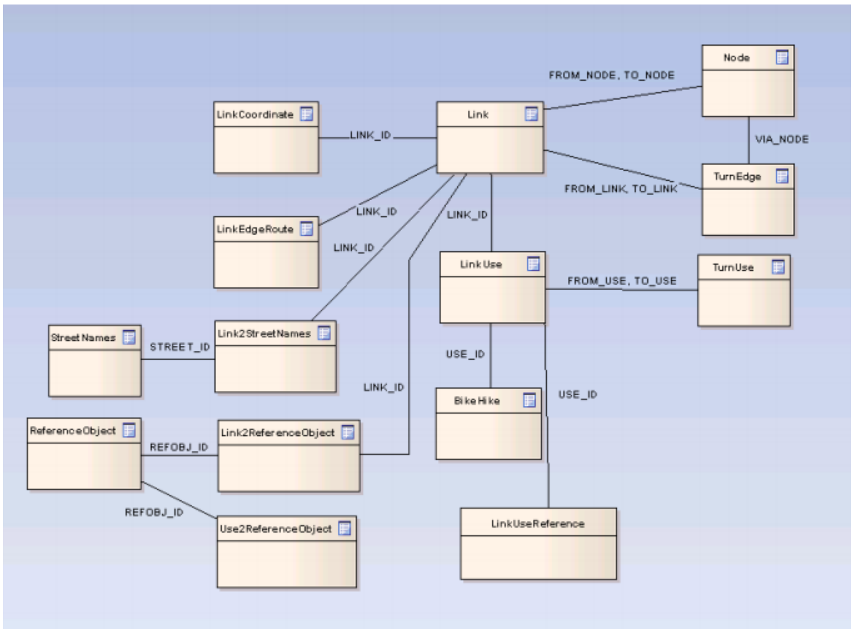


Figure 4.4: Overall database schema of the GIP routing export

4.4 Traffic Flow

Traffic flow specific data is handled by two main institutions in Styria: GIS Steiermark[10], and the department of roads of the city of Graz [11].
GIS Steiermark maintains all Styria specific county roads, federal highways and highways. They also handle some specific city street streets but not Graz.
Instead of GIS Steiermark, the city of Graz maintains all of their city roads and so also the according traffic flow measurement stations by their own. Both data sets are not open source.

4.4.1 GIS Steiermark

GIS Steiermark maintains 30 traffic flow stations for different county roads and federal highways in Graz. Traffic flow entries are available for this stations between 01.01.2017 and 31.12.2017 where each entry consists of the following attributes as mentioned in Table 4.9.

Attribute	**Details**
measurementId	unique identifier of the traffic flow measurement station
vehicle1	number of cars in digitization direction of GIS Steiermark
vehicle2	number of cars against digitization direction of GIS Steiermark
truck1	number of trucks in digitization direction of GIS Steiermark
truck2	number of trucks agains digitization direction of GIS Steiermark
dateTime	timestamp when the vehicle was tracked

Table 4.9: Traffic flow attributes for GIS Steiermark measurement stations

[10] `http://www.gis.steiermark.at` (Accessed on: 2019-09-07)

[11] `https://www.graz.at/cms/beitrag/10023623/7755415/Verkehrssteuerung_und_Strassenbeleuchtung.html` (Accessed on: 2019-09-07)

Each measurement station tracks a specific road link in both driving direction and distinguishes between cars and trucks. Compared to the GIP road network one can say that the GIS digitization direction not directly matches the GIP digitization direction.

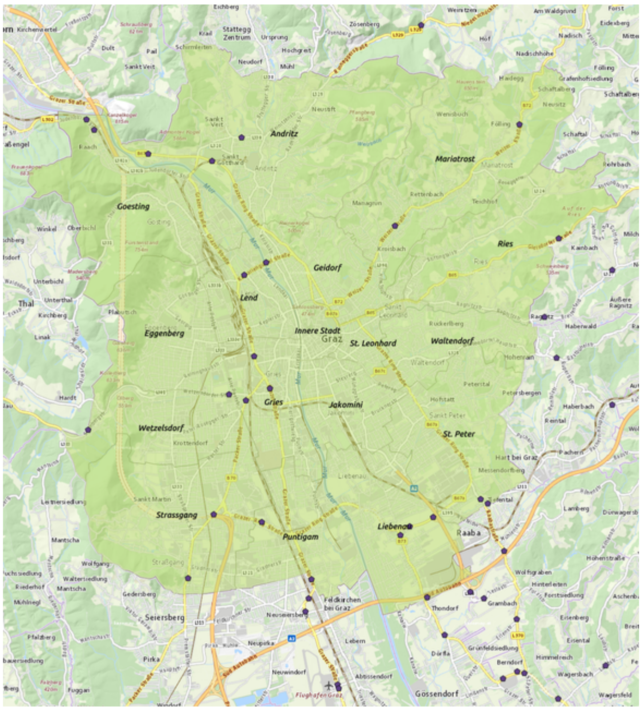


Figure 4.5: Overview over all GIS traffic flow measurement stations in Graz

4.4.2 Department of Roads Graz

The Department of Roads in Graz maintains about 200 measurement stations for traffic flow. In contrast to the GIS stations, traffic flow entries are now available between 01.01.2015 and 31.12.2017. One issue which has to be overcome is the fact that not all of these measurement units provide traffic flow values the whole time range. This has several reasons which will be discussed in a later chapter but the two obvious reasons are that one unit just started working some time later on or stopped working before the 31.12.2017.

Attribute	Details
measurementId	unique identifier of the traffic flow measurement station
datetime	timestamp of the measured value
count	number of tracked vehicles

Table 4.10: Traffic flow attributes for measurement units maintained by the city of Graz

Another difference compared to the GIS related traffic flow units is that now 4 wheel vehicles, independent of the vehicle type, are tracked and aggregated in an interval of a quarter-hour.
Also one measurement unit does not track the whole cross-section of the road link like the GIS units do, but only one lane of this specific road link. By adding up all lane specific measurement values at a certain timestamp of a given road link outputs the real traffic flow value in the last quarter hour.

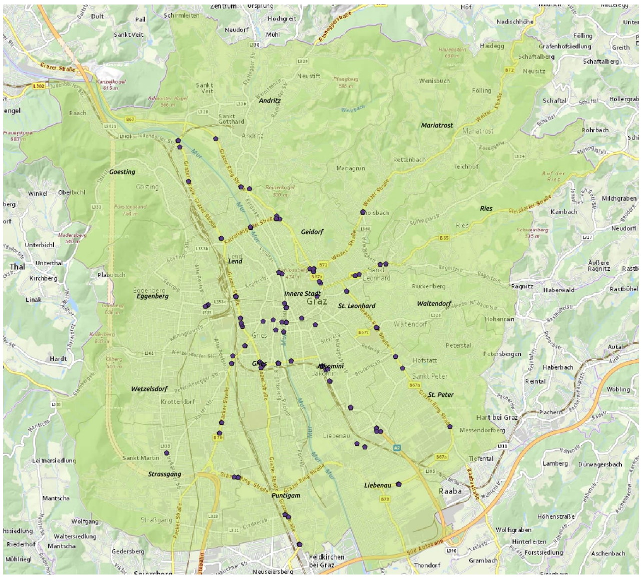


Figure 4.6: Overview over all traffic flow measurement stations in Graz maintained by the City of Graz

4.5 Weather Data

To make use of weather related explanatory variables for the accident prediction model, rainfall and temperature data of the main weather stations in and around Graz are considered. These stations are maintained by the official meteorological and geophysical service of Austria, named ZAMG [12].

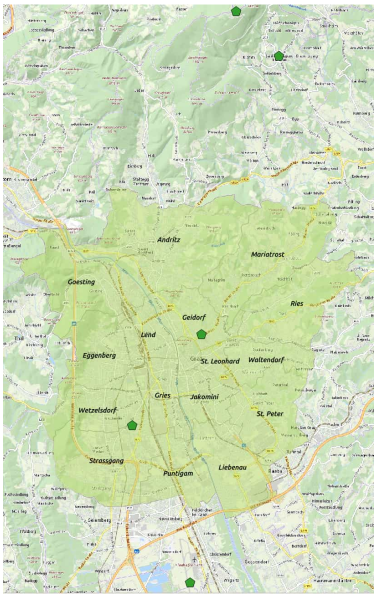


Figure 4.7: Overview over all ZAMG weather stations in and around Graz

weather station name	elevation in meters
Graz/Strassgang	357
Graz/Thalerhof	340
Schöckl	1445
St. Radegund	724
Graz/Universität	366

Table 4.11: Overview over weather station attributes

[12] https://www.zamg.ac.at/cms/de/aktuell (Accessed on: 2019-10-04)

In Figure 4.7 and Table 4.11 the considered weather stations are shown. One measurement sample of a weather station includes the temperature and the precipitation. The resolution of the measurements is one hour.

5 Data Preprocessing and Feature Engineering

In this chapter all necessary preprocessing steps for each data set are explained. Also some additional features are extracted and generated based on the different data sources. Another important step in this section is to define the right feature scaling methods for a standardized model input data set.

For the GIP related data set, all additional features are stored in a separate GIP metadata table. In this case, a completely new GIP export can be imported without corrupting the pipeline itself. For visualization purpose, a shape file is generated at the end of the pipeline with all the necessary attributes as metadata.

Against that, the additional generated OSM features are directly stored in a shape file and not in a separate table. This is because the OSM export already comes in a graphml file and not as a database export.

5.1 Road Network Pipeline

In this section the preprocessing pipeline for the road network graphs are explained. Both, the GIP and the OSM graph are considered so that in the following sections the evaluation on both can be done.

The GIP specific network data is read in from the different database tables:

- Link
- LinkCoordinate
- Node

Out of this sources the whole road graph can be generated via the NetworkX library. It must be stated that only Graz related roads are considered. This can be achieved as that each road link must lie in one of the districts of Graz or intersect one of them. Further, only roads with the attribute *access_tow* or *access_bkw* greater than 3 are taken into account. This is because the practicability of a road link is described via a bitmask where the two lowest bits stand for pedestrian and bike practicability only. In the accident data set no entry exists where only bikes or pedestrians are mentioned. So there must be at least one motorized vehicle involved in an accident.

ID Bit	Name Bit	Value (decimal)
0	pedestrian	1
1	bike	2
2	private car	4
3	public bus	8
4	railway	16
5	tram	32
6	subway	64
7	ferry boar	128

Table 5.1: Practicability-bits

The OSM graph is much easier to handle because it can be directly downloaded as a graphm file via the OSMNX function $graph\_from\_place$. This function returns a NetworkX graph from OSM data within the spatial boundaries of the given geocodable place. One can also specify the street network type via the parameter *network_type*. For this use case the type *drive* is chosen which means that the graph is generated out of all drivable public streets of the given place.

5.1.1 PageRank Road Network Graph

In this step all the road links from the city of Graz are page ranked. First, the graph is converted to a directed graph, and afterwards converted to a line graph. This means that a line graph of a graph G has a node for each edge in G and an edge joining those nodes if the two edges in G share a

common node.
Afterwards the line graph is used as input for the NetworkX PageRank method. At the end the ranked line graph nodes must be matched to the road network links of the original graph. The highest ranked roads are stated in Table 5.2

osmid	**road name**	**road category**	**pagerank**
122564305	Grazbachgasse	primary	0.89
187770191	Weblinger Gürtel	primary	0.83
48377963	Friedrichgasse	residential	0.76
147282924	Wiener Straße	primary	0.76
[188716194, 188716188, 187417426]	Pyhrn Autobahn	motorway	0.75
[4100026, 332304639]	Grüne Gasse	residential	0.72
[24302451, 1976536615]	Joanneumring	primary	0.70
4354717	Schmiedgasse	residential	0.70

Table 5.2: Highest page ranked roads in Graz based on the OSM graph

As one can see in Table 5.2, the calculated PageRank values are based on the OSM graph and scaled between zero and one. The PageRank measurements differ from the values calculated on the GIP graph. This is because in GIP not every graph node represents a junction and so the graph structure itself is not the same as for the OSM. So the PageRank values based on the OSM graph are much more representative than the values based on the GIP network, although GIP contains some more road links, especially residential roads.
For the calculation regarding the OSM graph the page ranked values are not based on the road link directly but on the direction of the specific road link lanes. This means if the road is passable in both directions than there exist two different PageRank values, each for one driving direction.

5.1.2 Centrality Calculation of the Road Network Graph

For the calculation of the road centrality the NetworkX function *closeness_centrality* [1] is used. This centrality measurement represents the reciprocal of the average shortest path distance from one node to all the other nodes in the graph.

$$C(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(v,u)} \tag{5.1}$$

where $d(v,u)$ is the shortest-path distance between node u and node v, and where n stands for the number of nodes that can reach node u.
Therefore, higher values of closeness indicate higher node centrality.
Again the road network graph is converted to a directed line graph. This new graph is used as input for the closeness centrality calculation. The output is visualized in Figure 5.1 and the most central road links are stated in Table 5.3.

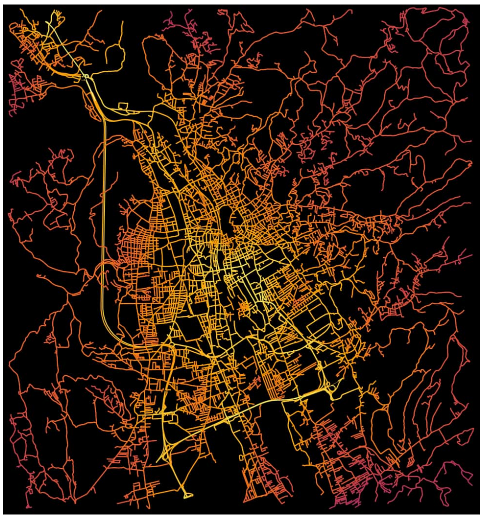


Figure 5.1: Closeness centrality for the road network of Graz

[1] https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.centrality.closeness_centrality.html (Accessed on: 2019-09-10)

osmid	road name	road category	value
[188652278, 25716155, 121165757, 127761350, 188652269]	Pyrn Autobahn	motorway	1.0
[526110760, 5132424, 5132425, 5113079, 191915152, 191915153, 26603987, 35401652, 35401653, 26603990, 206372489, 5113080, 191915156]	Süd Autobahn	motorway	0.93
[136332200, 136332201]	Liebenauer Tangente	primary	0.93
[95541665, 27445571, 27445573, 5116323, 108616723, 132659645]	Conrad-von-Hötzendorf-Straße	primary	0.93
[163080140, 173538868, 108616733, 163080142, 5112957]	Conrad-von-Hötzendorf-Straße	primary	0.93
134122989	Grazerstraße	primary	0.92
[32425488, 132659672, 163080160, 377169733, 197686471]	Schönaugürtel	primary	0.92
[108616640, 5098330]	Jakominigürtel	residential	0.92

Table 5.3: Roads with the highest centrality measurement values in Graz based on the OSM graph

Like for the PageRank calculation, the estimated centrality measurements are based on the OSM graph and scaled between zero and one. Once more, the calculation differs between OSM and GIP graph because of the different graph structure. The centrality measurement is now link specific and does not depend on the roads traffic directions.

5.1.3 Road Slope Estimation

The estimation of a road link slope values can only be done based on the GIP graph because GIP comes with height values in the *LinkCoordinate* table. Therefore the slope between the start and end node of a link can be calculated like mentioned in Equation 5.2.

$$\begin{aligned} \Delta_{height} &= |v_z - u_z| \\ slope &= tan(arcsin(\frac{\Delta_{height}}{length(link)})) * 100 \end{aligned} \tag{5.2}$$

where u represents the start node and v the target node.
For the right estimation of the *road_link* the distance has to be transformed from the Euclidean Distance to the orthodromic distance of the specific road link. Therefore the python library *pyproj* is used.

5.1.4 Road Curvature Estimation

A road links curvature is calculated like defined in Equation 5.3

$$curve_{link} = \frac{length(link)}{d(link_{start}, link_{end})} \tag{5.3}$$

where $d(link_{start}, link_{end})$ is the orthodromic length of a straight line between the first and the last link coordinate. Therefore, higher curvature values indicates higher degree of road curves.

5.1.5 Junction Plateau definition

In Austria the StVO law[2] states that a road junction is defined as a place where one road intersects or joins another, no matter in which angel. Further, the junction area is the overlapping space of these roads bounded by the imaginary lines of the contiguous roadsides.
Following this law, polygonal representations of street junctions can be generated. For rectangular X-junctions this approach is not so complex to handle as shown in Figure 5.2.

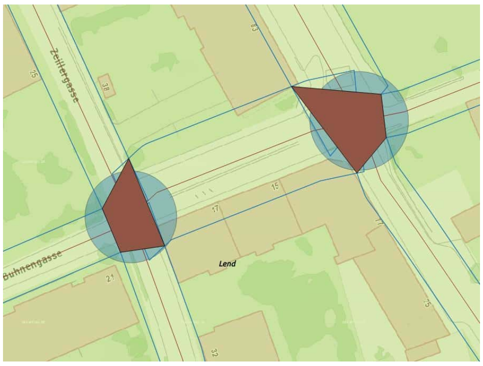
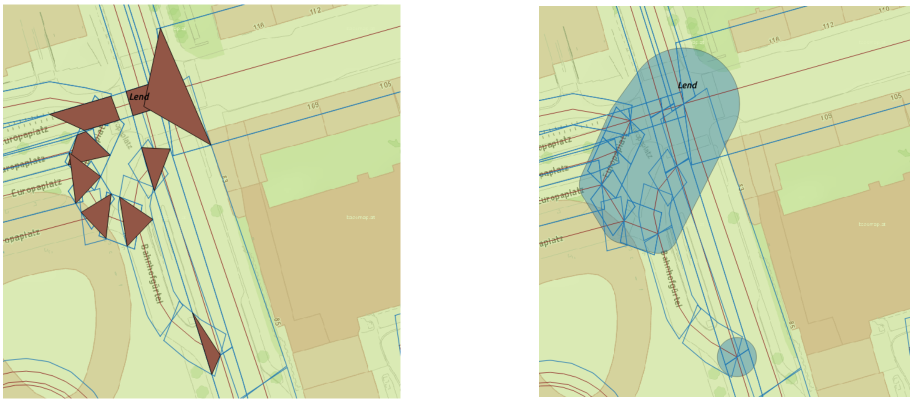


Figure 5.2: Road junction with polygonal and radial area representation

Most of the junctions are not in this shape. Another issue is that some junctions are very close to each other as one can see in Figure 5.3. In this case it would be preferable to handle these different junctions as one greater junction. So for an abstract representation of junctions the following steps are carried out:
First, only consider graph nodes where more than two links are joining. This is necessary especially for the GIP graph, because otherwise this graph node is a virtual node and does not represent a real junction.
Next gather all road links width joining this specific node and take the link with the maximum width.
Define a radial area around the node with a radius of the maximum link width.

[2] https://www.ris.bka.gv.at/Dokument.wxe?Abfrage=Justiz&Dokumentnummer=JJT_20080124_OGH0002_0020OB00062_07T0000_000 (Accessed on: 2019-09-11)

If radial areas intersect each other, merge them recursively until no more intersections occur.



(a) (b)

Figure 5.3: (a) represents the polygonal road junction areas whereas (b) visualizes the abstracted radial representation with the merging strategy

The generated plateau IDs are stored in a separate junction SQL table with reference to the specific graph node IDs.

5.2 Map Matching

In order to generate the input data set for the prediction model, all additional data sets have to be matched to the GIP and OSM road network. Therefore different map matching strategies, depending on the data set, are used.

5.2.1 Match Vehicle Crash Data with Road Network

For further analysis all road accidents which occurred in Graz are mapped to the road network graph. Accidents can either be matched with an according road link, a junction plateau or both.

Match on road network links

Each crash event is matched with the nearest road segment from the OSM and GIP network where the traffic mode of the accident entry matches the street practicability. This means, when there occurred an accident with a tram the accident has to be matched to the nearest link where the practicability of this link is also possible for trams.
One additional consideration was not only to check the nearest possible link with the right practicability but also if the street name coincides. This approach fails in some cases because not all accident entries contain the street name attribute and sometimes the street name of the crash entry just not matches with the n-nearest road links. So the assumption in this case is that the GPS data is more reliable than the manually filled attribute for the street name in the accident data set.
In Figure 5.4 the street segments with the highest accident rates are visualized.
One can also see that the most crashes happen in the districts of Gries and Lend, whereas the fewest occur in Mariatrost, Waltendorf and Ries. It also has to be said that ″Gries″ and ″Lend″ have a dense street network with a considerable number of arterial roads like the ″Bahnhofgürtel″ or ″Eggenberger Gürtel″. Also typically accidents take place on streets with one lane in a direction as the attribute *lanes_tow* and *lanes_bkw* shows.

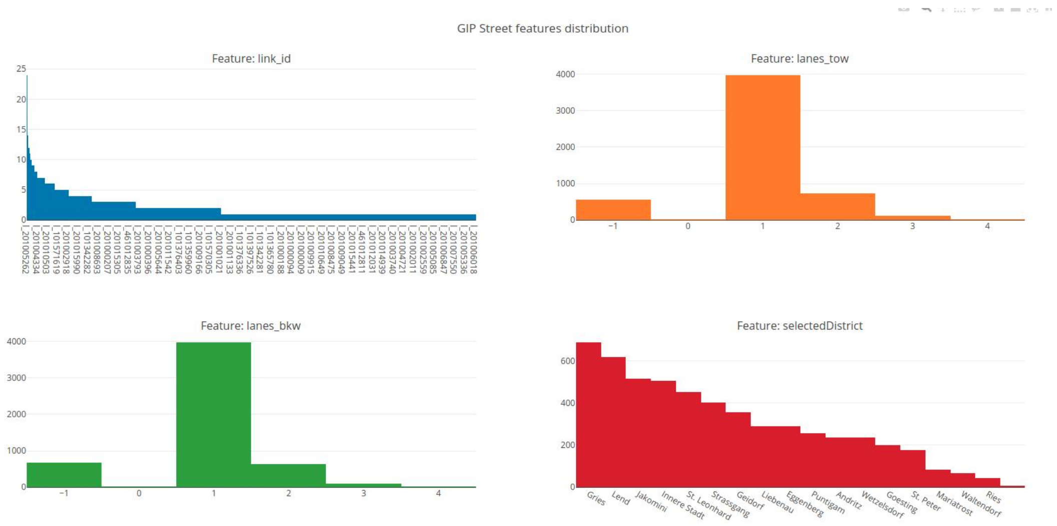


Figure 5.4: Distribution of road features where crashes occurred

GIP ID	road name	number accidents
201005262	Joanneumring	24
201007193	Annenstraße	18
201007177	Keplerstraße	17
201000754	Kärntnerstraße	15
101571709	Lazarettgürtel	14
101374559	Conrad-von-Hötzendorf-Straße	14

Table 5.4: GIP links with highest accident rates

In Table 5.4 the link IDs are matched with the according street names and in Figure 5.5 the street segment of Joanneumring is visualized being the link with the highest accident rate in Graz.

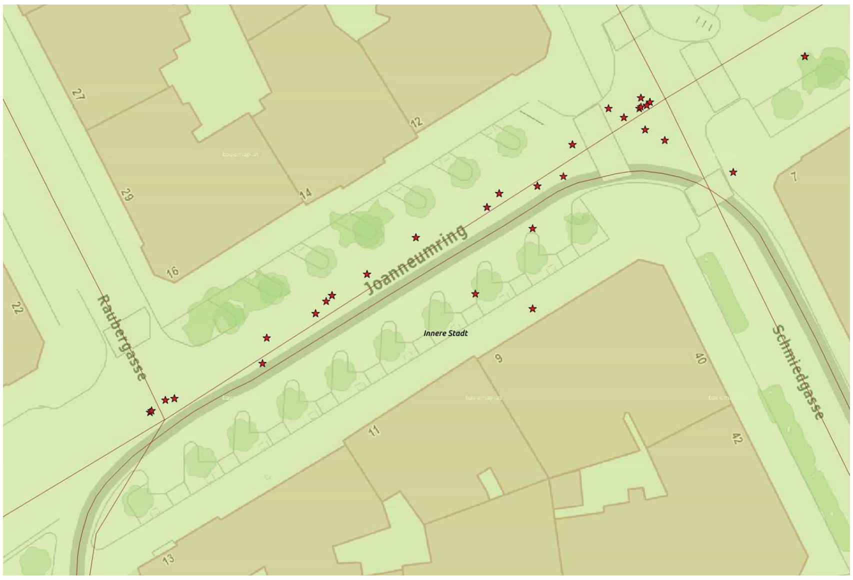


Figure 5.5: Street link Joanneumring with according crash events

Match on Road Network Intersection Plateaus

In this section accidents are matched to intersection plateaus if minimum one of the following cases is true:

- If the coordinates of an crash entry intersects a intersection plateau
- If an crash entry in the accident data set is mentioned to happen at an intersection.

The first point is straightforward. Each accidents entry coordinate pair in the KFV crash data set is checked if there is an intersection with the intersection plateau generated in Chapter 5.1.5. Then this crash belongs to the specific intersection plateau.
The second step is necessary if the specific accident is identified as an intersection accident in the crash data set but not directly intersects an intersection plateau. Then the plateau is chosen which has the smallest euclidean distance.
The accident entry in the crash data set is determined to be at an intersection if the attribute *Kennzeichnung1_IDs_Kve* has one of the IDs stated in table 5.5

Kennzeichnung1_IDs_Kve	description
14	4-way-intersection
15	3-way-intersection
16	5-way-intersection
17	intersection with shifted joining roads
102	intersection

Table 5.5: Lookup table Kennzeichnung1_IDs_Kve

In Figure 5.6 the junction plateaus with the highest accident rates are visualized. Against the road link, nodes do not have names but only IDs. The plateau with the highest accident rate is shown in Figure 5.7.

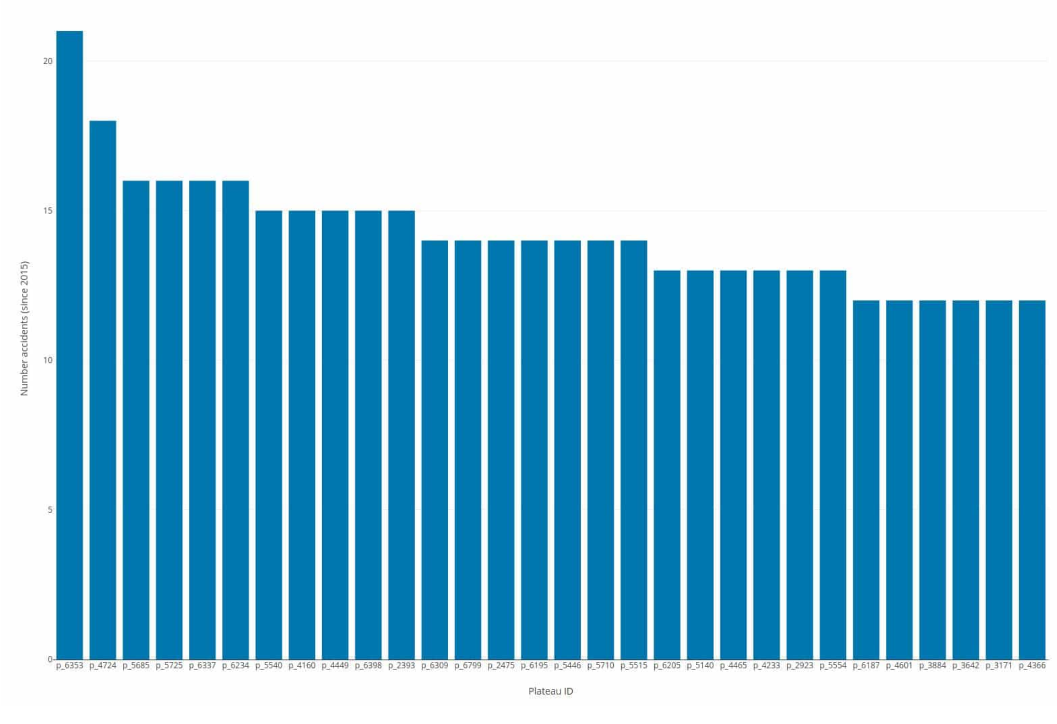


Figure 5.6: Distribution of crashes over intersection plateaus

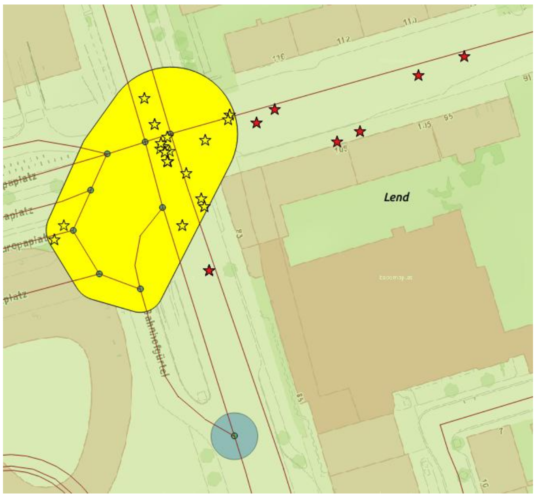


Figure 5.7: Intersection plateau with the highest crash rate

5.2.2 Match districts with road network

To get an impact of the population in the different districts the population density, as stated in Equation 5.4, is calculated. Therefore the according district has to be mapped to all road links. This is done by choosing the district which is intersected by a road link. If the road link passes multiple districts, choose the one containing the greatest part of the link.
Because the population not only depends on the district but also on date and time a static mapping of the population density can not directly be made. A more detailed examination follows in Chapter 7.3.

$$density(district) = \frac{population(district)}{area(district)} \tag{5.4}$$

5.2.3 Match weather data with road links

Since the weather related data from the different measurement stations can not directly be matched to each road segment, an interpolation technique is used to calculate these measurements at each road link separately. This can be realized via inverse distance weighting (Noori, Hassan, and Mustafa, 2014 and Mair and Fares, 2010) given the GPS coordinates of the 5 different weather stations. Not only the temperature but also the rainfall data is interpolated in that way. In python this can be realized via the KDTree [3] approach from the scipy library. The temperature and precipitation for each road link is estimated on the median coordinate entry of each road link.

[3] `https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.spatial.KDTree.html` (Accessed on: 2019-11-27)

6 Missing Value Imputation for Traffic Flow Data

In this section the traffic flow data set provided from the department of roads in Graz is analyzed and preprocessed so that it can be used as an additional input for the accident risk prediction model.
First some traffic flow stations (TFS) and their time series measurements are analyzed. One will see that, depending on the TFS, a great percentage of missing values occur. This problem can be overcome by using different imputation strategies. Some of these strategies will be compared and explained in detail depending on this special use case.
At the end one imputation strategy is chosen and evaluated on different subsets of the traffic flow data set so that the imputed values have a high level representation of the missing values.

6.1 Traffic flow data analysis

For analysis purpose, a small plotly dash[1] application should support to get a deeper insight about the measured values of a given TFS. Given a specific TFS and a date range, a histogram visualization in a quarter hour interval with standard deviation is generated.
Also a Kernel Density Estimation (KDE) graph, as shown in Figure 6.4 and 6.5, is visualized. One can aggregate the measured values of a station by day, hour and quarter hour intervals. This plot is for illustrative purposes only so no further fine tuning regarding the width of the histogram bins, which is defined as a constant value of 10, is carried out.

[1] `https://plot.ly/dash/` (Accessed on: 2019-09-12)

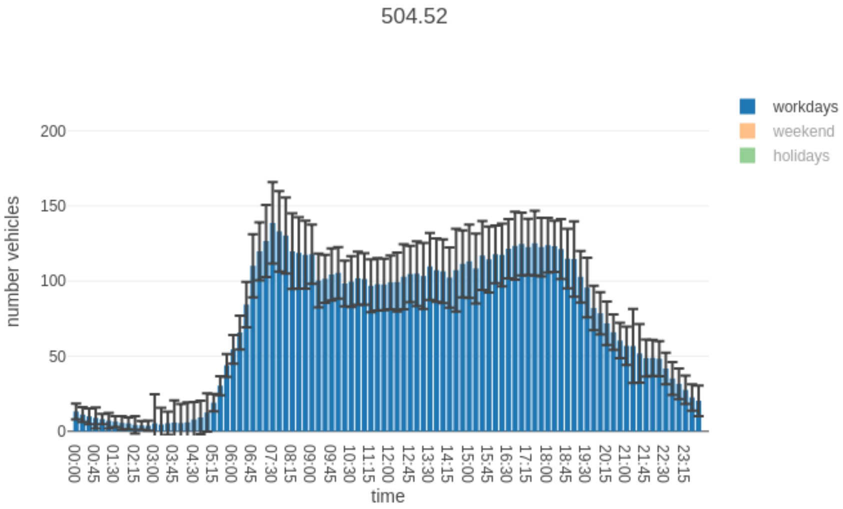


Figure 6.1: Histogram of traffic flow measurements on workdays between 01.01.2017 and 01.01.2018 in an quarter hour interval

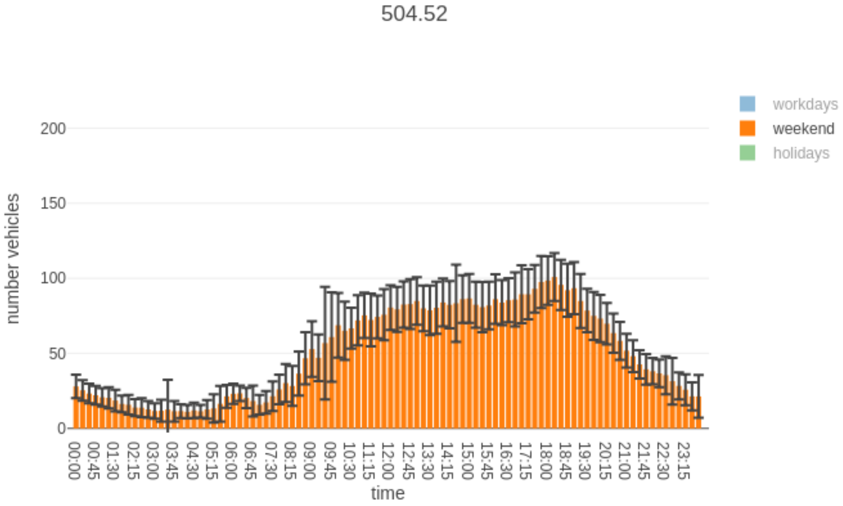


Figure 6.2: Histogram of traffic flow measurements on weekends between 01.01.2017 and 01.01.2018 in an quarter hour interval

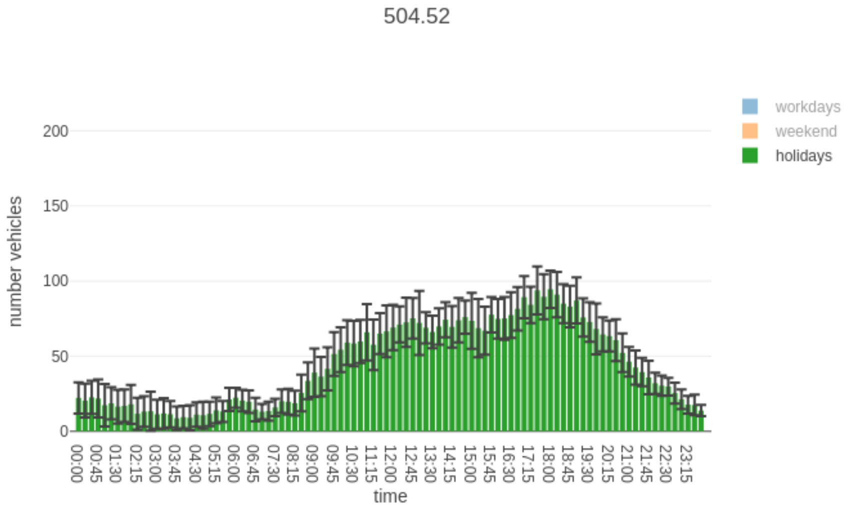


Figure 6.3: Histogram of traffic flow measurements on public holidays between 01.01.2017 and 01.01.2018 in an quarter hour interval

Looking at the TFS 504.52, which is mounted on an arterial road where a high traffic volume can be detected, the following conclusions can be drawn:

- On workdays, as visualized in Figure 6.1, a steep rise of numbers of vehicle can be viewed between 05:00 and 07:30 with a relative small standard deviation compared to the absolute values. At around 07:30 the highest traffic volume of the day is reached. The average value in this quarter hour is around 140 cars with a standard deviation of ± 25 cars.
  Afterwards the traffic volume decreases slowly over the day till around 16:30 when it slowly starts increasing again and reaches a local maximum between 18:00 and 18:45. One can not clearly detect a maximum in this afternoon rush our because the mean values are very close to each others in this time range and so the standard deviation plays an important role in this case.

- On weekends, visualized in Figure 6.2, the absolute values in the night between 00:00 and 04:00 are slightly larger than on weekdays. This is because of the vicinity to a greater nightlife scene with bars and clubs in Graz. Also there is no absolute rush hour in the morning but a slightly increase of the traffic volume over the day. The maximum is reached at around 18:30. One can also view an increase of the standard deviation compared to weekdays over the whole day. The increase of the standard deviation might be because of special events in the closer vicinity at certain time which may cause slightly different traffic flow values. So one solution in further consideration would be to get a very detailed event plan for the whole city so that an additional aggregation mode with "event day" / "no event day" can be introduced.
- The measurements on holidays as stated in Figure 6.3 are closely comparable to the traffic flow values on weekends although the standard deviation is not as large as for the weekend distribution.

Another reason why this station is good for analysis purpose is that compared to many other station it has no missing values in this time range. On can see that for specific intervals the standard deviation is quite small compared to the mean value. But there are still some regions where the opposite is true. This may be overcome by aggregate not only by day and time but also by months or by seasonal factors.
Another view on the problem is to generate a probability density function (pdf), which can be adjusted by the different aggregation modes. The goal is to get something like a unimodal distribution with a small variance so that for a given TFS under certain conditions like year, day, hour and so one one can sample missing values from the specific distribution.

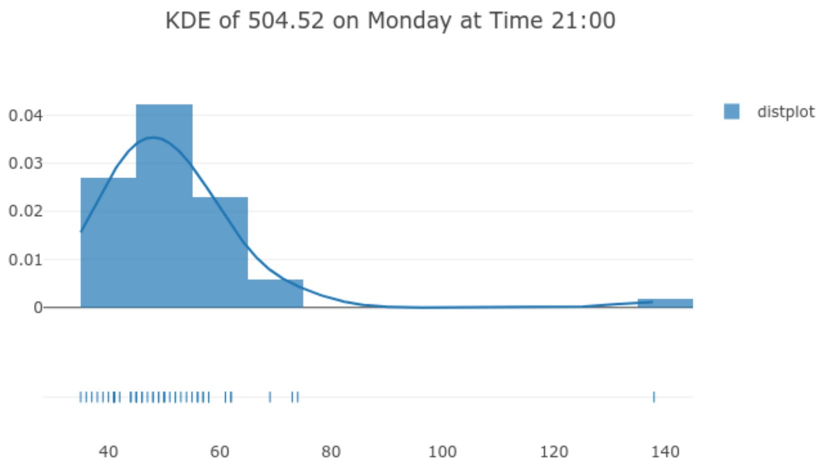


Figure 6.4: Unimodal probability density function with a small variance

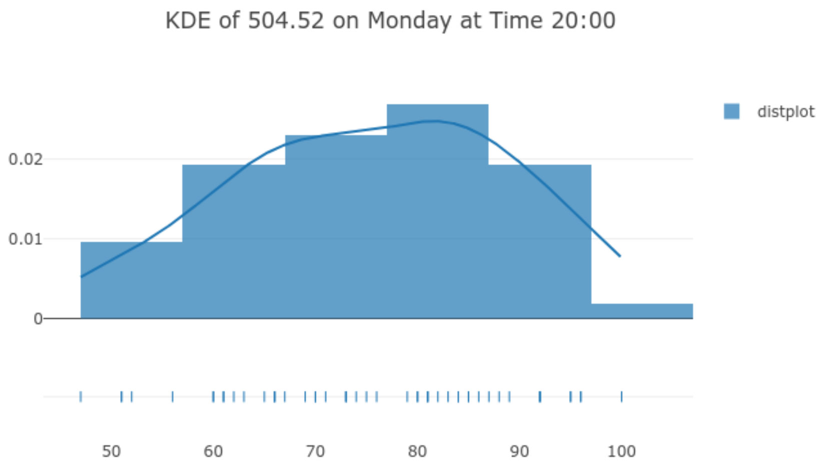


Figure 6.5: Probability density function with large variance

In Figure 6.4 a unimodal distribution with small variance is shown. So for

the TFS 504.52 on Mondays at 20:00 there are no great changes in the traffic volume. The problem of a pdf with a large variance as displayed in Figure 6.5 arises the most time at the rush hours.
So therefore the imputation can not be seen as a univariate analysis problem. Further experiments will show that other TFS might have an great impact on the imputation process of a missing value for a given TFS.

6.2 Missing Value Statistics

For a adequate imputation model selection the missing values over all TFS have to be analyzed. Not only the percentage of missing values regarding the TFS are considered but also if series of consecutive missing values occur. In Figure 6.6 the percentage of missing and valid values for all TFS are mentioned. One can see that for more than 170 TFS the percentage of missing values is lower than 15. But there are also about 15 stations where approximately half of the measurements are missing. A special case regarding missing values can be viewed for station 515.21: The station, located at the *Riesstraße* in the district of *Geidorf* recorded measurements between 01.01.2017 and 01.01.2018. Between this period only 8 valid values for a quarter hour interval could be generated.

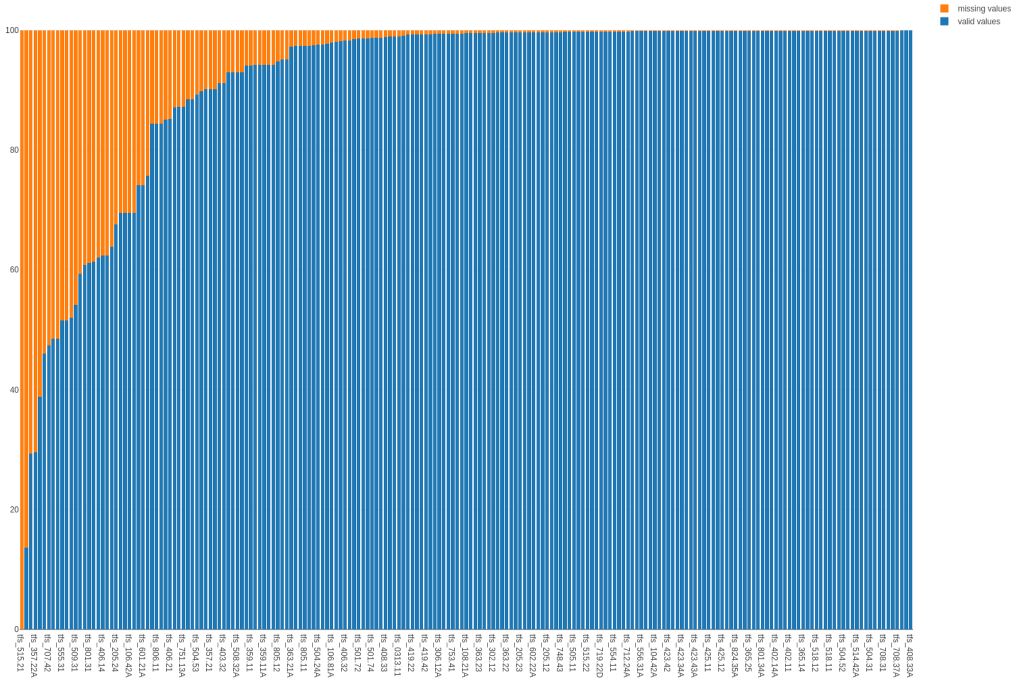


Figure 6.6: Missing value statistic in percentage

Figure 6.7 plots the cumulative sum of the number of different consecutive missing value series. One can see that about 40 percent of all missing value series is just a series with one not valid measurement. 61 percent of missing values are series lower than 4, which implies that these series are not longer than one hour. Also some peaks can be figured out at 26 (6.5 hours), 84 (21 hours) and 96 (1 day) consecutive missing values.
As a summarization it can be said that 90 percent of all missing measurements are values in a series of missing values which lasts less than one day, or in other words less than 96 consecutive missing periods.

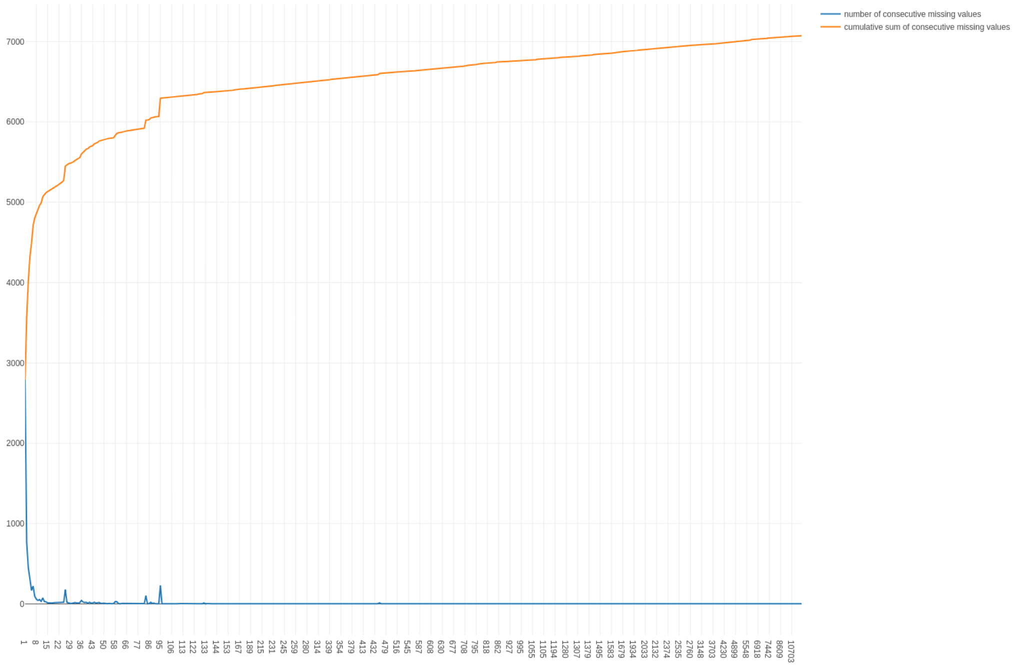


Figure 6.7: Cumulative sum over number of consecutive missing value series

6.3 Imputation Model selection

6.3.1 Missing data mechanism

Before the right imputation model can be selected the following missing data mechanism have to be discussed as stated in Soley-Bori, 2013:

- **Missing Completely at Random (MCAR)**
  Given a variable Y with some missing values. It can be stated that these values are MCAR if the probability of missing data regarding Y is unrelated to the value of Y itself or to any other variable in the data set.
- **Missing at Random (MAR)**
  Compared to MCAR this is a weaker assumption. The probability

of missing data for a variable Y is again unrelated of the value of Y but might be related to some other observed data. More formally speaking:

$$P(Y_{missing}|Y, X) = P(Y_{missing}|X) \tag{6.1}$$

- **Missing not at Random (MNAR)**
  Missing values do depend on some unobserved values.

If one of the first two cases, namely MCAR or MAR, is observed it is legitimate to remove data with missing values. If MNAR is detected, removing missing values can bias the prediction model.
For the use case of traffic flow imputation one can say that different TFS might correlate with each other. But there is at all no dependency between the missingness of the data samples itself. So a missing sample does not depend on its value nor on the measurements of other correlated TFS therefore this data samples are MCAR or MAR. Thus, from the pool of several methods for missing data handling, as shown in Figure 6.8 can be chosen.

6.3.2 Data deletion

Deletion of data can again be split in three different sections. For the purpose of traffic flow imputation the method of deleting columns is the most interesting one. This means, not a single invalid measurement sample is deleted, but the whole TFS and its measured values are left out if the missing value percentage exceeds a certain threshold.
In a later Chapter a deeper insight will be given how this column wise deletion is processed because it strongly depends on the chosen imputation technique and validation requirements.

[2] https://miro.medium.com/max/1222/1*_RA3mCS30Pr0vUxbp25Yxw.png (Accessed on: 2019-09-16)

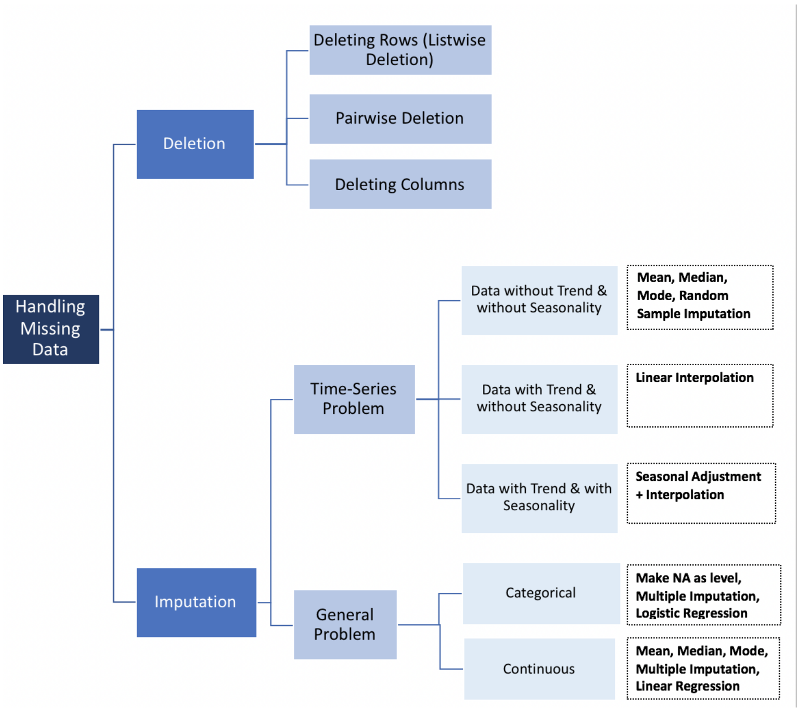

Handling Missing Data
Deletion
Deleting Rows (Listwise Deletion)
Pairwise Deletion
Deleting Columns
Imputation
Time-Series Problem
Data without Trend & without Seasonality
Mean, Median, Mode, Random Sample Imputation
Data with Trend & without Seasonality
Linear Interpolation
Data with Trend & with Seasonality
Seasonal Adjustment + Interpolation
General Problem
Categorical
Make NA as level, Multiple Imputation, Logistic Regression
Continuous
Mean, Median, Mode, Multiple Imputation, Linear Regression

Figure 6.8: Different methods for handling missing data[2]

6.3.3 Imputation

IterativeImputer

As discussed in Chapter 6.1 a multivariate feature imputation is preferred over a univariate method because there might exist some strong correlation between different TFS which can be used for estimating missing values. Therefore scikit comes with a multivariate imputer approach called *IterativeImputer*[3]. This imputer models each feature with missing values as a function of other features and uses the estimated values for imputation. This is done in an iterated round-robin fashion where by one iteration each feature column becomes the output and all the other feature columns act as input. One iteration is defined as that each feature column acts as an output

[3] `https://scikit-learn.org/stable/modules/impute.html#multivariate-feature-imputation` (Accessed on: 2019-09-16)

once. The maximum imputation rounds can be defined via the parameter max_iter.
This basic imputer can be used for different imputation strategies like AMELIA, MICE, missForrest just by passing in different regressors.

Single vs. Multiple Imputation

Single imputation is per definition running the whole imputation process once an tread the imputed values as real values afterwards. The drawback of this imputation method is that the uncertainty increases when the rate of the missing value increases. To overcome this problem, multiple imputation can be used. Therefore the whole imputation process is repeated multiple times resulting in multiple imputed data sets. This imputation strategy consists of three phases [4]:

- **Imputation phase**
  In this phase the imputation is done on several copies of the original data set. E.g. regression analysis is used to predict missing values.

- **Analysis phase**
  The analysis phase carries out the statistic, like mean, variance etc, of the imputed values for all different imputed data sets.

- **Pooling phase**
  Finally, the pooling phase creates the overall estimation of the imputed values by combining the results of the different imputed data sets. If the estimates are pooled by Rubin's Rules the result values are calculated by averaging the different parameter estimations of the different imputed data sets as given in Formula 6.2. The total variance can be calculated by combining the within imputation variance and the between imputation variance as stated in Equation 6.5

$$\theta_{MI} = \frac{1}{M}\sum_{i=1}^{M}\theta_i \tag{6.2}$$

[4] https://www.iriseekhout.com/missing-data/missing-data-methods/multiple-imputation/ (Accessed on: 2019-09-17)

$$Var_{within} = \frac{\sum_{i=1}^{M} SE_i^2}{M} \tag{6.3}$$

$$Var_{between} = \frac{\sum_{i=1}^{M} (\theta_i - \theta_{MI})^2}{M-1} \tag{6.4}$$

$$Var_{MI} = Var_{within} + (1 + \frac{1}{M}) Var_{between} \tag{6.5}$$

Where SE is the standard error, and M the number of imputed data sets.

6.4 Multivariate imputation by chained equations (MICE)

The *IterativeImputer* is inspired by the R MICE package but differs from it in a way that only the output of a single imputation is returned and not the outputs of multiple imputations. Still one can make use of it as a base imputer and expand it so that multiple imputations can be performed. For the following section Buuren and Groothuis-Oudshoorn, 2010 and Azur et al., 2011 serves as a template of how to expand this *IterativeImputer* to a specific multiple imputation approach called MICE.

6.4.1 Notation

X_j with $(j = 1, ..., p)$ stands for one incomplete feature vector in the data set where $X = (X_1, ..., X_p)$ represents the whole data set. According to the use case of imputing missing traffic flow values one can say that each feature vector represents a specific TFS for a given time period. The observed values of a specific TFS vector j is denoted by X_j^{obs} whereas the missing values are characterized by X_j^{mis}. Therefore $(X_1^{obs}, ..., X_p^{obs})$ and $(X_1^{mis}, ..., X_p^{mis})$ stands for the observed and missing measurement values of the whole TFS data set. X_{-j} denotes the subset of $p - 1$ feature vectors excluding X_j.

Q has the form of a multivariate vector and represents the model parameters like regression coefficients.
The number of different imputations is equal to $m \geq 1$.

6.4.2 Bayesian Regression

As mentioned in Chapter 6.3.3, in the imputation phase a regression model tries to predict the missing values X^{mis} of the data set X. In the case of the MICE imputation a Bayesian regression model called *BayesianRidgeRegression*[5] form the sklearn library is used.
Against a classical linear regression approach the Bayesian regression method, as cited in Kruschke, Aguinis, and Joo, 2012, uses probability distributions rather than point estimates as stated in Equation 6.6

$$y \sim \mathcal{N}(\beta^T X, \sigma^2 I) \tag{6.6}$$

Where y is sampled from a normal distribution where the mean is characterized by the multiplication of the weight matrix and the feature matrix and the variance is characterized by the multiplication of the square of the standard deviation and the Identity matrix.
Not only the model output is sampled from a probability distribution but also the model parameters are generated out of a posterior distribution as stated in equation 6.7

$$P(\beta|y, X) = \frac{P(y|\beta, X) * P(\beta|X)}{P(y|X)} \tag{6.7}$$

Regarding the Bayesian Ridge Regression, the variance of the distribution from where the output is sampled, is abstracted as a random variable which has to be estimated from the data as stated in formula 6.8:

$$y \sim \mathcal{N}(\beta^T X, \alpha) \tag{6.8}$$

[5] https://scikit-learn.org/stable/modules/linear_model.html#bayesian-ridge-regression (Accessed on: 2019-09-17)

The prior for the coefficient β for the Bayesian Ridge Regression is given by a spherical Gaussian:

$$P(\beta|\lambda) = \mathcal{N}(\beta|0, \lambda^{-1}I) \tag{6.9}$$

Further, the priors over α and λ are chosen to be gamma distribution, the conjugate prior for the precision of the Gaussian.
During the imputation process the parameters β, α and λ are estimated jointly, where α and λ are calculated by maximizing the log marginal likelihood 5.

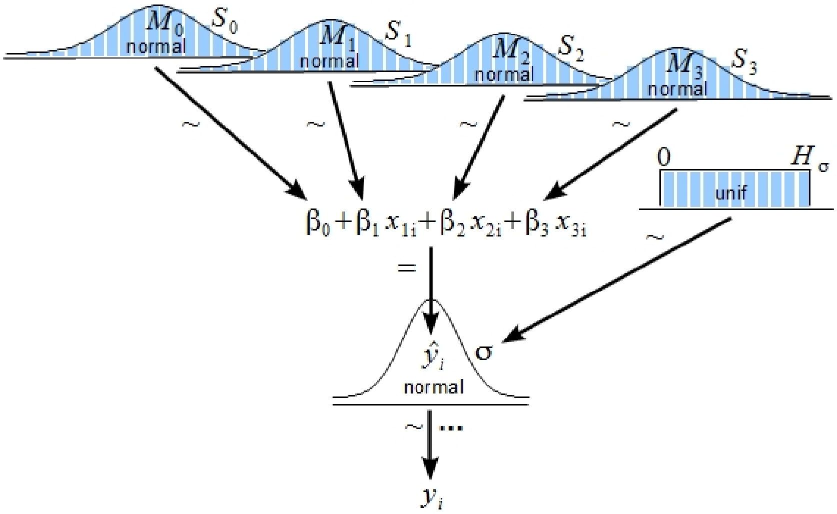


Figure 6.9: Diagram for multiple linear regression cited in Kruschke, Aguinis, and Joo, 2012

6.4.3 Method

Data Set

For generating the feature matrix the whole traffic flow measurements are split up after years. Therefore 3 different feature matrices with missing values exist. Further, three different imputation models are trained based on the specific data set. Also hyperparameter search is applied to each model independently.

Training

The chained equation approach can be split into six steps:

- In the first imputation step, all missing values are masked so that for further turns the imputed values still can be recognized and the mean over X_j^{obs} is imputed. These imputation can be thought of as place holders.
- In the second step, the place holder mean imputations for X_j is set back to missing.
- In the third step X_j^{mis} are regressed on X_{-j} via the *BayesRidgeRegression* approach. Not all X_{-j} features are taken for imputation but just the n strongest correlated feature vectors. The Pearson correlation coefficient is used for estimating these n vectors.
- In the fourth step the missing values X_j^{mis} are replaced with the regressed values.
- In step five step two to step four is repeated for all X_j^{miss} with $(j = 1, ..., p)$
- In step six the imputation process as stated in step 5 is repeated for a given number of cycles with the imputations being updated at each cycle.

Regarding the last step one can also track the change of the within variance and determine some threshold ϵ for the stop criterion instead of a fixed number of cycles. This ϵ strongly depends, in the use case of traffic flow imputation, on the number of missing values of a given TFS. So therefore no global ϵ exists.

As mentioned in the previous section, the hyperparameter search is applied to each model independently. The following parameters have to be optimized:

- **number_nearest_features**
  stands for the number of features which are chosen as input for the regression analysis.
- **max_imputation_cycles**
  together with the parameter *diff_variance_threshold* it describes the

maximum number of regression runs per imputation run.

- **diff_variance_threshold** early stopping criterion tracks the difference in the variance between regression runs and stops when $diff\_variance\_threshold$ is reached.
- **nr_multiple_imputation** number of imputation runs to fulfill the multiple imputation approach.

Validation

Because the whole traffic flow data set is split after years for the training process, the validation is also separately done for each of the three models. Therefore a given percentage of non missing values for each traffic flow station is randomly removed and has to be predicted. Also hyperparameter search with the Grid Search strategy is applied to each of the MICE models. The following hyperparameters can be estimated:

hyper parameter	**possible values**
number_nearest_features	5,10,15,20,25
max_imputation_cycles	10,15,20,25,30
diff_variance_threshold	1e3, 1e2, 1e1, 1
nr_multiple_imputation	5,10,15,20

In the experiment a missing rate between 10% and 70% with a step size of 10% is applied. To compare the different experimental results, the root mean squared error (RMSE) is used as evaluation metric as stated in Equation 6.10:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{6.10}$$

where $\hat{y}_i$ and y_i are the estimated and ground truth values of the ith imputed value. N represents the total number of imputed values. The evaluation is not only done on the whole imputed data set, considering all different feature vectors but also have a view on each traffic flow station separately.

Results

The estimated hyperparameter values for the different models are mentioned in Table 6.1.

hyper parameter	**model**		
	2015/2016	2016/2017	2014/2018
number_nearest_features	20	20	20
max_imputation_cycles	10	10	10
diff_variance_threshold	-	-	-
nr_multiple_imputation	10	10	10

Table 6.1: Selected hyper parameters for the three imputation models

Because of multiple imputation runs, one can first track the within variance of the imputed values. One might suggest that a small within variance of a given traffic flow station would mean a strong correlation to some other traffic flow stations and also a small missing rate of traffic flow values. A more general indication, not on the level of a single imputed value but on the level of traffic flow stations can be considered by the RMSE per station.

The validation result for each of those models with the according hyperparameters over the missing value percentages are shown in Figure 6.10.

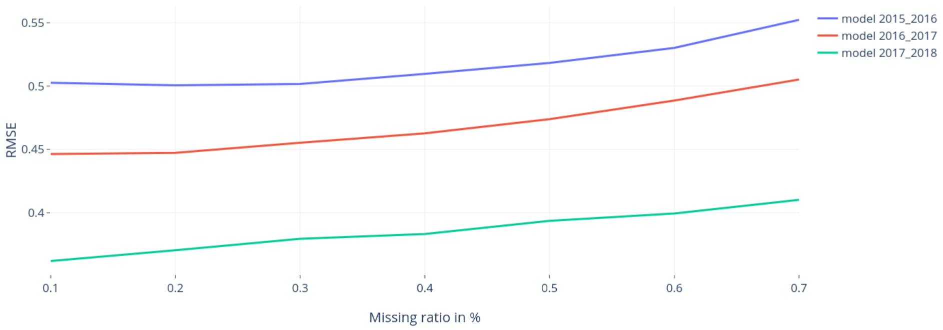


Figure 6.10: RMSE over missing value ratios of imputation models

The different MICE imputation models remain stable regarding the RMSE by choosing different missing rates. Also the overall absolute RMSE value is very low. A deeper insight is given in Table 6.2 where the RMSE of each traffic flow station is calculated separately.

traffic flow station	missing ration	RMSE
363.22	10%	0.18
	30%	0.2
	70%	0.26
801.31	10%	1.22
	30%	1.29
	70%	1.52

Table 6.2: RMSE evaluation per traffic flow station in 2015

traffic flow station	missing ration	RMSE
707.41	10%	0.2
	30%	0.21
	70%	0.25
824.36A	10%	1.1
	30%	1.1
	70%	1.3

Table 6.3: RMSE evaluation per traffic flow station in 2016

traffic flow station	missing ratio	RMSE
806.11	10%	0.18
	30%	0.2
	70%	0.24
824.34	10%	1.0
	30%	1.02
	70%	1.04

Table 6.4: RMSE evaluation per traffic flow station in 2017

As an example only the traffic flow stations with the lowest and highest RMSE per year are compared. One can see in Figures 6.2, 6.3 and 6.4 that over all three years the RMSE differ by a factor of lower equal 10. By

investigating the reasons of that, it can be stated that not only the overall missing rate of traffic flow values per station and year is decisive but the combination of the missing rate and the correlation to other traffic flow stations.
Considering station number 363.22, the overall missing rate is about 20% and the correlation factors to the 10 closest stations regarding the Pearson correlation coefficient is between 0.96 and 0.94 which is quite high. Against that for station 801.31 the missing rate is about 85% and also the correlation to other stations is very low with about 0.53 to 0.2 for the 10 closest stations. As stated above not only a low missing rate is relevant for a low RMSE but the combination with strong correlated stations with preferably low missing value rates. By looking at station number 707.41 and 824.36A for the year 2016, the missing rates are likely the same with about 0.5%. But 707.41 correlates much stronger to other stations than 824.36A does. So therefore the RMSE for 707.41 is much lower than the RMSE for 824.36A
Comparing station number 806.11 and 824.34 for the year 2017 the influence of strong correlated features again can be viewed. Although the missing rate for 806.11 with 25% is more than 10 time higher as the missing rate for 824.34 the RMSE is much lower just because of stronger dependencies to other stations with lower missing rates.

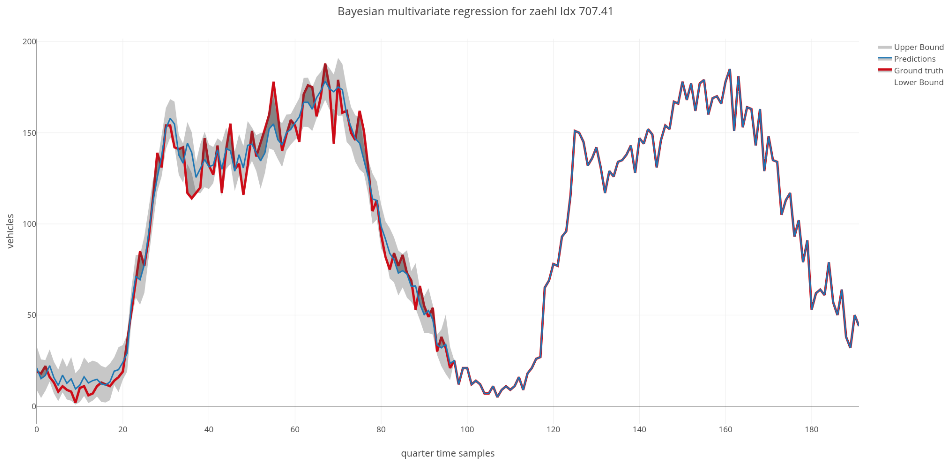


Figure 6.11: Imputation sequence of traffic flow station 707.41

Another issue which arise for some traffic flow stations can be viewed in

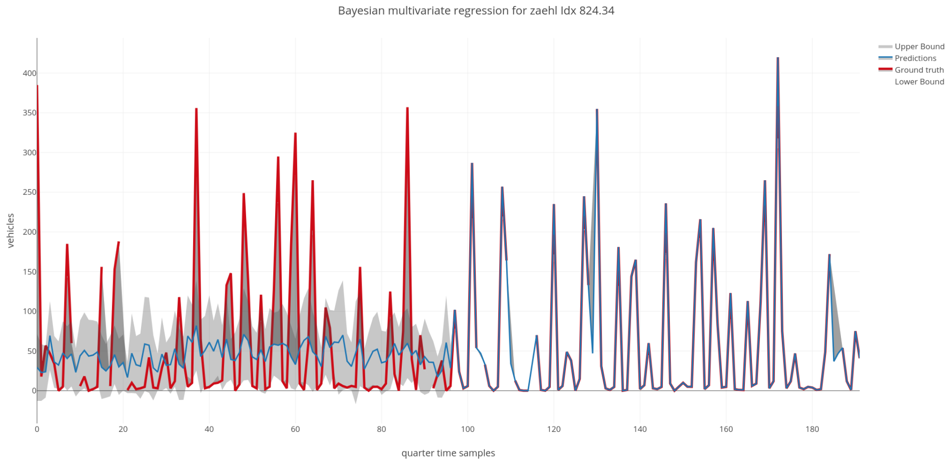


Figure 6.12: Imputation sequence of traffic flow station 824.34

Figure 6.12. One can see that compared to other stations, as shown in Figure 6.11, no classical daily pattern can be recognized. Also the variance of the imputed values and the deviation between imputed values and ground truth values is much higher as for those traffic flow stations where an daily pattern occur. One explanation might be that most of these confusing patterns occur on street segments where multiple driving lanes in the same direction are tracked separately. So on the 2nd lane or outside lane a regular daily traffic pattern can hardly be detected. Another simpler explanation is that wrong measurement values are outputted by these stations.

7 Traffic Accident Prediction

As mentioned in the introductory chapters, one of the main objective of this paper is to evaluate different classifier approaches for real-time crash likelihood prediction based on different data sources.
At first some basic statistics of the merged data set are generated in an explanatory manner. In a second step a negative sampling strategy for non-crash events has to be deployed. Based on the chosen classification model input features might also have to get scaled or one hot encoded.
Given the preprocessed feature matrix, the classifiers hyperparameters have to be estimated based on the given evaluation metric. These classifiers are not only trained and tested on the whole feature set but also on subsets of them. For the traffic flow features the imputed data set from Chapter 6 is used. At the end the different results get compared in terms of different evaluation metrics to figure out how well and robust the different classifiers behave.

7.1 Exploratory data analysis

7.1.1 Crash statistics

In this section, some basic statistics regarding different attributes in the crash data set are mentioned. Between 01.01.2015 and 31.12.2017, 5416 accidents in the streets of Graz occurred. The number of accidents per month can be viewed in Figure 7.1. Also the number of injured and deadly injured accident participants is visualized over the three years. Injured participants again can be split into slightly injured and seriously injured accident participants as displayed in Figure 7.2.

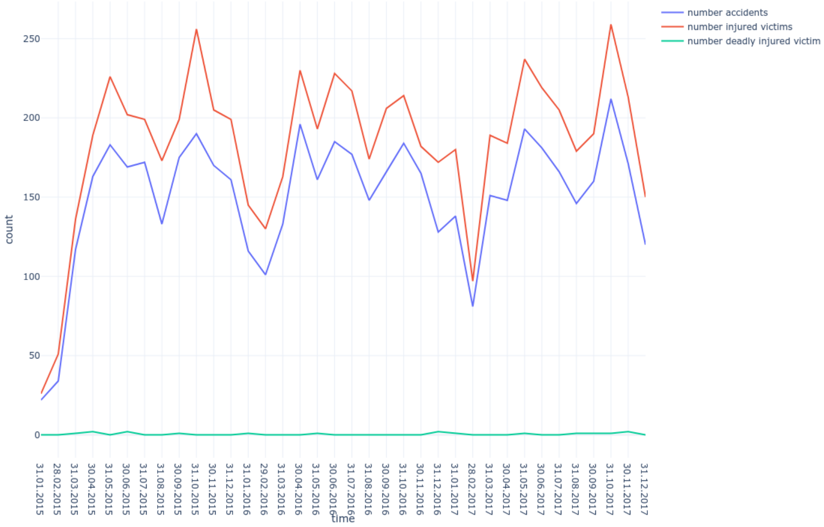


Figure 7.1: Accidents per month between 01.01.2015 and 31.12.2017

It is shown that the absolute numbers of accidents per year over the 3 years slightly increased. In the winter months between December and March the lowest accident rates can be mentioned. Against that the accident rate rises in the summer and autumn months between May and October. The relation between accidents and injured participants as well as between slightly and seriously injured participants can be viewed as constant over the three years.

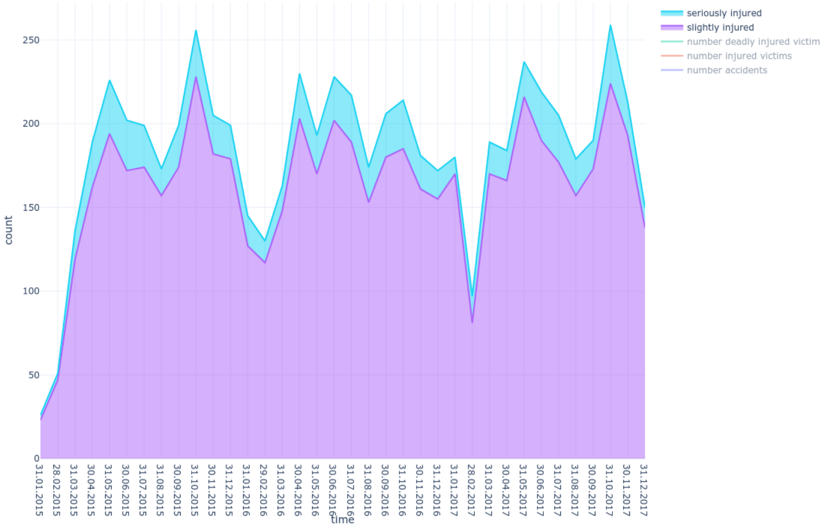


Figure 7.2: Stacked number of slightly and seriously injured accident victims between 01.01.2015 and 31.12.2017

Each accident is classified in the crash data set into 13 categories. The number of accidents per class can be viewed in Figure 7.3. Priority violation, including disregard red traffic lights and negligence or distraction of the driver followed by too low safety distance are the most important reasons why an accident happens in Graz.

In Figure 7.4 the influence of precipitation and the influence of alcohol over the different districts of Graz is visualized. One can see that the percentage of accidents under rainy or snowy road conditions is very low compared to the overall accidents per district. In the district of St. Leonhard the

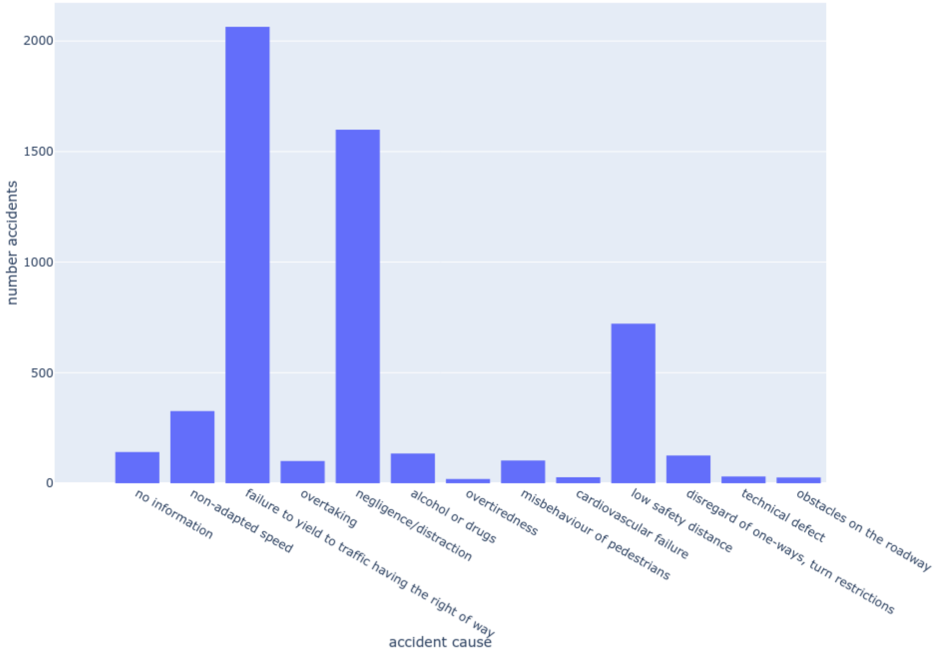
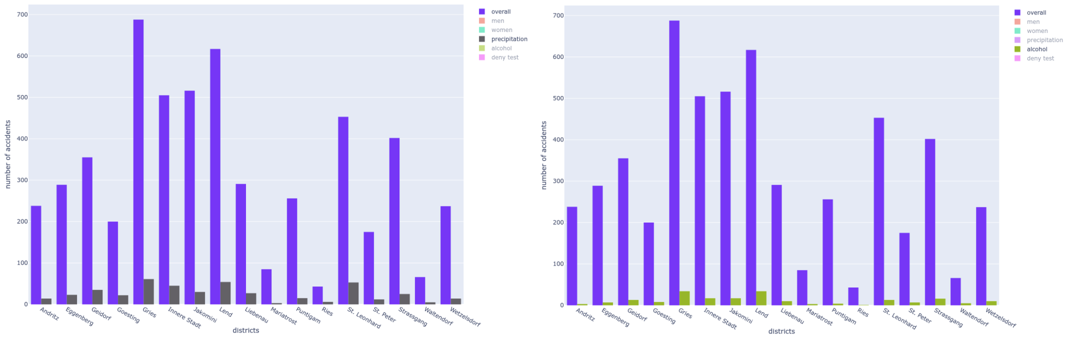


Figure 7.3: Cause of accidents categories

percentage seems a bit higher than compared to the other districts. Also accidents under alcohol influence are relatively rare compared to the overall number. The most accidents therefore can be figured out in the districts of Gries and Lend.

Another interesting statistic is the number of accident participants (not accident causers) per gender over the 17 districts as visualized in Figure 7.5. In every district more male than female are involved into crashes. Especially in the inner districts Gries, Jakomini and Lend the number of male participants is much higher than compared to the outer districts.



(a) (b)

Figure 7.4: (a) accidents under precipitation (b) accidents under alcohol influence

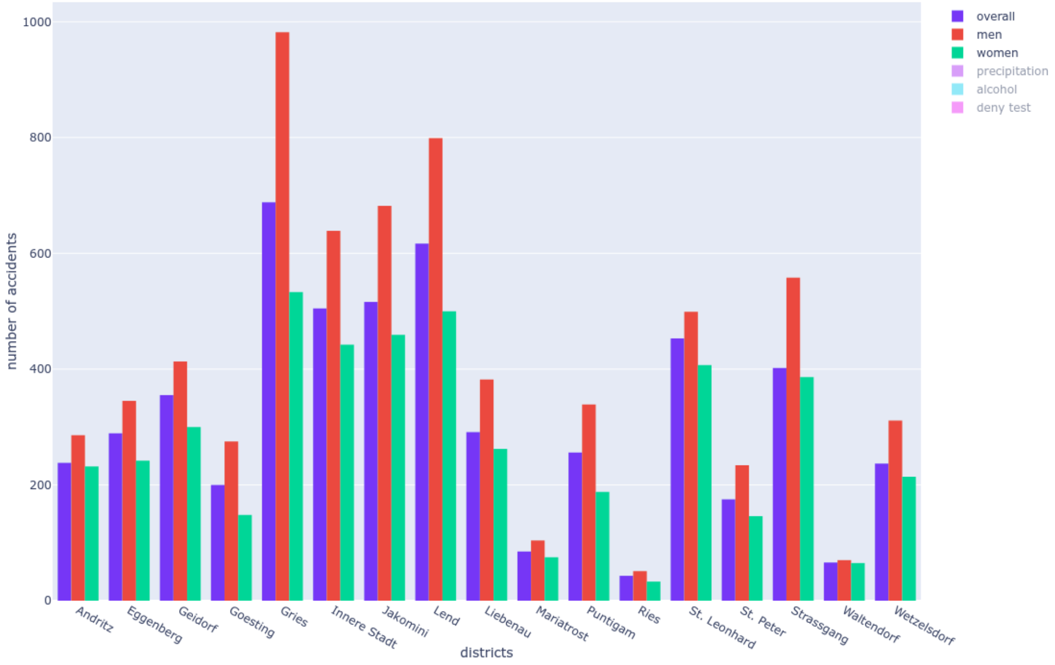


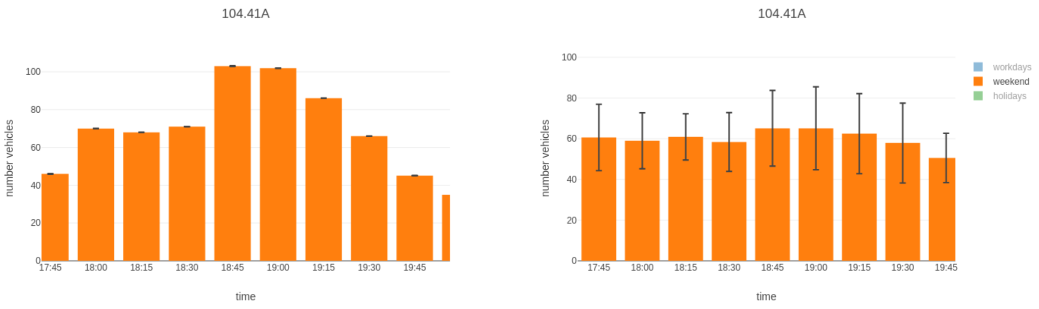
Figure 7.5: Number of accident participants per gender

7.1.2 Correlation between traffic flow and accidents

As stated in Chapter 2.2, not only the absolute traffic flow values for a given street segment seems to have an big impact regarding crash accidents but also the change of these values over time. In this section, two street links with a high accident probability over the years are considered and the traffic flow values for these links are analyzed at the time of crashes.

Joanneumring

The Joanneumring is a one-way street link with 3 lanes, each of them tracked separately by the measurement stations 104.41A, 104.42A and 104.43A. On the 7th of May 2017 at 06:45 p.m. a crash happened on this road link. By considering the traffic flow values of the three measurement stations as visualized in Figure 7.6, 7.7 and 7.8 one can see that for 104.42A the absolute traffic flow one hour before the crash is greatly increased compared to the average traffic flow in the same month. At the time of crash the flow dramatically decreased. Station 104.41A and 104.43A therefore show no relevant deviation from the months average till 06:45 p.m. where the flow significantly increased.

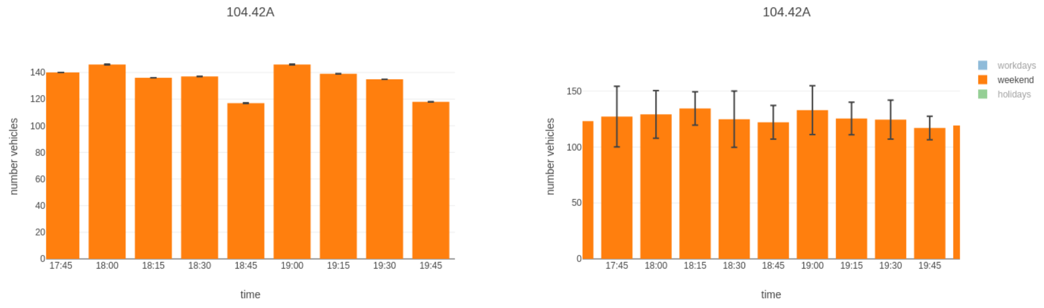


(a)

(b)

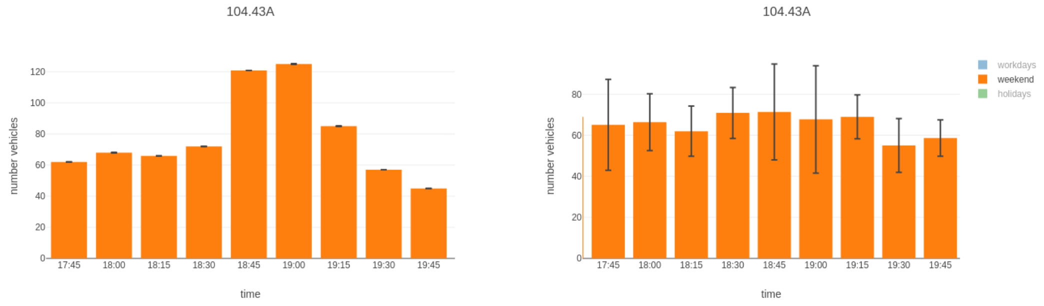
Figure 7.6: (a) traffic flow of 104.41A on the day of crash (b) average traffic flow of 104.41A over the month with standard deviation

Another crash on the 7th of June 2017 at 03:31 p.m. as Figures 7.9, 7.10 and 7.11 point out, that the traffic flow tracked at 104.42A and 104.43A confirms



(a) (b)

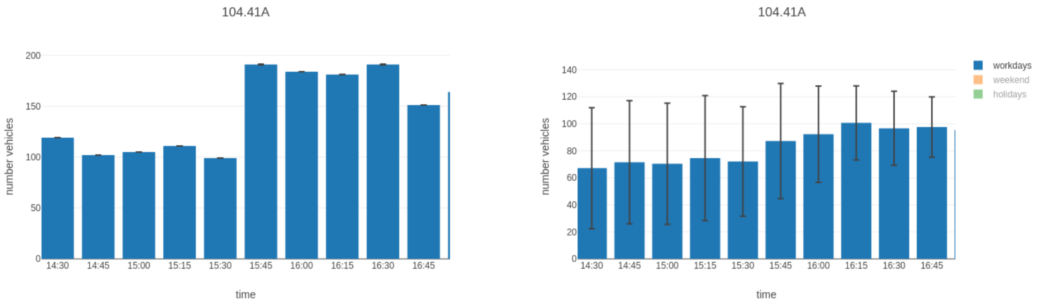
Figure 7.7: (a) traffic flow of 104.42A on the day of crash (b) average traffic flow of 104.42A over the month with standard deviation



(a) (b)

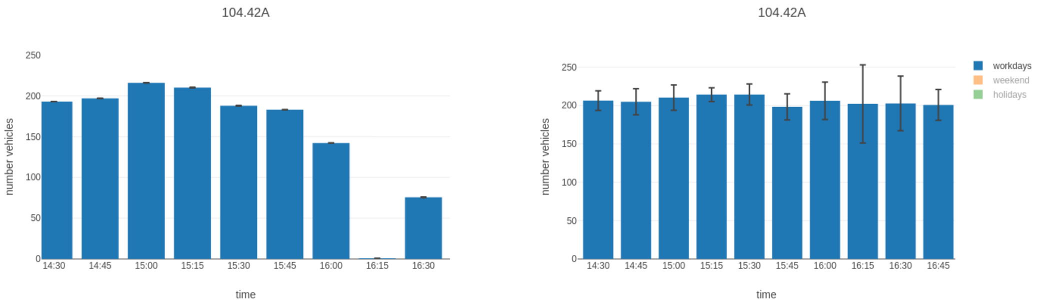
Figure 7.8: (a) traffic flow of 104.43A on the day of crash (b) average traffic flow of 104.43A over the month with standard deviation

with the average traffic flow of the June 2017. Against that, 104.41A shows a high flow rate also one hour before the accident. Short time after the accident, at 03:45 p.m. 104.41A and 104.43A a dramatically increase of the flow can be stated. One reason for this might be because of the dispersion of the traffic jam.



(a) (b)

Figure 7.9: (a) traffic flow of 104.41A on the day of crash (b) average traffic flow of 104.41A over the month with standard deviation

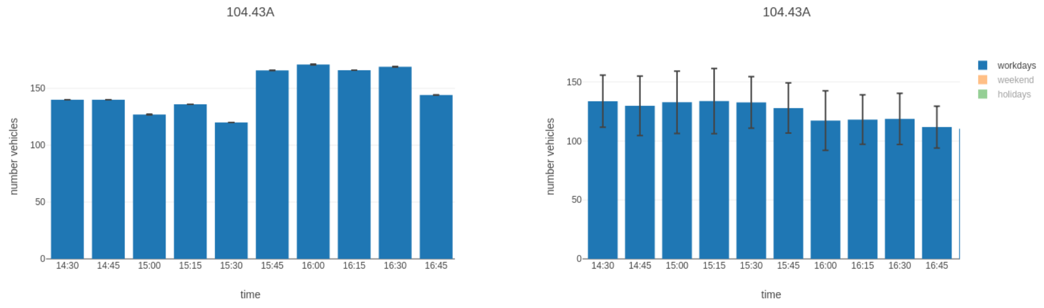


(a) (b)

Figure 7.10: (a) traffic flow of 104.42A on the day of crash (b) average traffic flow of 104.42A over the month with standard deviation

Weinzötlstraße

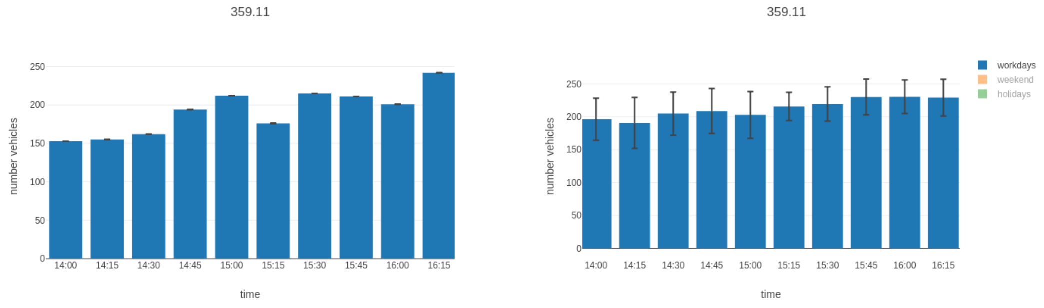
The Weinzötlstraße is a two-way street with 2 lanes, one leads to the north one to the south. The street lane to the north is tracked by the station $359.11A$, the lane to the south by station 359.11. One representative crash is taken from the 29th of May 2017 at 03:00 p.m. As visualized in Figure 7.12 and 7.13, a high increase of the flow can be noticed at station 359.11 in the 15 minute interval prior the crash. The absolute flow value matches the monthly average of May 2017. $359.11A$ matches at all the monthly average.



(a)

(b)

Figure 7.11: (a) traffic flow of 104.43A on the day of crash (b) average traffic flow of 104.43A over the month with standard deviation

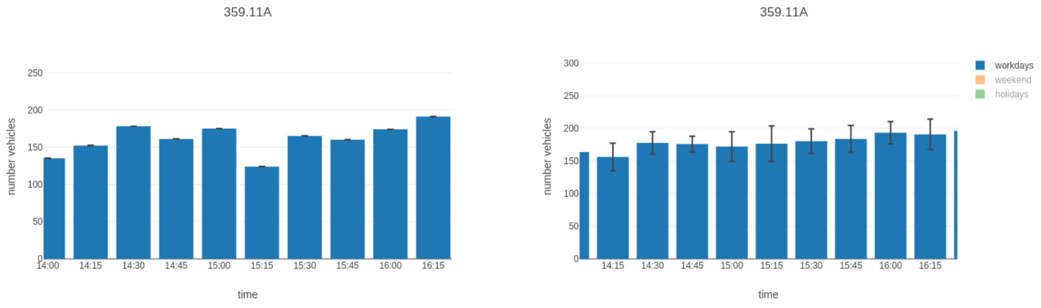


(a)

(b)

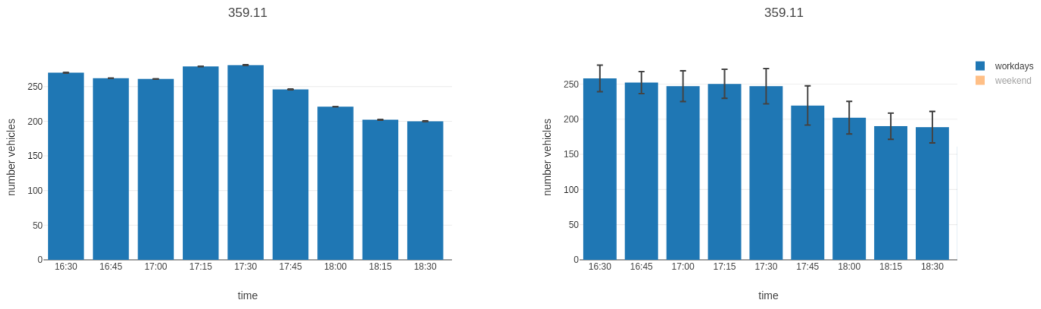
Figure 7.12: (a) traffic flow of 359.11 on the day of crash (b) average traffic flow of 359.11 over the month with standard deviation

A second crash happened on the 30th of March at 05:30 p.m. As one can see in Figure 7.14 and 7.15 the absolute flow value clearly increased from 05:00 p.m. to 05:15 p.m. and exceeds the average flow of March 2017 at station 359.11. Also around 05:30 p.m. the flow is still higher as the months average. Clearly the same can be said about station 359.11A. The traffic flow of both stations increased 30 minutes earlier than the months average of both stations show.



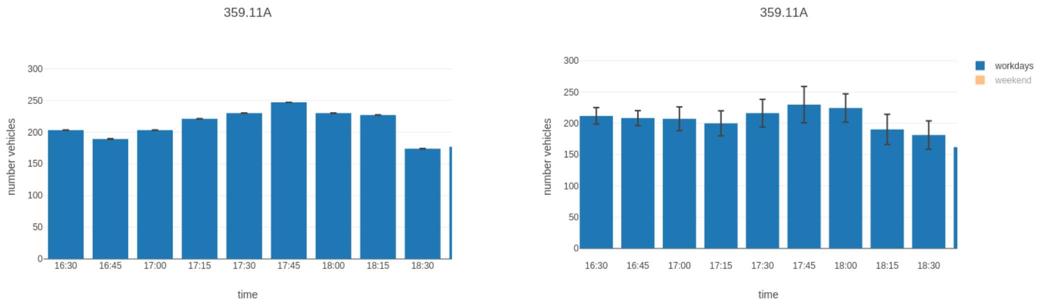
(a) (b)

Figure 7.13: (a) traffic flow of 359.11A on the day of crash (b) average traffic flow of 359.11A over the month with standard deviation



(a) (b)

Figure 7.14: (a) traffic flow of 359.11 on the day of crash (b) average traffic flow of 359.11 over the month with standard deviation



(a) (b)

Figure 7.15: (a) traffic flow of 359.11A on the day of crash (b) average traffic flow of 359.11A over the month with standard deviation

7.2 Accident statistic by beginning rainfall

In this Section the influence of beginning precipitation on road accidents is considered. The weather data is, as mentionde in Section 4.5, aggregated into 1 hour measurements over the years. The measurement unit is stated in mm. Given these measurements the beginning of rainfall and snowfall can be determined when in the prior hour no precipitation is measured. Some basic statistics can be stated:

- Between 01.01.2015 and 31.12.2017 5.416 accidents happened in Graz. 710 accidents happened when rain or snowfall was tracked and interpolated for a given roadway. Out of these 710 accidents, 184 accidents happened by beginning rain- or snowfall.
- Overall 26.280 measurement values for each of the different stations are within the data set. Approximately on 640 different timestamps the beginning of precipitation can be tracked.

Considering these facts the probability of an accident under the condition that it just started to rain or snow is about 3.4%. The marginal probability of beginning rain- or snow fall is about 2.45% taking the mean value of the different weather stations. It can be stated that the beginning of rain- or snowfall does not have a great impact on the crash likelihood in Graz.

7.3 Data set generation

To train the different accident prediction models, the first step is to generate a training, test and validation data set. Therefore the different data sources have to be merged together. Each record of the merged data set represents one crash sample. For each crash sample, n non-crash samples should be generated. For this thesis, the matched case control strategy (Ahmed and Abdel-Aty, 2013, Ke et al., 2019) with small adaption is used to select the negative samples.
As mentioned above all different data sources are merged but the classifiers not only get trained on the whole feature set but also on subsets.

7.3.1 Positive samples generation

In Figure 7.1 the extracted features are shown. Depending on feature type, the categorical features have to be one hot encoded. The accidents date and time features are one hot encoded in terms of the year, month, day, hour and minute. Depending on the classifier, the numerical features have to get scaled because of the different value ranges.
One can also see that for some features, especially features regarding the the GIP road network graph, still missing values occur which are hardly to impute. So therefore classifiers, which are able to handle missing values internally, would be preferable.
Because traffic flow measurements can not directly be calculated for each street link, the traffic flow values for each measurement station 15min and 30min prior to the occurrence of the crash is joint to the specific entry. As stated in Chapter 6.4.3 measurement stations do not deliver traffic flow values over all three years but only in a specific time range. Again missing values will occur which should be handled by the classifier itself. The schema with all necessary traffic flow features is provided in Appendix 9.1

7.3.2 Negative samples generation

For negative sampling the matched case control strategy is applied. For each crash sample 3 non-crash entries are sampled with the following matching rules:

- **Road Link:** If an accident occurred on road A, randomly pick a road link B which is similar to link A regarding the link attributes $average\_speed$, $width$, $page - rank$, $centrality$ and $curvature$. Similar links are calculated via the K-Means cluster functionality of the sklearn library. [1] Changing the road link will cause the change of road related features, district related features and also minor changes in weather related features.

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html` (Accessed on: 2019-11-25)

feature name	feature type	details	data source
accident_date	categorical	01.01.2015 - 31.12.2017	Crash data set
accident_time	categorical	24 categories	
is_holiday	categorical	2 categories	
lighting_conditions	categorical	4 categories	
fog	categorical	2 categories	
wind	categorical	2 categories	
roadway_ceiling	categorical	5 categories	
road_condition	categorical	5 categories	
max_speed	numerical	0-80 km/h	
is_junction	categorical	2 categories	GIP graph
average_speed	numerical	0-100 km/h	
number_lanes	numerical	0-6 lanes	
road_width	numerical	0-22 meters	
road_surface	categorical	2 categories	
page-rank	numerical	0.03 - 0.86	
centrality	numerical	0 - 1	
curvature	numerical	1 - 2	
slope	numerical	0% - 18.56%	
district	categorical	17 categories	Open Data Graz
population_density	numerical	663 - 10423 resident per km^2	
temperature	numerical	-14.3°C - +33.3°C	ZAMG data
precipitation	numerical	0 - 15.1 mm	

Table 7.1: Accident crash training sample without traffic flow features

- **Datetime:** If an accident occurred on road A at a timestamp $T1$, randomly take the same road link or one of the similar road links one hour prior and later if no accident occurred. Changes in date and time will cause the change of traffic flow related features and changes for weather related features.

Finally, the data set contains about 3 times negative samples than positive ones. The overall data set contains 37624 examples.

7.4 Model selection

For the accident likelihood estimation one of the most common concepts is the use of Decision Trees (DT) and Random Forests (RF) as mentioned in Yuan et al., 2017 or Ke et al., 2019. Regarding the input data set of this thesis, DTs and RF have some great advantages over other approaches like Neural Networks (NN):

- Good performance also on small data sets
- Works with continuous and categorical variable at the same time.
- Can handle missing values in the data set
- Does not require feature scaling

For the crash likelihood estimation in this thesis, a Gradient Boosting (Friedman, 2001) library called XGBoost (T. Chen and Guestrin, 2016) is used as the primary classification model.

7.4.1 Hyperparameter Search

The hyperparameters for the XGBoost classifier are evaluated via the GridSearchCV[2] method, offered by the sklearn library.
In a first step the whole data set, which was generated in Chapter 7.3, is split into a training and test set by a ratio of 4:1. For the grid search approach the training set is internally split again into 5 folds where by each iteration one fold is left out as validation set. The parameter search is done by searching the whole hyperparameter space for the best cross validation score. In this case the Area Under the Receiver Operating Characteristic Curve (ROC AUC) score is used as evaluation metric.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (Accessed on: 2019-11-25)

7.5 Experiments and Results

In this section, different experiments on the whole and on selected features of the data set are performed. The XGBoost classifier is trained and evaluated for each experiment separately. Therefore the hyperparameters have to be separately evaluated too.
The test set is once generated by randomly sample over the three years and once by taking the whole year 2017 into account.

7.5.1 Feature selection

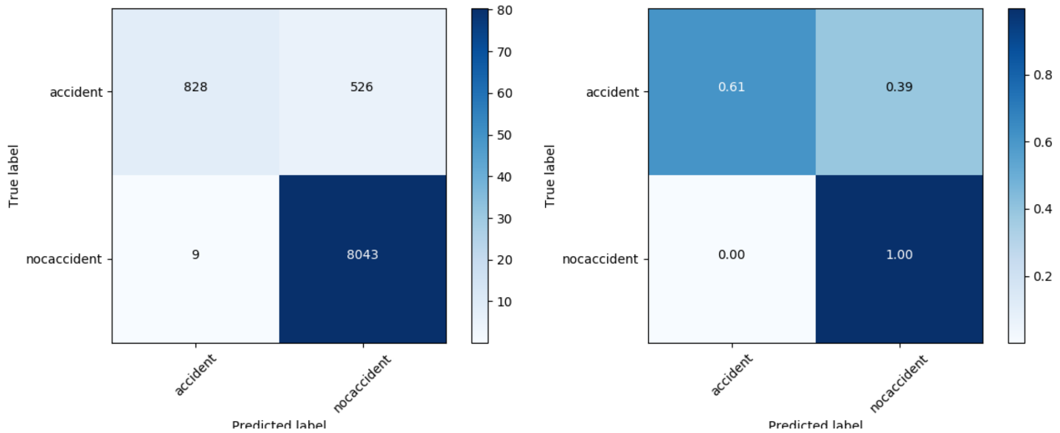
The crash likelihood estimation is processed only for accident, road link, population and weather related features, extracted form the overall data set as shown in Table 7.1. The traffic flow measurements are ignored in this initial experiment. For this data set the estimated model parameters are cited in Table 7.2.

hyper-parameter	**estimated value**
objective	binary:logistic
learning_rate	0.1
max_depth	10
min_child_weight	2
subsample	1
colsample_bytree	0.7
n_estimators	350

Table 7.2: Hyper-parameters for the XGBoost classifier

Testing the classifier with the retained test set, different evaluation metrics can be generated. As one can see in Figure 7.16 the TP value is quite low whereas the FP value is high. So the correct classification rate of accidents itself reaches not more than 61%, whereas the correct classification of non-accidents is straight 100%. By interpreting the precision-recall functions in Figure 7.17, the precision for class accidents drops dramatically at a recall of 0.8. Instead the curve for non-accident still keeps a constant precision score

of 1.0 over the whole recall space.
The XGBoost classifiers prediction method internally classifies samples by a prediction probability of greater 50% as non-accident. This is a quite low for samples of class non-accident. Regarding the output of Figure 7.17, one can try to increase this threshold.
Figure 7.18 visualizes the confusion matrix where the threshold for classifying samples as non-accidents is increased to 85%. The TPR increased to about 82% and the FNR hardly decreased to about 97%.
This seems to be a much more acceptable result as it is more important to increase the recall for crash events as to have a high precision score. In other words, trying to figure out most crash events by simultaneously classify some non-crash events wrongly as crash events. The different evaluation outputs regarding the probability threshold for the class non-accident can be viewed in Table 7.3.

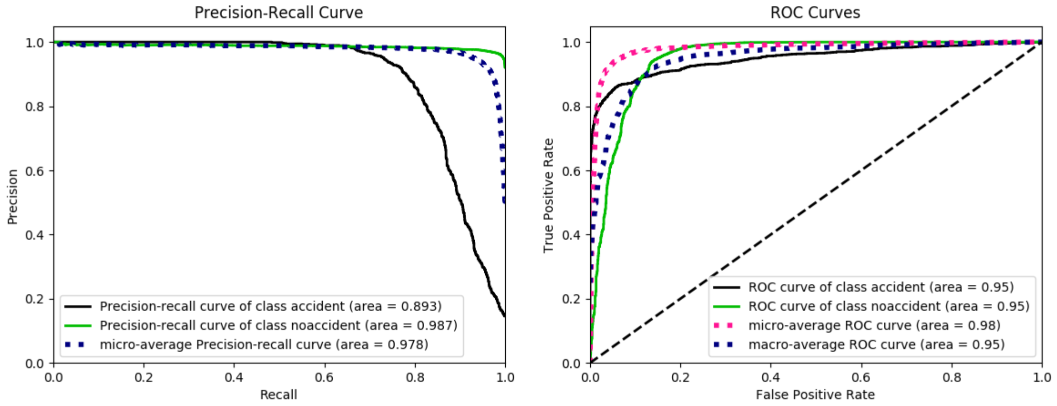


(a)

(b)

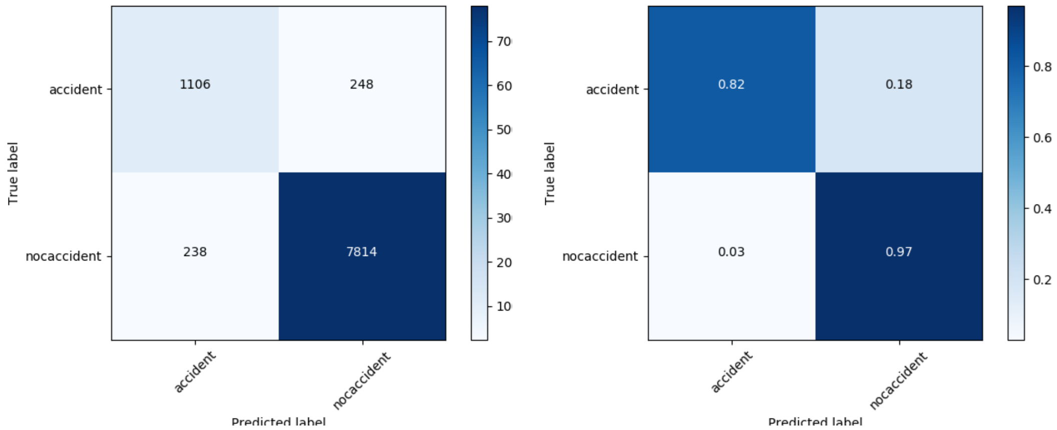
Figure 7.16: (a) confusion matrix absolute values (b) confusion matrix normalized

The XGBoost classifier is also utilized to calculate the feature importance of the explanatory variables as shown in Figure 7.19. Only the most relevant features are displayed here. The feature importance is based on the *Gain* importance metric. This metric implies the relative contribution of a specific feature to the models decision by taking the feature's contribution to each DT in the forest. It can be observed that the different districts might have a great impact on the output result. Also street graph related features like the



(a) (b)

Figure 7.17: (a) precision-recall (b) ROC



(a) (b)

Figure 7.18: (a) confusion matrix absolute values (b) confusion matrix normalized

number of lanes, PageRank and centrality are important features.
Another evaluation can be made by permuting the input features separately and looking at how much the score (accuracy, F1 score etc.) decreases when this feature is hold out. This method is also known as ”permutation importance” (Breiman, 2001) or Ablation study. In this experiment, iteratively all the features are shuffled and the decrease of the F1 score is tracked.

test set	threshold for class non-accident	precision	recall	accuracy	f1-score
sample over three years	>50%	0.98	0.61	0.94	0.76
	>85%	0.82	0.81	0.95	0.82
year 2017	>50%	0.90	0.63	0.94	0.74
	>85%	0.62	0.82	0.90	0.71

Table 7.3: evaluation metrics

The results can be viewed in Table 7.4. One can see that, against the gain evaluation metric, road specific features like the centrality, curvature and PageRank are the most important ones. For these attributes the F1 score decreases the most when permuted. Against that, features like lighting conditions and the speed does not really to have an impact at all.
One issue which arises when using the permutation importance is when the data set contains multicollinear features. As shown in the Appendix 9.2, there are still no great correlations of input features. In Figure 7.20 and 7.21 some weak correlations are carried out. In the plots higher correlated features are brighter than lower correlated once.

permuted feature	F1 score
none	0.85
centrality	0.59
curvature	0.63
PageRank	0.64
slope	0.64
width	0.66
lanes_total	0.69
population density	0.71
district	0.78
precipitation	0.81
temperature	0.82

Table 7.4: permutation evaluation

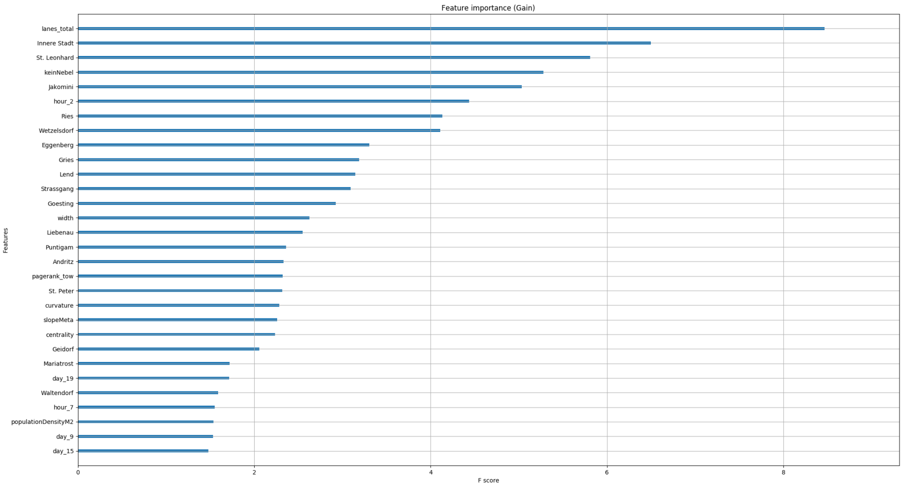


Figure 7.19: Feature importance ranking by XGBoost classifier

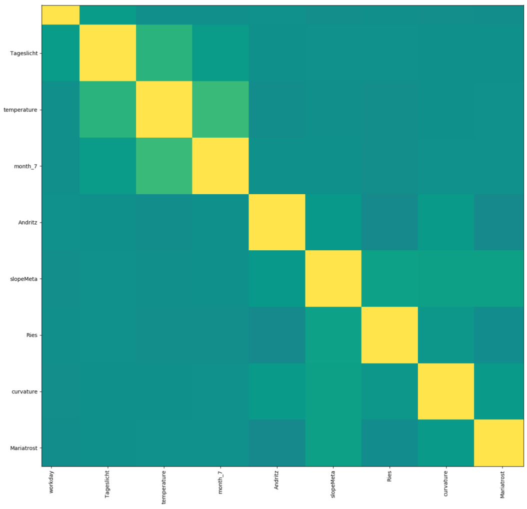


Figure 7.20: Correlation heatmap

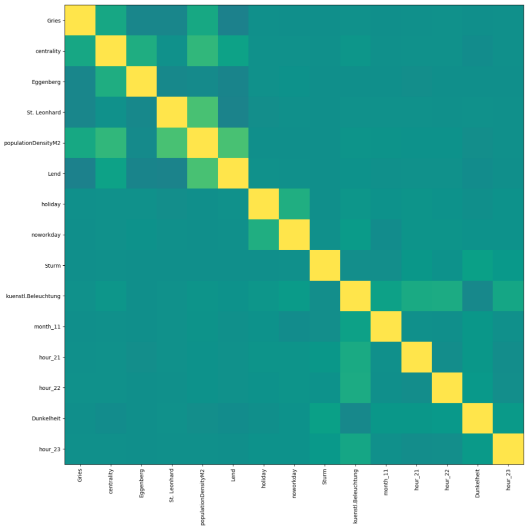


Figure 7.21: Correlation heatmap

7.5.2 Including traffic flow measurements

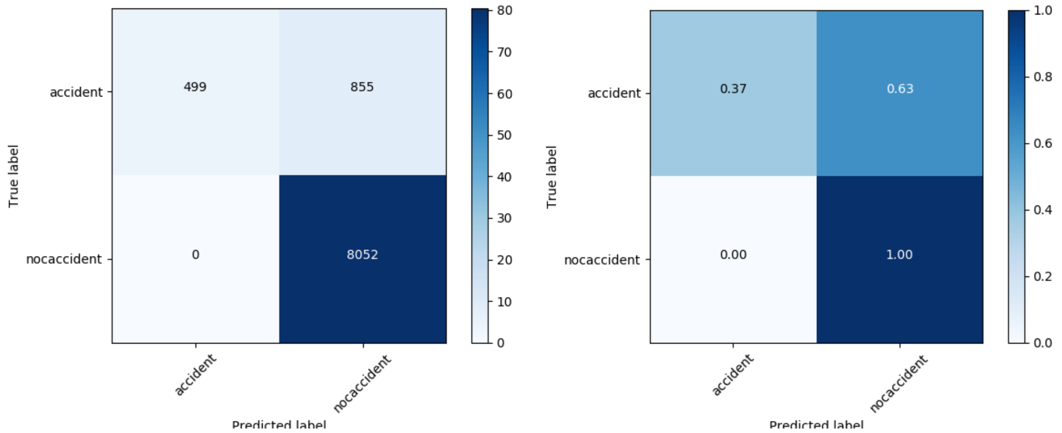
In this section, the crash prediction is performed on the whole input data set, also including the traffic flow measurements. The estimated model hyperparameters, evaluated via grid search, can be viewed in Table 7.5. As evaluation metric the area under the precision recall curve (aucpr) is chosen.

hyper-parameter	**estimated value**
objective	binary:logistic
learning_rate	0.05
max_depth	20
min_child_weight	2
subsample	1
colsample_bytree	0.7
n_estimators	400
eval_metric	aucpr

Table 7.5: Hyper-parameters for the XGBoost classifier

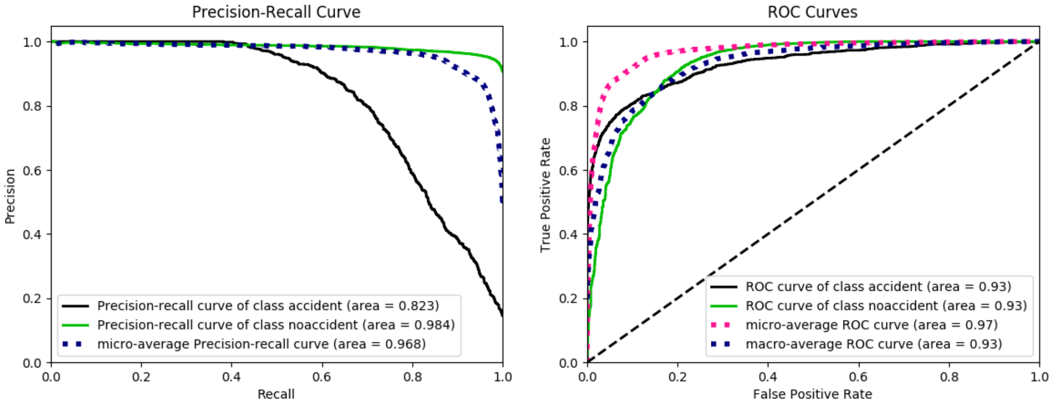
Testing the classifier with the chosen model parameters again with the retained test set, the following accuracy measurements as stated in Table 7.6 are estimated. In comparison with the results in Chapter 7.5.1, one can see that the recall decreased when using traffic flow data as pointwise explanatory variables. Also by having a look on the confusion metrics in Figure 7.22 and 7.24 it is obvious that the the TPR rate, also when choosing a classification threshold for non-accidents of 94% to get a relative close result in terms of the FNR compared to 7.5.1, decreased. This also can be viewed in the precision-recall plot in Figure 7.23, where the area under the curve also decreased.
So at all, adding the traffic flow values as pointwise features to the input data set does not seem to have a positive impact on the output results in general.

Also for this experiment, the feature importance for the most important values can be viewed in Figure 7.25. Closely the same features show a high



(a) (b)

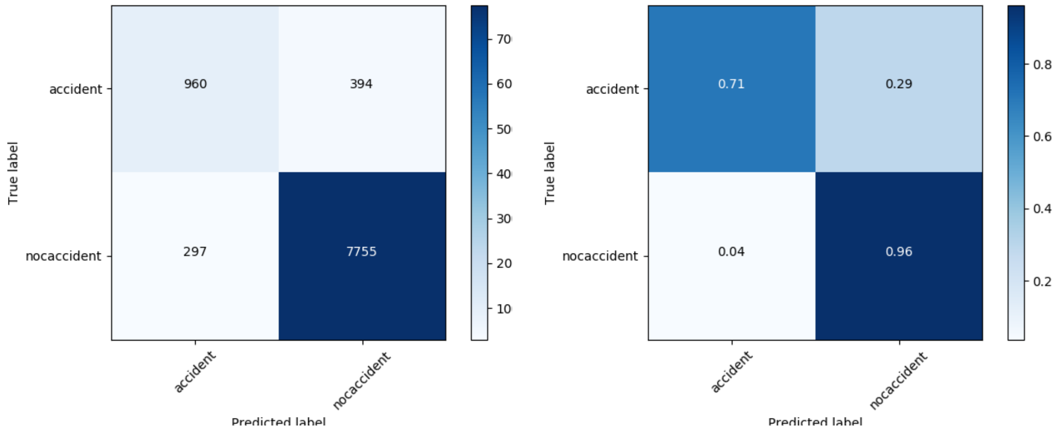
Figure 7.22: (a) confusion matrix absolute values (b) confusion matrix normalized



(a) (b)

Figure 7.23: (a) precision-recall (b) ROC

importance rate for the output. One can also see that no TFS has an great impact on the model.



(a) (b)

Figure 7.24: (a) confusion matrix absolute values (b) confusion matrix normalized

test set	prediction threshold for class non-accident	precision	recall	accuracy	f1-score
sample over three years	>50%	1.0	0.39	0.91	0.53
	>94%	0.76	0.71	0.93	0.74
year 2017	>50%	0.91	0.41	0.91	0.57
	>94%	0.36	0.84	0.76	0.51

Table 7.6: evaluation metrics

7.5.3 Discussion

The previous experiments show that the crash likelihood estimation score, based on different evaluation metrics, can be compared to other state of the art studies. Road link related features and the different districts of Graz seem to have the most impact on the overall result. Whereas taking pointwise measurement stations 15min and 30min prior accidents into account does not have any positive effect. One obvious reason for this might be that it is not essential for the output to include all measurement stations into the set of explanatory variables. It would be preferable to include only the closest n- stations for the given road link where the accident occurred. This will rise

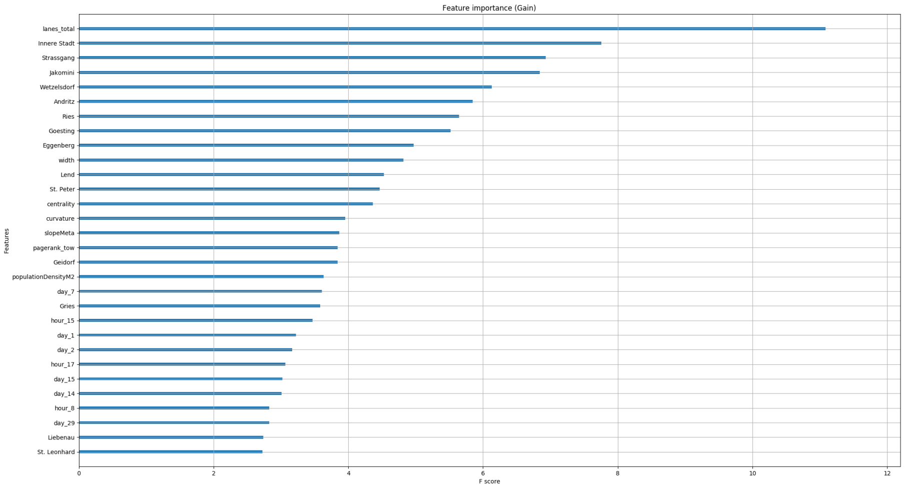


Figure 7.25: Feature importance ranking by XGBoost classifier

another problem, because not all accidents happen on street links closely to traffic flow measurement stations. So the model again will be biased through this effect. To overcome this problems, measurement data for all road links in the network graph of Graz would be necessary. This might be achieved by applying a traffic (forecasting) model based on the given traffic flow data. Another possibility would be to take other data sources under consideration where e.g. GPS data of the vehicles are tracked.

As stated in Chapter 2.2, the output of the model also strongly depends on the sampling strategy of negative samples. The non-accident events are sampled via the matched case strategy which can be said to be one of the state of the art sampling approaches. Another approach would be to sample similar roads only from a set of street links where accidents happened. Therefore a more accurate traffic flow data set which provides measurement data for all of these street links is required. One can argue that traffic flow data might has an much higher impact for the likelihood estimation when all the other explanatory variables vary not that much between accident and non-accident samples.

8 Conclusions

This thesis attempts the approach to estimate the crash likelihood in the city of Graz. Therefore different data sources like weather related measurements, population data, but also the street network graph with road link specific data and traffic flow measurements were matched with the crash data set.
An additional problem statement arised with the high missing value rate in the traffic flow data set. The MICE technique was used to overcome this multivariate imputation problem. It can be shown that the imputation quality of missing values strongly depends on the time series pattern of the stations. Values for stations with a clear daily pattern could be better predicted than stations where no pattern occurred at all. Also the correlation between measurement stations had a great impact on the estimation process itself.
Regarding the real-time estimation of the crash likelihood a Gradient Boosting approach was used. Before using this classification model, negative samples were generated with the matched case strategy. The output of the classification model strongly depended on the sampling strategy for negative samples.
It was observed that using the measurements of all stations 15min and 30min prior an accident as additional input variables, did not have an positive impact on the evaluation metrics for the classification model. Therefore an alternative strategy like using only the n-closest measurement stations would be preferable. For this approach a traffic flow model for the street network of Graz has to be applied or alternative data sources, tracking these traffic flow values per street link, have to be considered.
The results show that by including spatial, weather and crash data the precision and recall were around 0.82 and 0.81 respectively.

8.1 Future Work

This thesis assumes a clean crash data set, but as already mentioned in Chapter 2.2.1, crashes with car damage only do not have to be reported by the police. This is a great drawback in terms of a reliable crash forward prediction. Therefore other, more accurate, sources will be preferable for future work. Another interesting attempt might be, as tackled in Chapter 7.1.1, to figure out crashes based on the change in traffic flow measurements. In this work, the injury severity was completely left out in the prediction process. It might be interesting how the accuracy of the model changes when introducing some kind of weighting samples based on the severity of a certain accident.

With the rise of Internet of Things (IoT), also in the automotive industry, cars become the opportunity for data exchange in real-time. Cars will be able to provide and share detailed driving data like acceleration, speed, steering information but also driver related responses. This availability of real-time car and driver specific data will also have an great impact in the research field of road safety analysis and crash likelihood estimation. For the first time, personal driving data will be accessible and useable as an alternative explanatory data set in real-time. With this opportunity, many unobserved factors in terms of crash prediction will be overcome and the accuracy of the forecasting models will improve. This additional data might also serve as an opportunity to tackle the issue of near-accidents.

9 Appendix

9.1 Traffic flow features for classification model

number	traffic flow 15min prior	number	traffic flow 30min prior
1	t1_104.41A	197	t2_104.41A
2	t1_104.42A	198	t2_104.42A
3	t1_104.43A	199	t2_104.43A
4	t1_106.41A	200	t2_106.41A
5	t1_106.42A	201	t2_106.42A
6	t1_106.43A	202	t2_106.43A
7	t1_106.44A	203	t2_106.44A
8	t1_106.81A	204	t2_106.81A
9	t1_106.82A	205	t2_106.82A
10	t1_107.81A	206	t2_107.81A
11	t1_108.21A	207	t2_108.21A
12	t1_1123.21	208	t2_1123.21
13	t1_1123.22A	209	t2_1123.22A
14	t1_1123.23A	210	t2_1123.23A
15	t1_205.11	211	t2_205.11
16	t1_205.12	212	t2_205.12
17	t1_205.21A	213	t2_205.21A
18	t1_205.22A	214	t2_205.22A
19	t1_205.23	215	t2_205.23
20	t1_205.24	216	t2_205.24
21	t1_301.35	217	t2_301.35
22	t1_301.36	218	t2_301.36
23	t1_302.11	219	t2_302.11
24	t1_302.12	220	t2_302.12

25	t1_306.11	221	t2_306.11
26	t1_306.11A	222	t2_306.11A
27	t1_306.12	223	t2_306.12
28	t1_306.12A	224	t2_306.12A
29	t1_357.21	225	t2_357.21
30	t1_357.22A	226	t2_357.22A
31	t1_358.31	227	t2_358.31
32	t1_358.32	228	t2_358.32
33	t1_358.35A	229	t2_358.35A
34	t1_358.36A	230	t2_358.36A
35	t1_359.11	231	t2_359.11
36	t1_359.11A	232	t2_359.11A
37	t1_363.21A	233	t2_363.21A
38	t1_363.22	234	t2_363.22
39	t1_363.22A	235	t2_363.22A
40	t1_363.23	236	t2_363.23
41	t1_365.14	237	t2_365.14
42	t1_365.15	238	t2_365.15
43	t1_365.16	239	t2_365.16
44	t1_365.20	240	t2_365.20
45	t1_365.25	241	t2_365.25
46	t1_365.26	242	t2_365.26
47	t1_402.11	243	t2_402.11
48	t1_402.12	244	t2_402.12
49	t1_402.13	245	t2_402.13
50	t1_402.14A	246	t2_402.14A
51	t1_402.15A	247	t2_402.15A
52	t1_403.31	248	t2_403.31
53	t1_403.32	249	t2_403.32
54	t1_404.11	250	t2_404.11
55	t1_404.12	251	t2_404.12
56	t1_404.13	252	t2_404.13
57	t1_404.31	253	t2_404.31
58	t1_404.32	254	t2_404.32
59	t1_406.11	255	t2_406.11

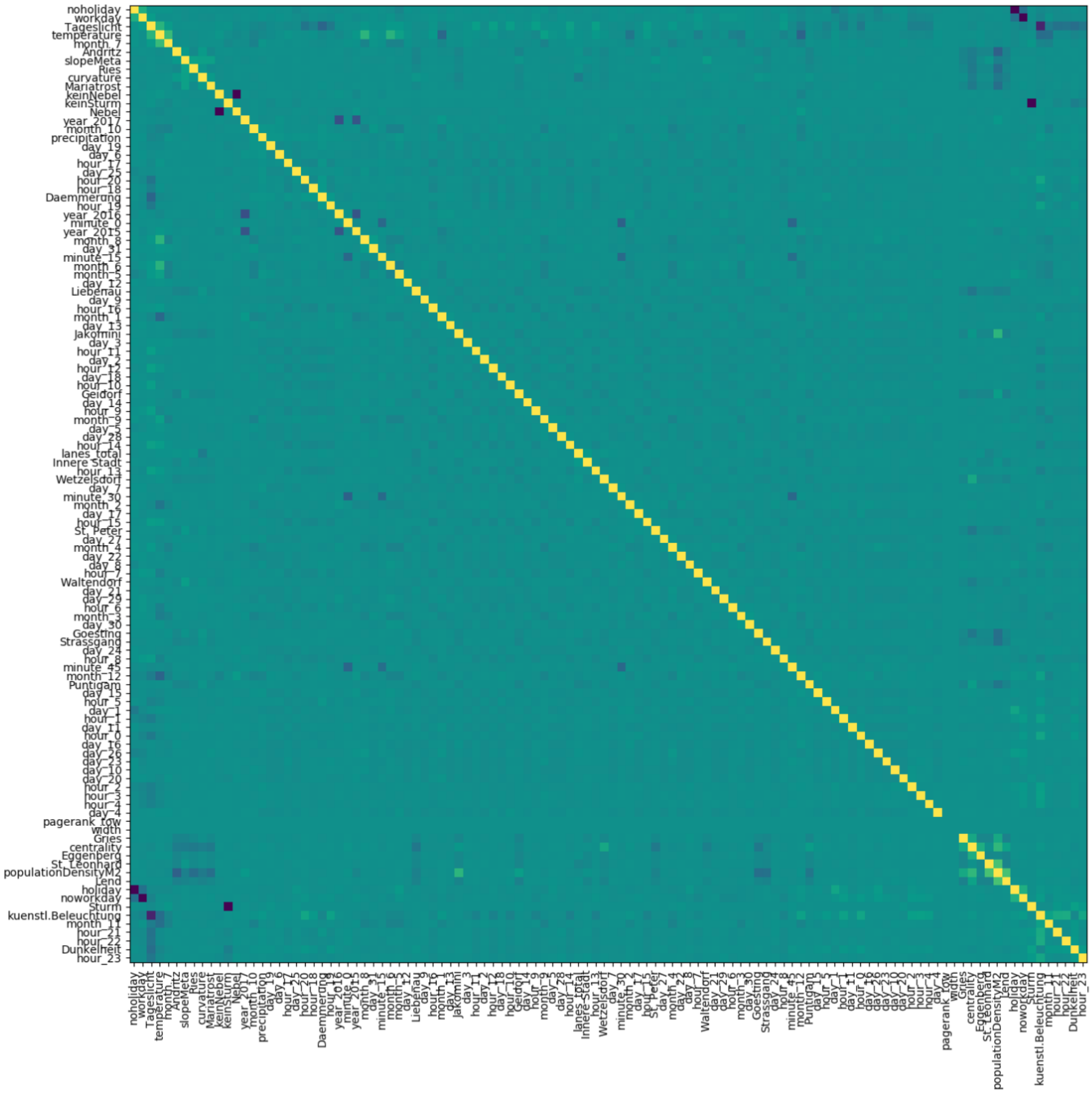
60	t1_406.12	256	t2_406.12
61	t1_406.13	257	t2_406.13
62	t1_406.14	258	t2_406.14
63	t1_406.15A	259	t2_406.15A
64	t1_406.16A	260	t2_406.16A
65	t1_406.21	261	t2_406.21
66	t1_406.22	262	t2_406.22
67	t1_406.31	263	t2_406.31
68	t1_406.32	264	t2_406.32
69	t1_408.31	265	t2_408.31
70	t1_408.32	266	t2_408.32
71	t1_408.33A	267	t2_408.33A
72	t1_408.34A	268	t2_408.34A
73	t1_408.35A	269	t2_408.35A
74	t1_419.22	270	t2_419.22
75	t1_419.23	271	t2_419.23
76	t1_419.24	272	t2_419.24
77	t1_419.25	273	t2_419.25
78	t1_419.42	274	t2_419.42
79	t1_419.43	275	t2_419.43
80	t1_419.44	276	t2_419.44
81	t1_423.32	277	t2_423.32
82	t1_423.33	278	t2_423.33
83	t1_423.34A	279	t2_423.34A
84	t1_423.35A	280	t2_423.35A
85	t1_423.41	281	t2_423.41
86	t1_423.42	282	t2_423.42
87	t1_423.43A	283	t2_423.43A
88	t1_425.11	284	t2_425.11
89	t1_425.12	285	t2_425.12
90	t1_425.13	286	t2_425.13
91	t1_425.14A	287	t2_425.14A
92	t1_425.15A	288	t2_425.15A
93	t1_501.71	289	t2_501.71
94	t1_501.72	290	t2_501.72

95	t1_501.74	291	t2_501.74
96	t1_501.75	292	t2_501.75
97	t1_501.81	293	t2_501.81
98	t1_501.82	294	t2_501.82
99	t1_501.83AD	295	t2_501.83AD
100	t1_501.84AD	296	t2_501.84AD
101	t1_504.21	297	t2_504.21
102	t1_504.22	298	t2_504.22
103	t1_504.23	299	t2_504.23
104	t1_504.24A	300	t2_504.24A
105	t1_504.31	301	t2_504.31
106	t1_504.32	302	t2_504.32
107	t1_504.51	303	t2_504.51
108	t1_504.52	304	t2_504.52
109	t1_504.53	305	t2_504.53
110	t1_505.11	306	t2_505.11
111	t1_505.31A	307	t2_505.31A
112	t1_505.32A	308	t2_505.32A
113	t1_508.31A	309	t2_508.31A
114	t1_508.32A	310	t2_508.32A
115	t1_509.31	311	t2_509.31
116	t1_509.32A	312	t2_509.32A
117	t1_514.21A	313	t2_514.21A
118	t1_514.41A	314	t2_514.41A
119	t1_514.42A	315	t2_514.42A
120	t1_515.22	316	t2_515.22
121	t1_518.11	317	t2_518.11
122	t1_518.12	318	t2_518.12
123	t1_518.13A	319	t2_518.13A
124	t1_554.11	320	t2_554.11
125	t1_554.12	321	t2_554.12
126	t1_554.13A	322	t2_554.13A
127	t1_554.14A	323	t2_554.14A
128	t1_555.31	324	t2_555.31
129	t1_555.32	325	t2_555.32

130	t1_556.31A	326	t2_556.31A
131	t1_601.21A	327	t2_601.21A
132	t1_601.22A	328	t2_601.22A
133	t1_602.21A	329	t2_602.21A
134	t1_602.22A	330	t2_602.22A
135	t1_602.23A	331	t2_602.23A
136	t1_606.21A	332	t2_606.21A
137	t1_606.22A	333	t2_606.22A
138	t1_707.41	334	t2_707.41
139	t1_707.42	335	t2_707.42
140	t1_707.43A	336	t2_707.43A
141	t1_707.44A	337	t2_707.44A
142	t1_708.31	338	t2_708.31
143	t1_708.32	339	t2_708.32
144	t1_708.33	340	t2_708.33
145	t1_708.36A	341	t2_708.36A
146	t1_708.37A	342	t2_708.37A
147	t1_712.21	343	t2_712.21
148	t1_712.22	344	t2_712.22
149	t1_712.23	345	t2_712.23
150	t1_712.24A	346	t2_712.24A
151	t1_715.31	347	t2_715.31
152	t1_715.32A	348	t2_715.32A
153	t1_719.21D	349	t2_719.21D
154	t1_719.22D	350	t2_719.22D
155	t1_719.23	351	t2_719.23
156	t1_720.41D	352	t2_720.41D
157	t1_720.42D	353	t2_720.42D
158	t1_748.21	354	t2_748.21
159	t1_748.22	355	t2_748.22
160	t1_748.33	356	t2_748.33
161	t1_748.34	357	t2_748.34
162	t1_748.41	358	t2_748.41
163	t1_748.42	359	t2_748.42
164	t1_748.43	360	t2_748.43

165	t1_751.11	361	t2_751.11
166	t1_751.12	362	t2_751.12
167	t1_751.13A	363	t2_751.13A
168	t1_753.31	364	t2_753.31
169	t1_753.31A	365	t2_753.31A
170	t1_753.41	366	t2_753.41
171	t1_753.41A	367	t2_753.41A
172	t1_801.31	368	t2_801.31
173	t1_801.32	369	t2_801.32
174	t1_801.33A	370	t2_801.33A
175	t1_801.34A	371	t2_801.34A
176	t1_801.41	372	t2_801.41
177	t1_801.42A	373	t2_801.42A
178	t1_805.11	374	t2_805.11
179	t1_805.12A	375	t2_805.12A
180	t1_805.13A	376	t2_805.13A
181	t1_806.11	377	t2_806.11
182	t1_806.12A	378	t2_806.12A
183	t1_806.13A	379	t2_806.13A
184	t1_818.21	380	t2_818.21
185	t1_818.22A	381	t2_818.22A
186	t1_822.11	382	t2_822.11
187	t1_822.12	383	t2_822.12
188	t1_822.13A	384	t2_822.13A
189	t1_822.14A	385	t2_822.14A
190	t1_824.31	386	t2_824.31
191	t1_824.34	387	t2_824.34
192	t1_824.35A	388	t2_824.35A
193	t1_824.36A	389	t2_824.36A
194	t1_905.21	390	t2_905.21
195	t1_905.31	391	t2_905.31
196	t1_905.32	392	t2_905.32

9.2 Correlation heatmap



Bibliography

Ahmed, Mohamed M and Mohamed Abdel-Aty (2013). "Application of stochastic gradient boosting technique to enhance reliability of real-time risk assessment: use of automatic vehicle identification and remote traffic microwave sensor data." In: *Transportation research record* 2386.1, pp. 26–34 (cit. on p. 101).

Anderson, Tessa K (2009). "Kernel density estimation and K-means clustering to profile road accident hotspots." In: *Accident Analysis & Prevention* 41.3, pp. 359–364 (cit. on p. 29).

Asif, Muhammad Tayyab et al. (2016). "Matrix and tensor based methods for missing data estimation in large traffic networks." In: *IEEE Transactions on intelligent transportation systems* 17.7, pp. 1816–1825 (cit. on p. 2).

Azur, Melissa J et al. (2011). "Multiple imputation by chained equations: what is it and how does it work?" In: *International journal of methods in psychiatric research* 20.1, pp. 40–49 (cit. on p. 82).

Bae, Bumjoon et al. (2018). "Missing data imputation for traffic flow speed using spatio-temporal cokriging." In: *Transportation Research Part C: Emerging Technologies* 88, pp. 124–139 (cit. on p. 30).

Boeing, Geoff (2017). "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks." In: *Computers, Environment and Urban Systems* 65, pp. 126–139 (cit. on p. 37).

Breiman, Leo (2001). "Random forests." In: *Machine learning* 45.1, pp. 5–32 (cit. on p. 107).

Buuren, S van and Karin Groothuis-Oudshoorn (2010). "mice: Multivariate imputation by chained equations in R." In: *Journal of statistical software*, pp. 1–68 (cit. on p. 82).

Chen, Quanjun et al. (2016). "Learning deep representation from big and heterogeneous data for traffic accident inference." In: *Thirtieth AAAI Conference on Artificial Intelligence* (cit. on pp. 28, 29).

Chen, Tianqi and Carlos Guestrin (2016). "Xgboost: A scalable tree boosting system." In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, pp. 785–794 (cit. on pp. 34, 104).

Chong, Miao, Ajith Abraham, and Marcin Paprzycki (2005). "Traffic accident analysis using machine learning paradigms." In: *Informatica* 29.1 (cit. on p. 28).

Friedman, Jerome H (2001). "Greedy function approximation: a gradient boosting machine." In: *Annals of statistics*, pp. 1189–1232 (cit. on pp. 34, 104).

Géron, Aurélien (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O'Reilly Media. ISBN: 978-1491962299 (cit. on p. 33).

Gross, Jonathan L and Jay Yellen (2005). *Graph theory and its applications*. CRC press (cit. on p. 35).

Hagberg, Aric A., Daniel A. Schult, and Pieter J. Swart (2008). "Exploring Network Structure, Dynamics, and Function using NetworkX." In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, pp. 11–15 (cit. on p. 37).

Haworth, James and Tao Cheng (2012). "Non-parametric regression for space–time forecasting under missing data." In: *Computers, Environment and Urban Systems* 36.6, pp. 538–550 (cit. on p. 30).

Henrickson, Kristian, Yajie Zou, and Yinhai Wang (2015). "Flexible and robust method for missing loop detector data imputation." In: *Transportation Research Record* 2527.1, pp. 29–36 (cit. on pp. 30, 31).

Howard, Mark E et al. (2004). "Sleepiness, sleep-disordered breathing, and accident risk factors in commercial vehicle drivers." In: *American journal of respiratory and critical care medicine* 170.9, pp. 1014–1021 (cit. on p. 1).

Ke, Jintao et al. (2019). "PCA-based missing information imputation for real-time crash likelihood prediction under imbalanced data." In: *Transportmetrica A: transport science* 15.2, pp. 872–895 (cit. on pp. 29, 30, 101, 104).

Kruschke, John K, Herman Aguinis, and Harry Joo (2012). "The time has come: Bayesian methods for data analysis in the organizational sciences." In: *Organizational Research Methods* 15.4, pp. 722–752 (cit. on pp. 83, 84).

Langville, Amy N and Carl D Meyer (2005). "A survey of eigenvector methods for web information retrieval." In: *SIAM review* 47.1, pp. 135–161 (cit. on p. 36).

Little, Roderick JA and Donald B Rubin (2019). *Statistical analysis with missing data*. Vol. 793. John Wiley & Sons (cit. on p. 30).

Lord, Dominique and Fred Mannering (2010). "The statistical analysis of crash-frequency data: a review and assessment of methodological alternatives." In: *Transportation research part A: policy and practice* 44.5, pp. 291–305 (cit. on pp. 1, 26).

Mair, Alan and Ali Fares (2010). "Comparison of rainfall interpolation methods in a mountainous region of a tropical island." In: *Journal of Hydrologic Engineering* 16.4, pp. 371–383 (cit. on p. 70).

Mannering, Fred L and Chandra R Bhat (2014). "Analytic methods in accident research: Methodological frontier and future directions." In: *Analytic methods in accident research* 1, pp. 1–22 (cit. on pp. 1, 27).

Noori, Mohamad J, Hussein H Hassan, and Yaseen T Mustafa (2014). "Spatial estimation of rainfall distribution and its classification in Duhok governorate using GIS." In: *Journal of Water Resource and Protection* 6.02, p. 75 (cit. on p. 69).

Page, Lawrence et al. (1999). *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab (cit. on p. 36).

Qu, Li et al. (2009). "PPCA-based missing data imputation for traffic flow volume: A systematical approach." In: *IEEE Transactions on intelligent transportation systems* 10.3, pp. 512–522 (cit. on pp. 2, 30).

Ren, Honglei et al. (2018). "A deep learning approach to the citywide traffic accident risk prediction." In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 3346–3351 (cit. on pp. 28, 29).

Rubin, Donald B (2004). *Multiple imputation for nonresponse in surveys*. Vol. 81. John Wiley & Sons (cit. on p. 31).

Shang, Qiang et al. (2018). "An imputation method for missing traffic data based on FCM optimized by PSO-SVR." In: *Journal of Advanced Transportation* 2018 (cit. on p. 30).

Shepard, Donald (1968). "A two-dimensional interpolation function for irregularly-spaced data." In: *Proceedings of the 1968 23rd ACM national conference.* ACM, pp. 517–524 (cit. on p. 37).

Soley-Bori, Marina (2013). "Dealing with missing data: Key assumptions and methods for applied analysis." In: *Boston University* 23 (cit. on p. 78).

Sun, Jie and Jian Sun (2015). "A dynamic Bayesian network model for real-time crash prediction using traffic speed conditions data." In: *Transportation Research Part C: Emerging Technologies* 54, pp. 176–186 (cit. on p. 28).

Tan, Huachun et al. (2013). "A tensor-based method for missing traffic data completion." In: *Transportation Research Part C: Emerging Technologies* 28, pp. 15–27 (cit. on p. 30).

Yuan, Zhuoning et al. (2017). "Predicting traffic accidents through heterogeneous urban data: A case study." In: *6th International Workshop on Urban Computing (UrbComp 2017)* (cit. on pp. 28, 29, 104).