Predicting the risk level of zones within cities to avoid accidents

- Louis Lefebvre, Mathieu Pieronne, Venkatesh Subramani, Julian Kopp -

Business Understanding

Insurance companies are interested in reducing the amount of accidents as this decreases cost. We offer them a solution which warns a driver before entering a 'dangerous zone'. The driver increases their attention and therefore, decreases the amount of accidents. The danger-level of a zone is determined by our model. We base this model on police report data which provides several environmental variables.

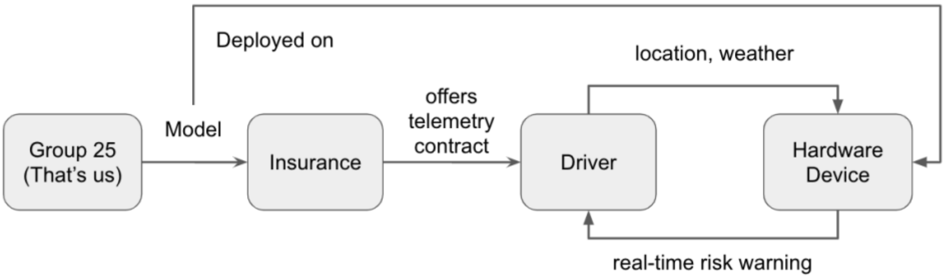


Figure 1: Overview of business model

In order that the driver can be warned a small hardware device is integrated into the car as part of a telemetry contract between the driver and the insurance company. The hardware device contains our trained model and constantly feeds the real-time position and respective weather conditions of the car. The model outputs if the zone ahead is dangerous or not. This telemetry contract poses two advantages for a driver. First, as accident prone zones are announced the driver can prepare and avoid accidents. Second, the insurance can offer this telemetry contract at a lower fee as decreased quantity of accidents mean reduced cost as well. The cost saving potential is shown in Appendix A.

Legal Understanding

We see three main sources of data. First, being police reports which are open-source for the prototype. Only requirement by the city of Chicago which provisions the dataset is a disclaimer of restricted liability. The second type of data is weather data which is obtained by a third party service.

So far, there is no personal data involved. This changes with the geolocation of a driver. This information is necessary for our business model as the driver gets warnings based on her trajectory. To comply with the GDPR ruleset we anonymise the message containing the geolocation from a car. On top, this anonymised ID changes for every session or drive. The combination ensures that tracing a driver is not possible. Additionally, the entries are not stored longer than 24 hours.

Lastly, it is important to separate our model, service and included information about the driver from the contract information of the insurance. Trajectories which often end up in one place could be mapped to the address of a client. A could be identified and ultimately her privacy is violated.

Data Understanding

The dataset consists of 450 thousand samples of Chicago police accident reports from 2015-2020. It includes extensive information about the external conditions and the context of an accident. The dataset is provided by the city of Chicago on their official data website and continually updated. Thus, data-quality is not an issue.

Own Creation	CRASH_HOUR, CRASH_DAY_OF_WEEK, CRASH_MONTH
kept	CRASH_DATE, WEATHER_CONDITION, LIGHTING_CONDITION, ROADWAY_SURFACE_COND, Latitude, Longitude

Table 1: Reasoning of removal of variables and overview which variables remain.

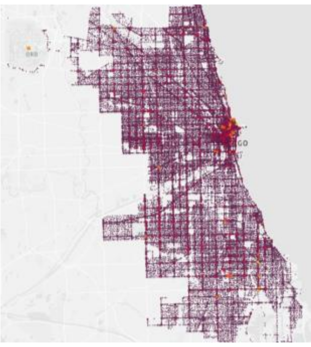


Figure 1: Dataset visualised on Chicago administrative area (using kepler.gl)

As a first step we reviewed different features the dataset offers and removed non-useful features as well as useful features with non-usable data. Table 1 gives an overview of the remaining features. We limit us to this rather narrow selection of features as these can be generated in real time by the hardware device in the car. The new sample is then compared to the historic model.

We determine a large imbalance in labels in the columns WEATHER_CONDITION, LIGHTING_CONDITION, ROADWAY_SURFACE_COND. In each column the ideal state (weather: clear, lighting: daylight, road surface: dry) is pre-dominant. As these are all samples of accidents, it leads to the conclusion that much of the variation in accidents and therefore much of the causality is not caused by difficult external circumstances. We will have to test if enough variance is explained by our external features or if the human behaviour has a too large behaviour.

Figure 2 also allows some sanity checks. For example, there are less accidents during the night and the dusk and dawn label for the lighting condition appears at appropriate hours. We can also clearly see the level of traffic as in the morning between 7-9am people go (rather went) to the office and return between 15-17pm. In addition to temporal variation we can also see variation based on location in figure 3.

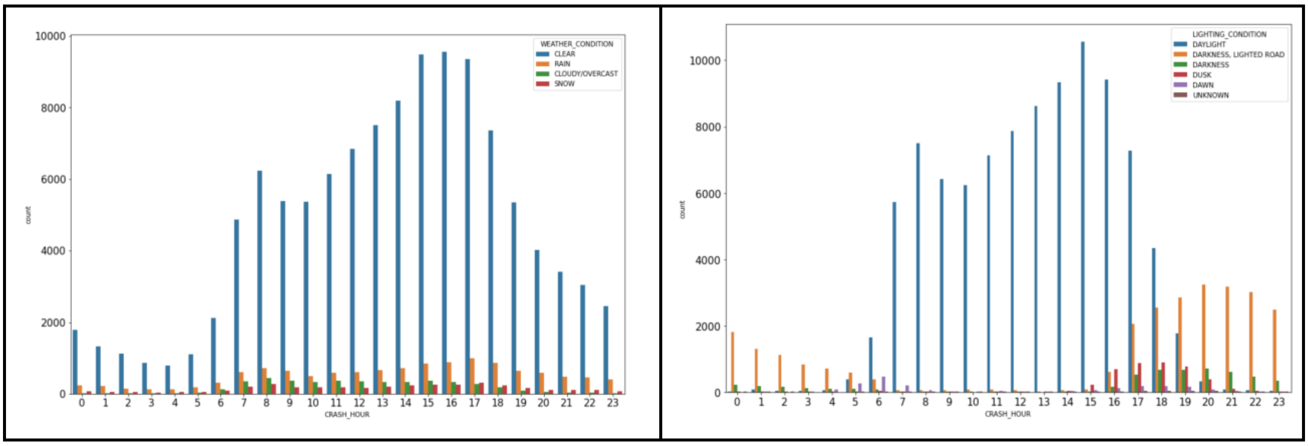


Figure 2: Label distribution of weather condition (left) and lighting condition (right) across daily hours

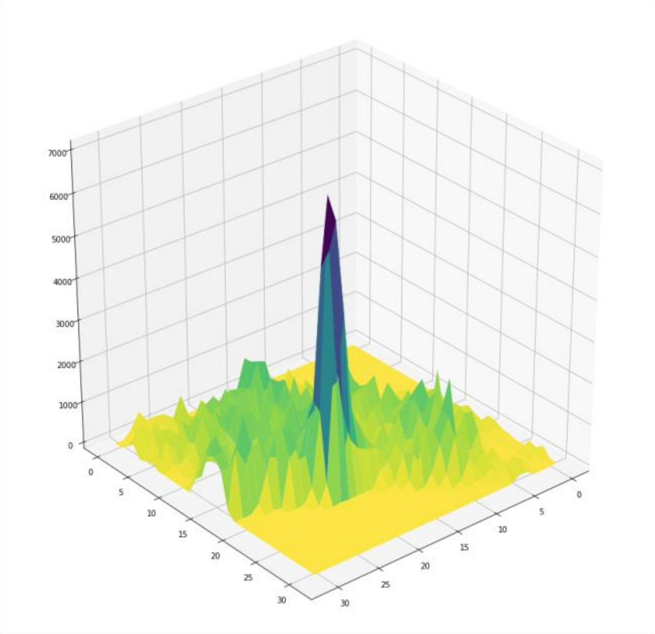
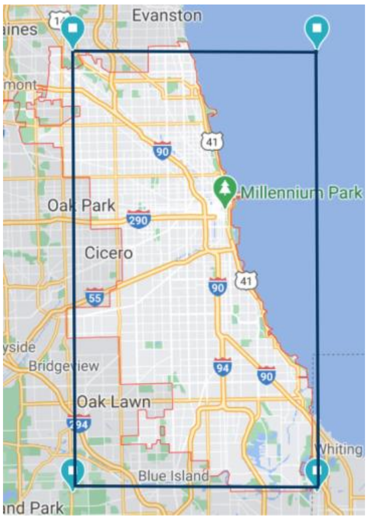
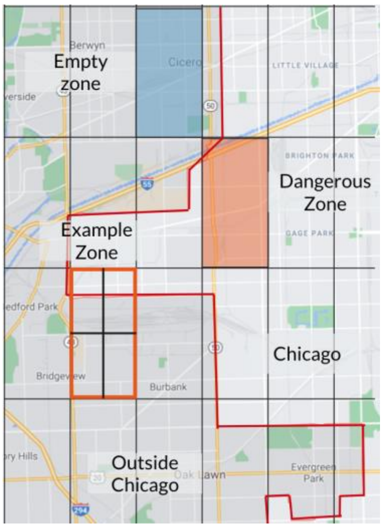


Figure 3: Data already partitioned in zones. There are clear differences across zones.

Data preparation I

What is a zone?

In the Business Understanding section we talk about creating a score for a zone. But what really is a zone? We define a zone as a rectangular area of small size within the city of Chicago. We realise this concept by spanning a grid between four confining coordinates with latitude and longitude (Figure 2). This rectangle is divided into $x * x$ cells where x specifies the amount of zone along both axes. Note, that the created cells do not have a quadratic, absolute length as the bounding box used here is rectangular. A zone can then be considered dangerous or not. We also get empty zones as they are part of the bounding box but not part of the administrative area.

 Evanston Oak Park Cicero Millennium Park Bridgeview Oak Lawn Blue Island Whiting	 Empty zone Berwyn Cicero Little Village Brighton Park Dangerous Zone Gage Park Example Zone Chicago Bridgeview Burbank Outside Chicago Evergreen Park	*Figure 2: Left - Bounding box of the grid over Chicago. Note, a more precise and rotated bounding box caused technical difficulties and is not essential in building a PoC model version.* *Figure 3: Right - Zooming into the grid. Right of the red line is the city area of Chicago.*

How large can x be without data sparsity problems?

We create several grids with different x based on the resulting km^2 per zone. We then analyse the mean, standard deviation and median to detect potential data sparsity. In addition we plot the distribution to get better intuition and also do scatter plots for each grid version.

Grid configuration	33x33	48x48	67x67	96x96
km^2 per zone	1.0	0.5	0.25	0.12
South-North length in **m**	1255	863	618	431
E-W length in **m**	819	563	403	282
Total Zones	1119	2284	4476	9135

Table 2: Grid versions and the dimensions of zone size and quantity

Empty zones are calculated based on how many zones there should theoretically be, based on x. Because of the large bounding box, many zones are empty. The increase of empty zones is marginal when increasing the resolution. Increasing the number of zones means a single cell will get smaller and could then potentially be rendered outside the city bounds (see Figure 3, "Example Zone"). In all versions a considerable number of zones have only few accidents. From the scatterplots we can see that there are no (or only few) empty zones which have no accidents at all. Thus, data is not sparse and we can assume that the majority of cells have enough data for prediction from historical observations. Due to some extreme outliers we also check the median which is robust against outliers.

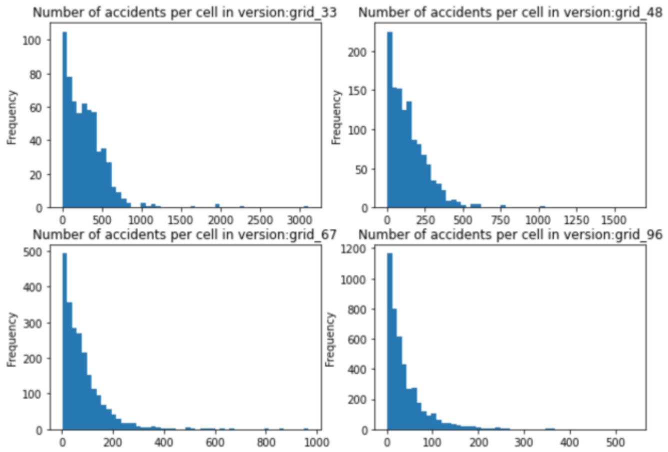
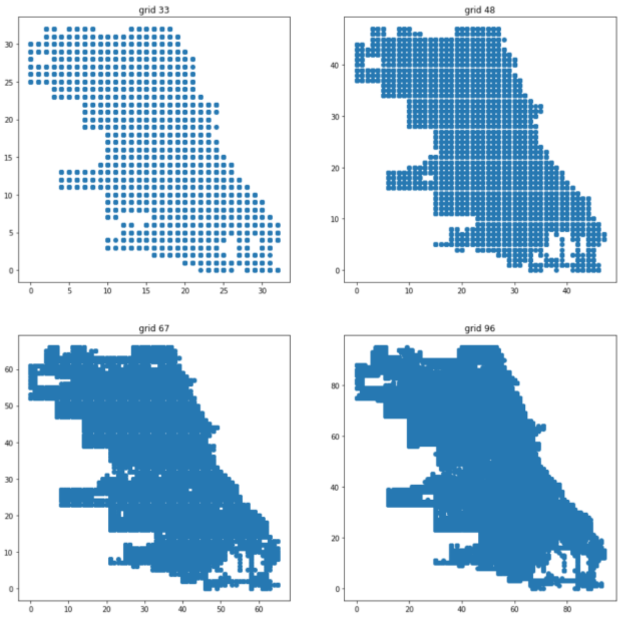


Figure 4: Top - Histograms of the amount of accidents per zone.

Figure 5: Right - Scatter plots of different grid versions. All resemble the administrative area of Chicago and show no 'holes' of data within this area.



We would like to choose the grid version with most zones and therefore highest granularity of a 'risk zone'. However, the granularity needs to be balanced with data sparsity. We decide to proceed with grid_67 as it has a median of over 80 accidents per cell which we judge good enough. grid_96 seems 'okayish' as well but we do not want to take an extreme first decision. If further analysis shows that the model could work with even less data we will come back and quickly rerun the process for grid_96.

	33x33	48x48	67x67	96x96
mean	295.55	149.23	80.11	41.50
std	278.97	144.88	83.98	49.31
median	257.00	121.00	59.00	27.00
% empty cells	45.04	46.67	49.31	52.05

Table 3: Overview of metrics for grid versions

Modelling I

Creating a score based on feature variables

Our first idea is to construct a formula which takes into account the different circumstances (excluding location) of an accident to create a danger score. To keep it short, results were not satisfactory. We experienced the same issues as a similar project: the score itself correlated with the variables it was based on but wasn't helpful in predicting accidents. We come to the conclusion that the location feature needs to play a more important role.

Data Modelling with OneClass SVM

Our dataset of police reports only includes samples of accidents. Thus we cannot use the usual classification algorithms choices hence we opt for OneClass SVM. This algorithm determines the main characteristics of the studied class and encircles these samples. All new samples which are not inside this circle belong to the opposite class. In our case this opposite class means 'not an accident'. Another option with this algorithm is to leverage the distance between the circle and a new sample to create a gradual scale (high, medium, low, no risk) instead of binary classification. Lastly, we can also include the location as a feature.

To keep it short as well, the results of one class SVM with different kernels are worse than random choice and therefore not usable. We suspect that the circle drawn by the one class SVM is very large as accidents happen in nearly all conditions.

Using our finding of severe label imbalance and a majority of accidents in ideal conditions (clear weather, dry road, daylight), the circle basically includes all samples with ideal conditions no matter if an accident or not.

Evaluation I

In light of the business understanding the so far examined approaches and algorithms are useless. While searching for alternatives we realise our severe limitation in having only one class of data. We decide to go for a second iteration of the CRISP-DM cycle beginning with several findings in the data understanding: Our original idea of a score was fruitless - we broaden the scope of our accident prediction to a score or a binary classification (accident occurs in a zone in certain conditions or not). For conventional classification algorithms we need non-accident data. Also, we realise that the imbalance of labels in features is more problematic than anticipated. We continue with a second phase of data preparation.

Data preparation II

Sampling Non-Accident Data

After initial analysis we had to drop many features of the original dataset due to various reasons. An alternative option to building a numeric score of dangerousness could be a binary classifier between dangerous and not-dangerous. However, all classic classification algorithms need two classes to be able to classify and our dataset only provides accident samples.

In order to generate realistic non-accident data we base the distributions of the features (weather, road condition, etc.) on the distributions of the augmented data set. A uniform random variable is drawn. Based on the percentage occurrence of a label in a feature the respective range of the random variable result space is allocated to this label. For example, dry makes up 70% of the weather column so if the random variable is between 0 and 0.7 it will be sampled as dry. This is done for all other columns except the location.

The location inside the grid poses a difficulty. We cannot assume that non-accident samples have the same underlying location distribution as accident-samples. That would deny the difference of dangerousness of two zones with identical amount of traffic. Therefore, we leverage data from the tomtom traffic stats service. The dataset gives the level of traffic per street which we map to zones and calculate the percentage of traffic per zone. Again, we partition the result space of a uniform random variable into ranges. Each range corresponds to the percentual amount of traffic of one zone.

Data Augmentation

Because of the previously described imbalance of labels in external factor features we suspect that any type of model could easily predict a high danger score for ideal conditions all the time. This would give the model a good performance but would also defy the usability and logic of the model. We therefore apply the renowned SMOTE algorithm (We use the NC version of the algorithm as it can deal with categorical data) on the dataset in order to reduce the ratio between the ideal condition labels (dry, clear, daylight). The idea is to generate additional accident data to help later models recognise these accidents in non-ideal conditions. We will compare the performance between the original dataset and the augmented dataset and further analyse the implications of this step in the evaluation section.

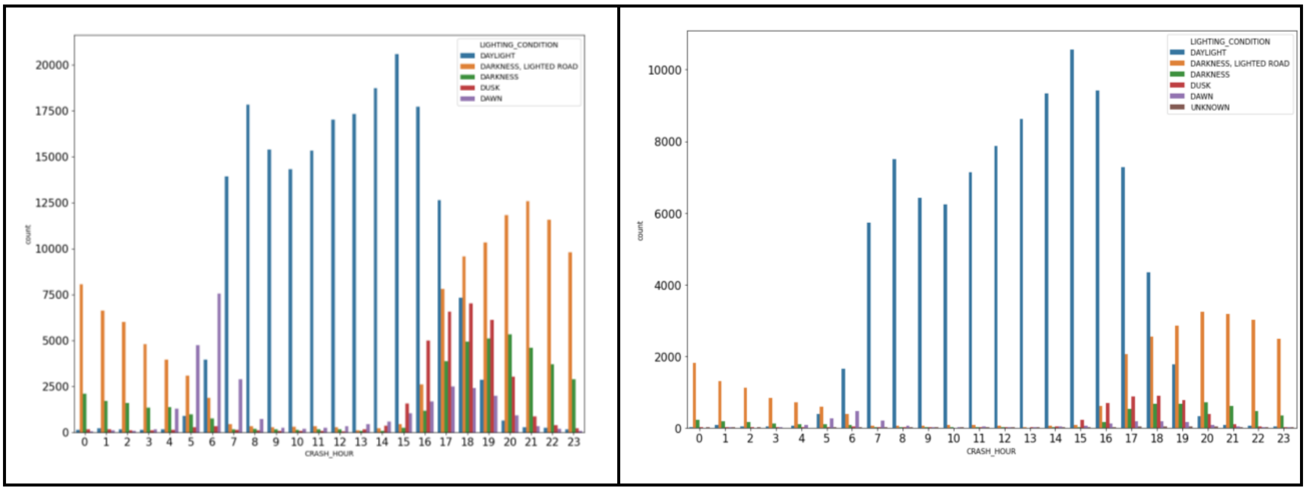


Figure 6: Comparison of lighting condition per hour. Left is augmented data, right is original data.

The comparison between SMOTE-NC augmented and original data shows the successful reduction of the imbalance between the majority and minority classes. Comparing the amount of accidents at 0 am with the peak of accidents at 14 pm the original ratio was 1:5 (~2000:~10000 accidents) while the new ratio is 1:2 (~7500 + ~2500 : ~200000). We also observe that other underlying distributions are kept as all labels show the same patterns as before. A weakness is the amount of dawn label occurrences in the evening time.

Modelling II

We again start with our initial idea of building a score manually. After some experimentation we realise that we do not have any features which give us information about the real dangerousness of external events. Especially the fact that we ourselves decide the ratio between accident and non-accident samples is problematic. In other words, our score doesn't have anything to rely on. We finally discard this approach.

For all following algorithms we use the augmented version of our dataset with the accident class as well as the non-accident class. Also, we split the data into a training (90%), validation (5%) and test set (5%) as we had a large set of samples (~800,000).

We start with the classic SVM to get some intuition on the needed complexity of our model. We apply a linear kernel and after 23 minutes of training we obtain an accuracy of 60.2% for the test set and 60.4% for the validation set. We do not observe any variance but a heavy bias in the model.

We increase the complexity of our model by using a multi-layer perceptron or neural network. Figure 7 shows its architecture. We spent considerable hours tuning hyperparameters to have the best version of the said-network. The network is trained for 50 epochs which takes around 20 minutes. It obtains an accuracy of 85.2% for both the test and validation set. Again we obtain very low variance. We attribute this to the simplistic architecture of the network. We also observe some bias but it is much better than with the previously tested SVM.

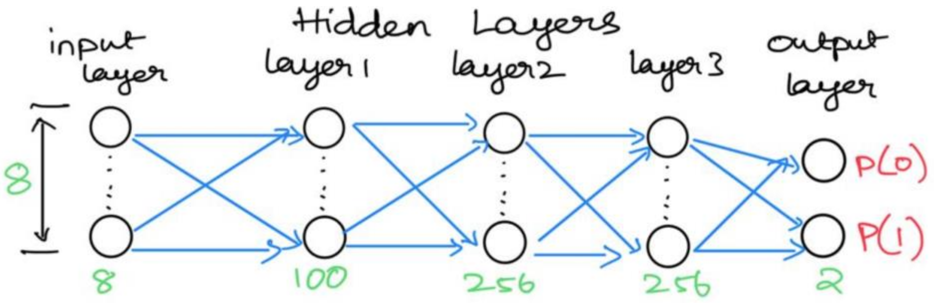
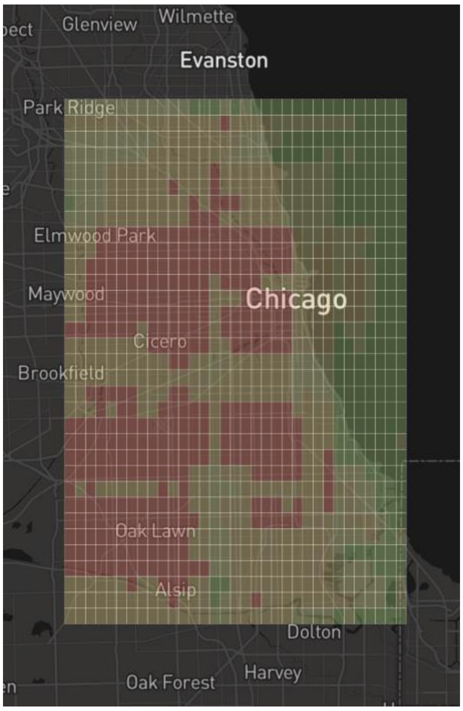


Figure 7: Architecture of the neural network.

Lastly, we apply a random forest classifier. After 6 minutes of training we obtain 94.3% accuracy on the test set and 94.6% accuracy on the validation set. With this model we obtain a performant bias with neglectable variance. With the output vector for class 0 and 1 (no accident, accident) of the random forest we get a level of confidence which can be interpreted as the danger-level of a zone. To visualise this danger-level we fix the external parameters and plot the grid over Chicago.



Evaluation II

Our second iteration of models was successful. We determine that a random forest model suits the data best. However, we would need more extensive testing with different metrics and datasets to fully validate the model. Also, the predictions are not fine tuned yet as can be seen in the previous visualisation. Large parts of Chicago are considered dangerous which in itself can be helpful but poses difficulties in user acceptance and lets us question the final impact on accident reduction.

We come to the conclusion that the approach of basing an zone danger-level prediction model on police reports is not sound enough for a real world business use case. We achieve preliminary results but do not see from which features fine tuning and robustness should come from. We can give a general tendency on the risk of a zone, however this is not sufficient for a market product.

Another issue is that a major part of variance in the cause of accidents is induced by human behaviour. With the gained experience, we recommend the decision maker to look for other data sources which explain a good portion of the variance of human behaviour. A feasibility study concerning economic and legal aspects should come before actual model development with such data sources.

Learnings

We reflect on our learnings throughout this project:

1) Expressive features are fundamental for every step down the (CRISP-DM) line. This gives enormous importance to the analysis of correlation of features with the target variable.
2) It is worth thinking about the natural system one wants to model with data science tools. Initially, we thought we would model accidents but later realised that a good portion of modeling accidents is modeling human behaviour. Several caveats with modeling human behaviour: A) it is very complex B) one needs different kinds of data and features C) these features can be difficult to obtain in terms of infrastructure needed (cameras, sensors, etc.) D) these features pose a difficulty (or at least considerably additional work) in legislative matters E) triggers problems related to veracity of the data itself.
3) It is always important to keep in mind the business use case. During development there is a certain risk that requirements are adapted to technical details and difficulties.
4) Communication is difficult. Small details which decide over a successful run or Error in coding and implicit assumptions which seem obvious to one but not to another can cause mischief.
5) The legal aspects of data usage can be show-stopper. Yet, they are not really present in main-stream online data science literature.

Next Steps

From a standpoint of curiosity (not economic), certain points and experiments would be interesting:

- comparing the performance of the final model on the augmented dataset with its performance on a dataset without the smote-nc sampling.
- Usage of Ensemble Methods to leverage the best out of the multiple models
- More extensive testing with different metrics
- Testing the ‘user-feeling’. That means simulating a trajectory through Chicago and getting the danger level of passed zones, signaling in case of a dangerous zone. Would we be annoyed by the system? Would we pay attention to it?

Appendix A

Type	QTY (abs.)	QTY (%)	Cost per Acc. (€)	Insurer Cost per Acc. (0.5) (€)	Insurer Cost per Acc. Type (Mio. €)
fatal	22	0.05	1,226,624	612,312	13.4
severe injury	585	1.29	133,501	66,750	39.1
slight injury	4683	10.30	11,433	5,716	26.8
without injury	40164	88.36	6,479	3,239	130.1
Total	**45454**				**209.4**

Cost of accidents for Munich in the year 2017.

Source : Qty from [1], cost calculation from [2]. The cost from [2] include socio-economic cost (human capital loss) which are not all covered by car insurances. We apply the factor 0.5 based on the 'without injury' category as they are the main block in the distribution of accidents. [3], [4] report that the average cost for basic insurance in Germany was around 3300€. However, costs for partial and full insurance as well as technical experts and lawyer fees are not taken into account. With a factor of 0.5 we get an overall weighted average cost per accident of ca. 4600€ which is reasonably close to the reported 3300€ (which does exclude certain costs.)

In the following table we present how the insurer's cost changes with an assumed reduction of accidents of 1%. The difference between total cost shows a 2.1 Mio. € opportunity in cost reduction.

Type	QTY (abs.)	QTY (%)	Cost per Acc. (€)	Insurer Cost per Acc. (0.5) (€)	Insurer Cost per Acc. Type (Mio. €)
fatal	22	0.05	1,226,624	612,312	13.3
severe injury	579	1.29	133,501	66,750	38.7
slight injury	4,636	10.30	11,433	5,716	26.5
without injury	39,762	88.36	6,479	3,239	128.8
Total	**44999**				**209.4**

Cost of accidents for Munich in the year 2017 with an assumed 1% reduction of accidents. Greyed out columns are the same as in the previous table.