Hack the Crack

Luis Antonio Ortega Andrés
José Antonio Álvarez Ocete
Pedro Bonilla Nadal
Francisco Luque Sánchez

1 Introduction

In this document we discuss our approach to the McKinsey&Company challenge in HackUPC 2019. In this challenge we try to predict the severity of car crashes in UK from a given data set.
Our initial attempts consisted in running some basic ML algorithms that did not work out due to the data size and complexity, as expected. We invested a few hours in studying the problem. After applying some preprocessing to the data we run a Random Forest algorithm to predict variable relevance. With this knowledge we used other ML algorithms to obtain our final results.

2 Initial analysis

During this challenge we used both Python processing with Pandas and KNIME. With the latter, we started by running **C4.5** and **Random Forest** algorithms since decisions trees are models from which you can extract information. We obtained `F1-scores` for each class instead of a global one. This showed a fairly good `F1-score` for the non-severe accidents class (around `0.8`) and a critically low one (around `0.01`) for the severe accidents class, the one we are trying to predict. Because of this we realized that the class was completely imbalanced: Only `17.15%` of our dataset were severe accidents.
After this we really dived into the data to gain problem knowledge. On one hand we developed a geographic study. By displaying the latitude and longitude attributes over a world map we discovered a lot of invalid instances. The map also shows a South focused distribution, where the population agglomerates.

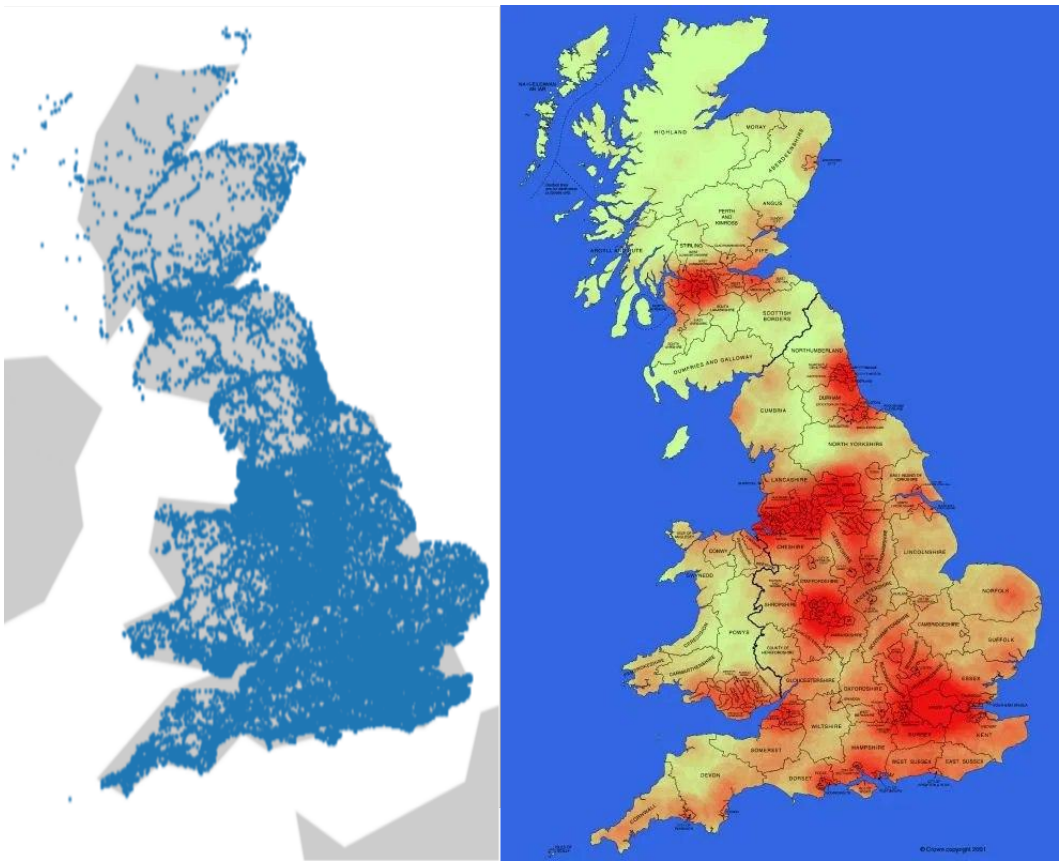

Figure 1: Heatmaps of accidents and population in UK

Figure 2: Accidents in Morocco coast

On this study is also worth mentioning the number of variables that contains geolocation information. Here we include `Latitude`, `Longitude`, `Location Easting OSGR`, `Location Northing OSGR`, `Local Authority (District)`, `Local Authority (Highway Authority - ONS code)` and `Police Force`. In fact, `Police Force` is the one that best summarizes this information since it describes the police force in charge of the area where the accident took place, dividing UK in districts. This reasoning was confirmed afterwards by the **Random Forest** algorithm, which is explained in the next section.

3 Preprocessing strategies

We started by deleting duplicated and invalid instances, such as the accidents that are mapped in Morocco coast. Additionally, a couple of synthetic variables were added in order to potentially help us obtain valuable information from the other variables. These were `Weekday` and

`Day period`.
In order to find dependencies between variables we studied marginal conditional probability between them. These shows huge correlations between the geolocation variable mentioned above, and within some others. By re-running the **Random Forest** algorithm with selected `Police Force` as the most informative variable between them. We filtered the rest of them and run the algorithm again with the same result, assuming our choice was correct.
We have applied diverse reasonings similar to this one to select the variables that give the most insight. We explain here some of them:

- We filtered variables with too much invalid or non computed values such as `pedestrian_crossing-human_control`, `pedestrian_crossing-physical_facilities` and `carriageway_hazards`.
- Converted variables into booleans, such as `Urban or rural`.
- Filtered roads numbers since road classes give more information.
- Deleted the sex of the driver on purpose since we prefer not to take this information into account.
- Transformed other numeric variables into categorical ones, since the number represented a code instead of an actual numerical value.
- Decided to shuffle the instances of the dataset.
- Created a unified dataset using both accidents and vehicles by duplicating accident information.
- Applied class weighting to the severe accidents to overcome the unbalanced dataset problem.
- Divided categorical attributes into boolean ones. These implied other problems since the domain of this attributes changed from the training dataset to the test, meaning that we couldn't train we those values.

Another mayor problem we encountered was the missing or bad data. To fight this issue we took three different approaches, depending on the data characteristics and instances:

- Educated Guessing/mode value: In those cases were there was a strong mode and the bad data tend to have a similar target value we collapsed those into the mode.
- Common-Point Imputation: For real variables, instead of using the mode, we replaced those values with the mean value.
- Listwise Deletion: In very extreme cases we decided to delete all the data from instances with invalid entries.

4 Final model

Although we used various models after gaining enough problem knowledge, the best `F1-score` was still obtained by the **Random Forest** model. By applying cross validation with obtained the following results for the training dataset:

F1-Score	Accuracy
0.35651964122826657	0.7107775245899353

5 Solution: The Builder

The first to take into account when trying to provide solutions to this problem is that not all of our attributes can be easily influenced. For example, the age of the driver involved or the precise maneuver uver they were doing. We present here an actionable plan that focused on the ones within reach.
By looking into the data we are quite concerned about the severity of the accidents that happen within metropolitan areas involving either motorbikes or bicycles. In particular, `30.23%` of accidents involving two wheels means of transport are severe and `16%` of the accidents took place only in the City of London.
Our plan consists on developing two services:

- A software that designs new road network focussing on metropolitan areas. This service would optimize different parameters such as implementation costs and connectivity. It will be called **The Builder** in this document.
- An accident simulator which given a real road network, a dataset of real accidents that took place in the network and a fake road network, provides fake accident data within the new network.

Once the Builder creates a network, a new dataset of fake accident data is built for that network. Our trained model predicts the severity of those accidents and compares them to the real one.
With this approach we provide a tool that given accident data creates new road networks reducing the severity of the its accidents. This would work for any dataset given, not only for UK.
In terms of impact, building this system would be really expensive for arbitrary road networks. In order to minimize that cost we could take into account the current network system to change within a given budget.

6 Other possible solutions

Using this dataset we could consider other studies that would also reduce the accident severity. However, we didn't have the time to look at this in deep. Some of these are studying how impacts with other objects affect severity in order to change them or the first point of impact in a car, to improve car resistance.

7 Personal takeaways

Although we had previously worked on data prediction, this challenge felt like the first real world experience for us. At the beginning we were quite stuck, our initial attempts didn't work at all and the information we obtained didn't feel really useful. However, after we started using some common sense and applied less usual techniques the data started to unfold. These are precisely our takeaways from this challenge: the different approaches we took upon getting stuck as well as how common knowledge can be useful for this kind of problems.
We also learnt a lot about teamwork and coordination since it was our first hackaton.