Understanding the Road Accident in UK

Yushi Yang

October 11, 2020

1 Introduction

1.1 Motivation

Road accidents, unfortunately, happened frequently and they can lead to severe consequences such as fatality. Reducing the number of road accidents, especially the severe ones, is especially important to make people's lives better. Appreciating the inherent randomness of these events, I am trying to understand the deterministic factors in these tragic incidents. For instance, one might intuitively speculate the conditions of the vehicles being relevant, and the drivers with poorly conditioned cars might be more likely to get involved in accidents.

In order to concretely understand what factors might be important, I am going to use statistical methods to find the correlations between different attributes of the car accidents, and then I will try to build a probabilistic model (i.e. machine learning model) to predict the severity of the car accidents, basing on the highly correlated factors. The final model is expected to help the law makers to propose more effective regulations to reduce the number of severe road accidents.

1.2 Elaborating My Ideas

I would love to express the idea in my head with the following graph. Ideally, I can find a nice "feature space" where all the road accident with different severity are separated, like the situation in the left side of Fig. 1. In this case, knowing the feature of the one typical road travel would allow us to *predict* the danger level of such travel. Realistically, it is not possible to find such ideal feature space because the inherent randomness of the road accident. (The noun "accident" already suggests some level of randomness.) This means, under identical conditions, different results can still happen.

However, it is still reasonable to believe that there are *deterministic* factors embedded inside road accidents. For instance, higher speed limit might lead to more severe accidents. This ultimately means a situation exhibited in the right side of Fig. 1, where we can not draw precise boundaries for different severity levels, but generally we can identify "safe" and "dangerous" regions in the feature space. In this case, I can still tell how dangerous a road travel is, in a probabilistic manner. For instance, I would love to say something like "tomorrow is raining, and you are driving in the rural area so you have 60% chance to face a terrible accident, please be extra careful", based on the outcome of the project.

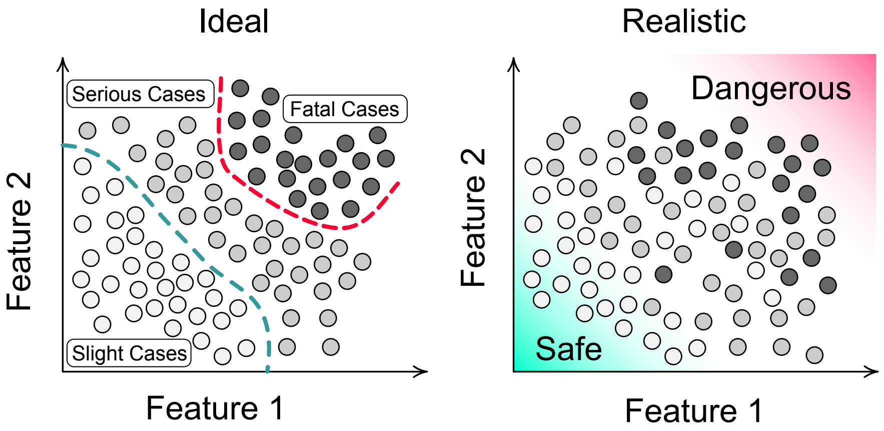

Figure 1: The ideal outcome of this project in my head. Left: the ideal scenario where all the different road accidents have distinct regions in the feature space. Right: a more realistic case where the feature space can be separated into dangerous and safe regions.

2 Data Processing

2.1 Data Source

I used the data from website kaggle (link to the source), which contains the detailed information about traffic accidents across the country from 2004 to 2016. The dataset contains two tables (`.csv` files). One of them is related to the accidents alone and another is related to the conditions of the involved vehicles. The same entry in the two tables is identified by its unique accident ID. There are 55 features in total for each accidental event, such as `Number_of_Casualties` , `Carriageway_Hazards` , and `Vehicle_Manoeuvre` . There are totally 1,793,224 records in these dataset, and the file size of the entire dataset is about 1GB. Further processing is needed to make the data useful.

2.2 Data Cleaning

The data were stored as standard `.csv` format which makes the data cleaning procedure very easy. One can simple use the python library `Pandas` to load the two tables with the help of `read_csv` function. The missing values were represented in different fashions, such as “Data missing or out of range”, “None”, and “Other”. I manually identified all these tokens and replaced them with value `NaN` .

A naïve way to process the missing values is simplying discarding the entires that contains missing values. But there exists some features (columns) whose values were missing for most entries (rows). Therefore I have to do something clever, otherwise I will drop more than 95% of the cases. Practically, I identified the features, whose `NaN` values occupied more than 10% of the entire dataset. I then discarded the such features to preserve more cases.

2.3 Data Processing

The label to be predicted is the `Accident_Severity` , which is categorised into 3 classes `Fatal` , `Serious` and `Slight` . Since there are *orders* in these categories (`Fatal` > `Serious` > `Slight`) so it is possible to convert them into numerical numbers, and study the correlations between features and the result. The very relevant features will be used to build the model.

Additionally, I discovered that the features are *structured*. For instance, the feature `Weather`

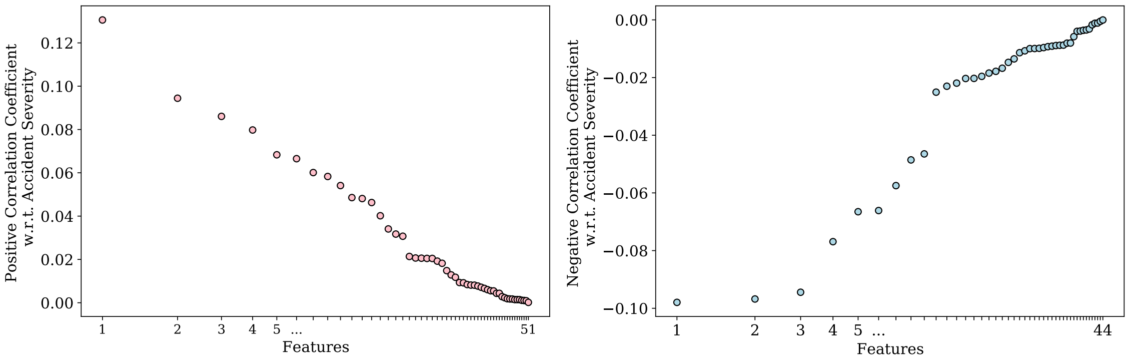

Figure 2: The numerical correlations of different features with the accident severity. Left: different features with positive correlation values. These features indicate a higher chance for bad accident to happen. Right: different features with negative correlations. These features indicate a lower chance for bad accident to happen.

were recorded as `Snowing + high winds` , `Snowing no high winds` , `Fine + high winds` , and `Fine no high winds` . It is therefore very natural to split these values into orthogonal values like `Snowing` , `High winds` , and `Fine` . For these values, I further transform them into numerical values based on their correlation with the accident severity. For instance, `Fine` is positively correlated with the severity so it was mapped to value 1; whereas the weather `Snowing` and `High winds` are not correlated with the severity so there were mapped to value 0. Finally, these "refined" features were used to explore the accident severity, similar to the features presented in Fig. 1. In the figures there are two features but in the real calculation there are 18 features that spenned a 18 dimensional feature space.

3 Understand the data

3.1 Correlation analysis

I analysed the correlation between different features in the table with respect to the accident severity with the help of the `corr` method of the `pandas.DataFrame` objects. The correlations were plotted in Fig. 2, where the positive values were plotted on the left side and the negative values were plotted on the right side. The relatively large correlation coefficient values (~ 0.1) are quite encouraging because if all the features were irrelevant w.r.t. the road accident severity, then all the correlation values would be close to 0. In fact, a quick Monte-Carlo simulation[1] indicated that irrelevant and random feature values for 1,793,224 cases would lead to a correlation value of 0 ± 0.001. This value is far smaller than the observed case.[2]

The features where the highest/lowest correlation values are presented in table 1. The results are in accordance with our daily experience. Specifically, the big motorcycle have the leading correlation value of 13%, which might be related to the fact that these fast vehicles provides little protection to the drivers. Another example is the fact that the leading feature that correlate negatively to the accident severity is "not leaving the carriageway", which is reasonable. If one do not make drastic move during driving, it is not likely that bad accident

[1] Specifically, I calculated the correlation between 1,793,224 labels from $\mathcal{U}(0,2)$ and the same amount of features values from $\mathcal{N}(0,1)$, where $\mathcal{U}$ and $\mathcal{N}$ represent the uniform distribution and normal distribution respectively.

[2] In contrast, if we only have 100 cases, then the random feature values would yield a correlation value of 0 ± 0.1, which would make the observed value 0.1 not surprising. This demonstrates the power of big data: with a lot of data, I am quite confident with my conclusions.

Table 1: Selected correlation values of different features with respect to the accident severity. The positive correlation values indicate the dangerous nature of the corresponding features, and the features with a negative correlation values can be considered as safe-ensuring.

Feature Name	Correlation Value
Vehicle Type: Motorcycle over 500cc	13.06 %
Urban or Rural Area: Rural	9.44 %
Speed limit	8.61 %
Junction Detail: Not at junction or within 20 metres	7.98 %
Sex of Driver: Male	6.65 %
Vehicle Leaving Carriageway: Did not leave carriageway	-9.79 %
Vehicle Type: Car	-9.68 %
Urban or Rural Area: Urban	-9.44 %
Vehicle Manoeuvre: Waiting to go - held up	-7.69 %
Sex of Driver: Female	-6.65 %

would happen. Based on these correlation values, I further refined the existing features and only kept those that have high correlation values for further modelling.

3.2 Exploring the Structure

Apart from the correlation analysis, I further studied the structure of the refined features by the means of the principle component analysis (PCA). The PCA results would give a general view of the distribution/structure of the result. The left subplot in Fig. 3 shows the percentage of variances explained by different numbers of principle axes. The gradual increase shown in the figure indicated that there is no "redundant" features after the refinement. In other words, eliminating any figure would loss some information of the distribution. The right subplot of Fig. 3 illustrates the projection of 50,000 cases on the first two principle axes. The different severity were indicated by different marker styles in the plot. It is very clear that the cases of different severity were "mixed together", being very different from the idealised sketch (shown in Fig. 1). This result means the features space that I am working with is not very nice, and the modelling would be very challenging. Appreciating the stochastic nature of the road accidents, the result is not too surprising.

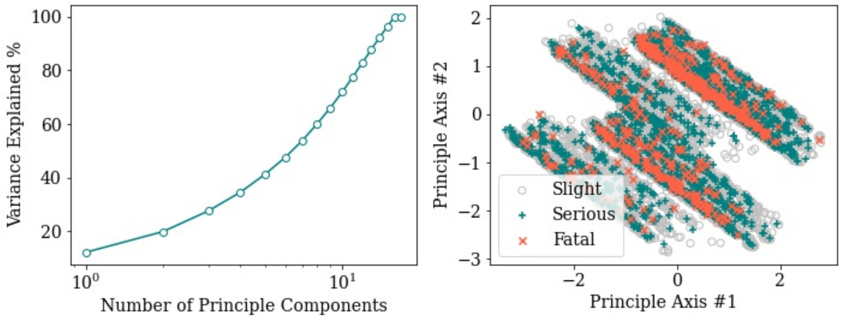

Figure 3: The PCA results. Left: the variance that explained with different numbers of components. The linear increase of the variance values indicates there is no redundant feature. Right: 50,000 cases projected on the first 2 main components. There is no clear separation between features with different accident severity, which means the modelling would be challenging.

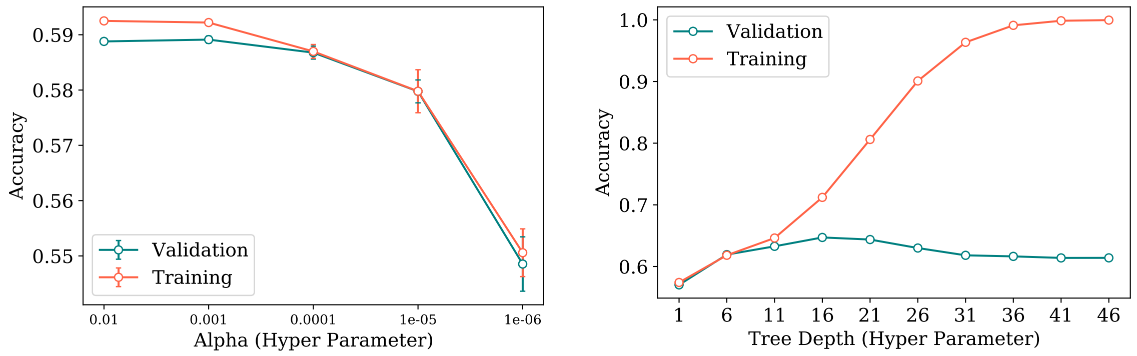

Figure 4: The accuracy values of different models with respect to their hyper parameters. Left: the accuracy of SVM with different regularisation constant (alpha). The error bar represent the standard error from 10 different fitting. Right: the accuracy of decision tree with different tree depth values.

4 Modelling the Data

The data that I had is not balanced. In fact, 90% of the cases were slight, 9% were serious and only 1% were fatal. I combined the serious cases and the fatal cases, so that the final task is to do a binary classification. I further randomly choose equal amount of cases for both classes, so that the training result is not biased. A successful training would yield accuracy higher than 50% in this case. I split my dataset into training, validation and testing sub-groups, where the training set is used for training the model (fitting the parameters); the validation set is used to optimise the hyper-parameters such as the regularisation constant; and the testing set is used to finally evaluate the performance of the model. I tried different models such as the supporting vector machine (SVM), the decision tree, and the neural work, with the help of package `scikit-learn` and `tensorflow` .

4.1 SVM

I tested the support vector machine with linear kernel and optimised the regularisation constant. The final accuracy is presented in the left side of Fig. 4. The result indicates that the best alpha value is around 0.01, and the achieved accuracy is 59%, which is a bit better than a random guess with 50% accuracy. The close matching of the validation accuracy and the training accuracy indicates that the SVM is not overfitting the dataset. However, the low overall accuracy might indicate that the SVM is a highly biased model. The final accuracy of the SVM on the testing set is 54.6%.

4.2 Decision Tree

In addition to the SVM model, I also tried to use the decision tree to fit the datasets. The accuracy values as a function of the depth of the tree is shown in the right subplot of Fig. 4. The situation is very similar to that of the SVM, and we are overfitting our data, were I end up with a maximum validation accuracy of around 65% when the tree depth is 16. However, for the decision tree, the model is indeed overfitting the dataset as the training accuracy is far larger than the validation counterpart. This indicates that the decision tree might be a good model for my purpose. However, I will need far more data in order to beat the overfitting problem. The final accuracy of the decision tree on the testing dataset is 63.4%.

4.3 Neural Network

In order to ultimately test the machine-learning approach, I used a deep neural network which is composed of 17 input neurons, plus a hidden layer of 50 neurons, and then two output neurons that represents the probabilities of a sample being serious or not. The training procedure is illustrated in Fig. 5 where the validation accuracy and the training accuracy were plotted as a function of the epoch number. By gradually increasing the training steps, both quantities increased which indicates that the model is not overfitting the dataset. I would expect the model getting more accurate by increasing the training time or by increasing the complexity of the structure of the network. However, due to the limited time and limited computational resources available to me, I have to stop there and report the current result. The final accuracy on the testing set is 62%.

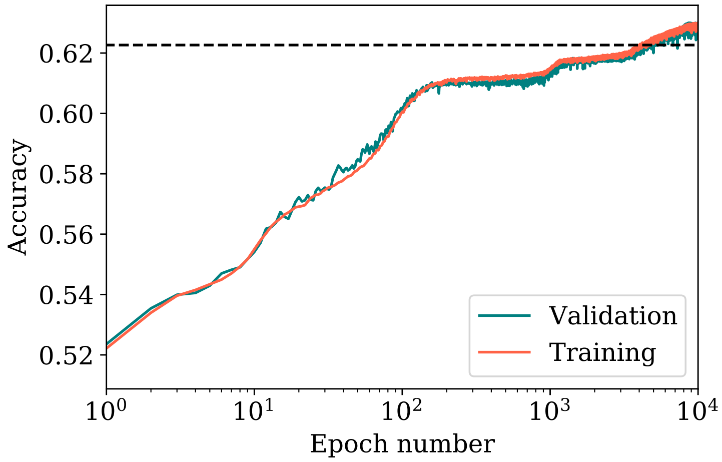

Figure 5: The accuracy values of the neural network model during the training process.

5 Conclusion

In this project, I analysed the road accident in United Kingdom from year 2004 to 2016. After data cleaning, I find the numerical correlation of different factors that would link to the road accident. Seventeen features were refined based on the numerical correlations, and were used to train a probabilistic model. The neural network and the decision tree had better performance comparing with the SVM. The highest accuracy I obtained so far is 63.4%, but a more complex neural network with longer training time is expected to yield a far better outcome.

The correlation of the different features with the road accident severity presented in this report is an informative summary of the dangerous and save-ensuring factors during a road trip. A more accurate model is expected to be finished in the future, that can help people to estimate the safety level of their travel.