Big Data Processing and Applications

Prediction of the traffic accidents and their severity from UK, and data analyses

Roni Latva

Danila Goncharenko

Santtu Orava

Casimir Saastamoinen

Project description

The project is focused on analyzing a dataset consisting of data about traffic accidents in the UK using the PySpark environment. The dataset used for the analysis is a comprehensive collection of traffic data collected by the UK government between 2000 and 2016, comprising over 1.6 million recorded accidents.

The use of PySpark is required in this project to make it possible to analyze the dataset in a reasonable time. First the dataset must be preprocessed and cleaned for a high quality analysis. This will ensure that the results will be easier to produce and represent in a more readable format. This format will most likely be graphs that show the correlation between the different features of the dataset. These possible correlations can be between severity and time of day or speed limit and weather conditions for example.

These graphs will be used to identify trends and patterns in accidents and their severity through time and how they have or have not changed over the years. The goal for this project is to create graphs that highlight the trends and predict how severe an accident would be for the given situation. These results can then be used to adjust the speed limits for the given conditions which can be bad weather, road quality or lighting or the type of junction control.

Related work

There are lots of research papers published on the topic of traffic accident prediction in the UK and the world at large. The attention received by this topic is evident since it has great economic and social impacts on the lives of billions of people. In the UK alone it has been reported that around 1,600 people were killed and more than 26,000 were severely injured in 2021 [1]. Worldwide around 1.35 million people die on roads each year [2]. More than half of the fatalities consist of pedestrians, cyclists and motorcyclists. Given that humans are responsible for building and designing the traffic systems around the world it is reasonable to expect that with deeper knowledge and better planning these numbers can be reduced. If right decisions are made regarding traffic planning it may be possible to lessen the negative impacts traffic accidents have both economically and socially.

One excellent example of big data analysis for the creation of a traffic prediction model is “Live Prediction of Traffic Accident Risks Using Machine Learning and Google Maps” by Meraldo Antonio [3]. The author worked with the same Kaggle’s dataset, complemented by a DarkSky’s dataset with more accurate weather conditions. Meraldo Antonio notices that in Kaggle’s dataset weather is assumed to be constant throughout the day, thus additional dataset was used. The location of the traffic accidents was visually presented in Google maps. Overall, the article uses a variety of tools that makes the results more realistic. It can be compared to the results obtained in this project to see how much precision was lost because of inaccurate weather conditions in Kaggle’s dataset.

Meraldo Antonio does not show the full code, with which the analysis was performed. Nevertheless, the full description for big data analysis in PySpark can be found in the second example - “Big data Analysis of Road Crash Data using PySpark with PySpark Tutorial” by Rakesh Nain [4]. The article discusses road crashes in South Australia. It provides in-depth explanations of how the analysis was conducted in PySpark, making it a valuable reference for this project. In addition, the article gave the idea of visual design for graphs and charts that can be used in this project to present the results of analysis.

Another example, in which a prediction model for UK traffic accidents was created is “Analysis and visualization of accidents severity based on LightGBM-TPE” by Kun Li, Haocheng Xu, and Xiao Liu [5]. The article uses a different Kaggle’s dataset, which was collected in 2017. It aims to determine which factors play the biggest roles in accident severity using LightGBM-TPE algorithms. The authors visually represent all their discoveries, which showed that the highest fatal accident ratio is in high-Latitude regions and that its ratio peak is at 4:00 am. It would be interesting to see how the situation with UK traffic accidents had changed in 2017, compared to the 2005-2014 dataset.

The predictive power of machine learning methods, when it comes to accident severity prediction, have been demonstrated by many other papers, evaluating many different architectures. Models utilizing support vector machines, k-nearest neighbors and multilayer perceptron networks have achieved over 90% accuracy in severity prediction tasks [6]. This indicates that there are many different ways of obtaining adequate prediction performance in this project. It is also notable that the features used in the training have differing impacts on the model based on model architecture. Extracting the most significant factors for our model could be an interesting addition to our results analysis.

Some more advanced methods utilize autoencoders to encode the data, reducing its dimensionality while retaining the same patterns [7]. This means the input vector is encoded to a smaller size which is then passed to a classifier network for predicting the accident severity. Doing this improves training speed by reducing the size of the neural network with essentially no observed performance drawbacks.

Dataset description

The dataset used for this project comes from the UK government statistics on traffic accidents. The dataset covers accidents from years 2005-2007 and 2009-2014 along with the average annual daily flow for all roads in the UK for the same time period. Altogether the data consists of four csv files, one for the AADT values and three for the accident details. The combined size of the data is 465 Mb. In the AADT file the AADT values are presented for separate vehicle classes ranging from cycles to buses and trucks. Total number of columns is 29 but many of them are not strictly necessary to the data processing task. The flow statistics can possibly be linked to the accident data through shared location data between the two tables if we want to include them in the analysis.

Data in the accident files includes 33 columns. Important information includes spatial information like location, road, weather and lighting conditions and temporal information like time of day, day of the week and date. Additional important information includes the speed limit, details about potential junctions and if pedestrians were involved. The label description is included in a separate documentation [8].

Overall the data should allow for analysis of multiple factors when it comes to predicting the severity of a traffic accident. The spatial and temporal components could be analyzed separately to find out whether there are identifiable trends in these aspects. Other factors such as road conditions, amount of traffic and junction data could also be investigated separately to reveal which factors influence accident severity the most.

Methods and tools

We will utilize SparkSession from PySpark for analyzing the dataset. Since the dataset is large we will be using PySpark's Spark Streaming, so we can analyze the data while it is being processed. We will analyze the different variables and draw graphs from the ones that could correlate. Analyzing correlation of different feature values with accident severity will constitute the inferential statistical analysis portion of the project where we aim to highlight trends in the data. We will then move onto predictive analysis where we aim to utilize a multilayer perceptron network to predict traffic accident severity based on the input features. The network will get as input a vector of values capturing the different feature values in the dataset and its output size will be three to capture the three different severity levels defined in the dataset. The work will be compiled into a jupyter notebook environment where experiments can be repeated and modified easily.

Preprocessing

Data cleaning was the first step in the preprocessing of the raw data. The features (related with the "Accident Severity") that were considered in the analysis are the location ("Longitude", "Latitude", "LSOA_of_Accident_Location"), the timing of the accident ("Time", "Date", "Day_of_Week"), and the environment conditions ("Light Conditions", "Weather Conditions", "Road Surface Conditions", "Speed Limit", "Road Type").

Data formatting was the second step in the preprocessing of the raw data. A new variable "Month" was created based on the variable "Date". A new variable "Hour" was created based on the variable "Time". Numerical variable "Day_of_Week" was converted into a string. Numerical values in "Accident Severity" were converted into "Fatal", "Serious", and "Slight".

Heatmaps were used to visualize the data patterns.

Prediction model

Our goal for the second part of the project was to construct a prediction model that would be able to predict the 'accident severity' of an accident based on the other features of that accident. The model architecture was inspired by the template of course exercise 4 which

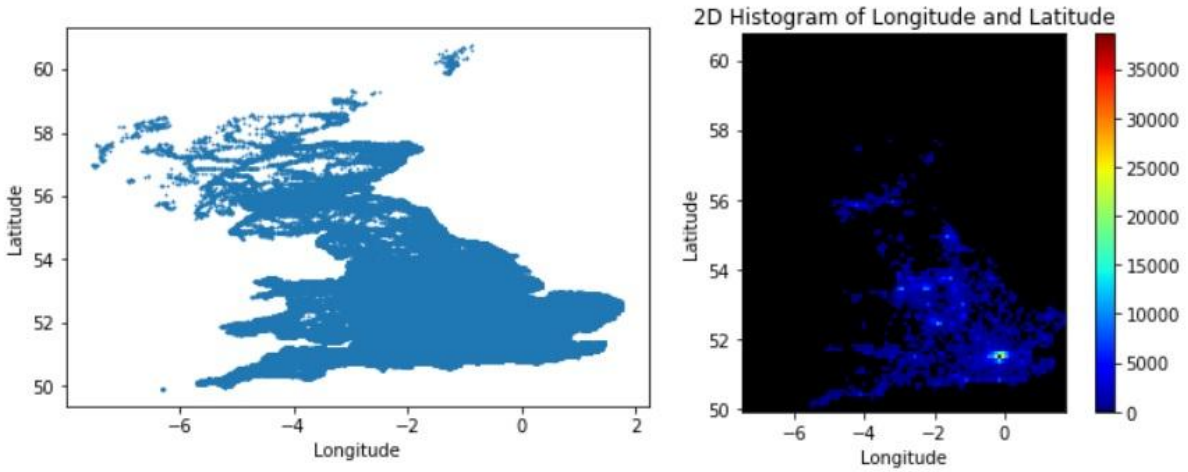
uses a multilayer perceptron classifier. The data is transformed by using the 'StringIndexer' from PySpark which assigns numeric values to columns containing string values. The features are then collected into a feature vector using PySpark's 'VectorAssembler' after which the data can be fed into a specified model to be optimized for. After the model has been fit to the data we evaluate the model by measuring precision, recall and accuracy and by constructing the confusion matrix for the test data. The confusion matrix is a particularly useful element of the pipeline as it clearly highlights issues that can arise in machine learning which we will get into in the data analysis section.

We chose to approach the problem with the exercise template as our guide due to the relatively simple structure of the provided notebook and the seeming congruence with our chosen topic. Moreover, our thought was that the exact architecture of the model should not be that significant to the general performance of the model and as long as we got an error-free training run done the model should perform 'well enough'. Both of these initial assumptions turned out to be at least partially incorrect and we ended up needing to do a lot of extra work and investigating we did not foresee. These aspects of the project will also be unpacked as part of the data analysis.

Data analysis

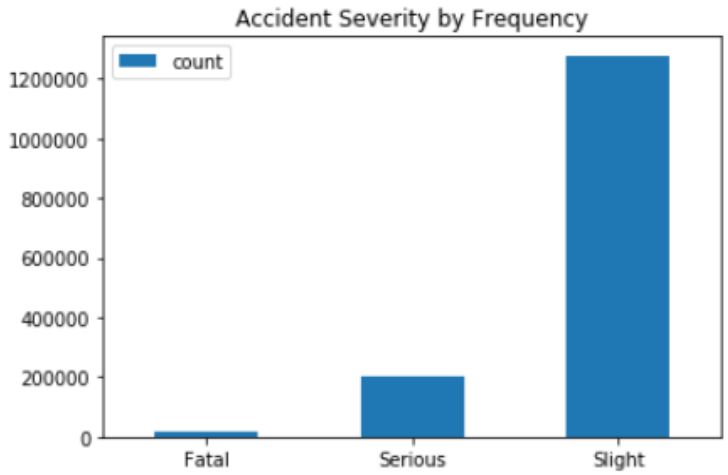
Processed data analysis

Longitude and Latitude were plotted to see if there are any outliers in the data. As can be seen, the data for northern Ireland is missing. The accidents appear most frequently in the big cities, such as London, Birmingham, Liverpool, Manchester and Sheffield.



Map of accidents by geographical location with frequency

The fatal accidents rarely occur in the data, so the slight accidents shape most of the data tendencies. Therefore fatal and non-fatal accidents will be analyzed separately on the basis of environmental conditions.



Accident count divided into 3 severities

The relation between natural environment conditions (weather, light, and road surface conditions) and fatal and non-fatal accidents was studied first. The "Count" will show how often certain conditions appear in the data. The "Fatal_%" and "NonFt_%" shows the percentage of certain conditions appearing in the data from the fatal and non-fatal data. "Total_%" shows the percentage from the whole data.

Fatal accidents

Weather_Conditions	Count	Fatal_%	Total_%
Raining without high winds	1854	9.54	0.12
Snowing with high winds	13	0.07	0.0
Snowing without high winds	90	0.46	0.01
Unknown	232	1.19	0.02
Other	294	1.51	0.02
Fine with high winds	339	1.75	0.02
Fine without high winds	16144	83.11	1.07
Raining with high winds	280	1.44	0.02
Fog or mist	179	0.92	0.01

Non-fatal accidents

Weather_Conditions	Count	NonFt_%	Total_%
Raining without high winds	175785	11.86	11.7
Snowing with high winds	1947	0.13	0.13
Snowing without high winds	11210	0.76	0.75
Unknown	26545	1.79	1.77
Other	33143	2.24	2.21
Fine with high winds	18008	1.21	1.2
Fine without high winds	1187359	80.09	79.05
Raining with high winds	20530	1.38	1.37
Fog or mist	8005	0.54	0.53

Fatal and non-fatal accidents by weather conditions

Most of the accidents (fatal and non-fatal) happen in fine weather without high wind, because this is the most common weather throughout the year. There is no clear weather factor that could affect the severity of the accident.

Fatal accidents

Light_Conditions	Count	Fatal_%	Total_%
Daylight: Street light present	11456	58.98	0.76
Darkeness: No street lighting	3597	18.52	0.24
Darkness: Street lights present and lit	4055	20.88	0.27
Darkness: Street lights present but unlit	127	0.65	0.01
Darkness: Street lighting unknown	190	0.98	0.01

Non-fatal accidents

Light_Conditions	Count	NonFt_%	Total_%
Daylight: Street light present	1089189	73.47	72.52
Darkeness: No street lighting	78882	5.32	5.25
Darkness: Street lights present and lit	291893	19.69	19.43
Darkness: Street lights present but unlit	6774	0.46	0.45
Darkness: Street lighting unknown	15794	1.07	1.05

Fatal and non-fatal accidents by lighting condition

In light conditions, fatal accidents are more likely to occur when street lights are not present.

Fatal accidents

Road_Surface_Conditions	Count	Fatal_%	Total_%
Flood (Over 3cm of water)	41	0.21	0.0
Frost/Ice	326	1.68	0.02
Wet/Damp	5955	30.66	0.4
Dry	13029	67.07	0.87
Snow	74	0.38	0.0

Non-fatal accidents

Road_Surface_Conditions	Count	NonFt_%	Total_%
Flood (Over 3cm of water)	2102	0.14	0.14
Frost/Ice	31073	2.1	2.07
Wet/Damp	417466	28.16	27.79
Dry	1021468	68.9	68.01
Snow	10423	0.7	0.69

Fatal and non-fatal accidents by road surface conditions

A lot of accidents happen when the road surface is wet or damp.

Fatal accidents

Speed_Limit	Count	Fatal_%	Total_%
10	2	0.01	0.0
20	99	0.51	0.01
30	6490	33.41	0.43
40	1782	9.17	0.12
50	1044	5.37	0.07
60	7472	38.47	0.5
70	2536	13.06	0.17

Non-fatal accidents

Speed_Limit	Count	NonFt_%	Total_%
10	12	0.0	0.0
15	10	0.0	0.0
20	17047	1.15	1.13
30	960328	64.78	63.94
40	120448	8.12	8.02
50	47703	3.22	3.18
60	230358	15.54	15.34
70	106626	7.19	7.1

Fatal and non-fatal accidents by speed limit (mph)

Interestingly, there have been some fatal accidents reported in areas with small speed limits.

Fatal accidents

Road_Type	Count	Fatal_%	Total_%
Slip road	127	0.65	0.01
One way street	202	1.04	0.01
Unknown	60	0.31	0.0
Roundabout	307	1.58	0.02
Single carriageway	14860	76.5	0.99
Dual carriageway	3869	19.92	0.26

Non-fatal accidents

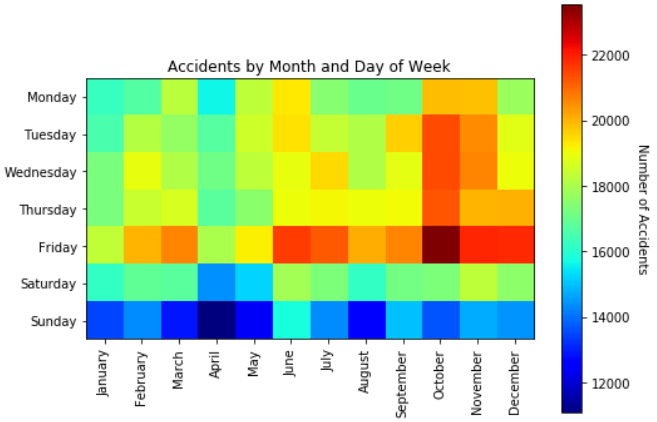
Road_Type	Count	NonFt_%	Total_%
Slip road	15524	1.05	1.03
One way street	30712	2.07	2.04
Unknown	8262	0.56	0.55
Roundabout	99930	6.74	6.65
Single carriageway	1110426	74.9	73.93
Dual carriageway	217678	14.68	14.49

Fatal and non-fatal accidents by road type

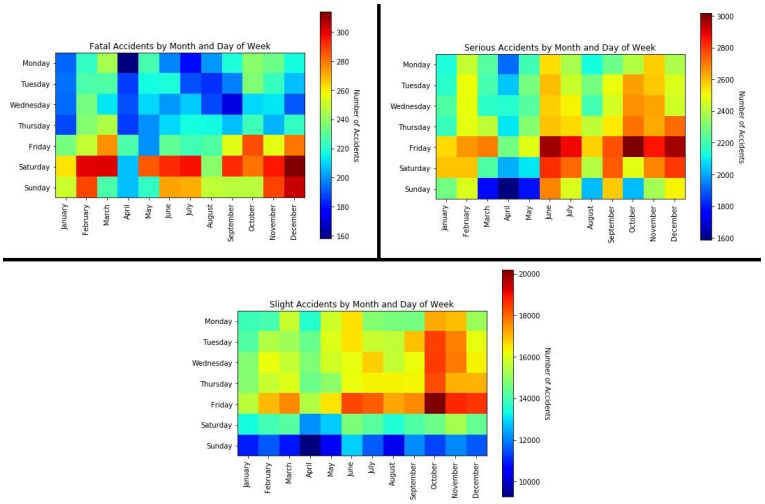
Accidents are more likely to be fatal on a dual carriageway, and fatal accidents happen on roundabouts less often than non-fatal accidents.

Heatmaps for accident count on different days and months were plotted. It can be compared to the similar map in the article by Rakesh Nain. As we can see, the article used a smaller sample of data, but overall tendencies shown there remain for big data as well. Sunday is the day with the least accidents, our guess is that Sunday is the least busy day on the road, leading to less potential accidents. On the other hand, Friday has the most accidents for a weekday. It is the first day of the weekend, so more car traffic and people driving back from partying, while not in the best state of awareness, increasing the risk of an accident.

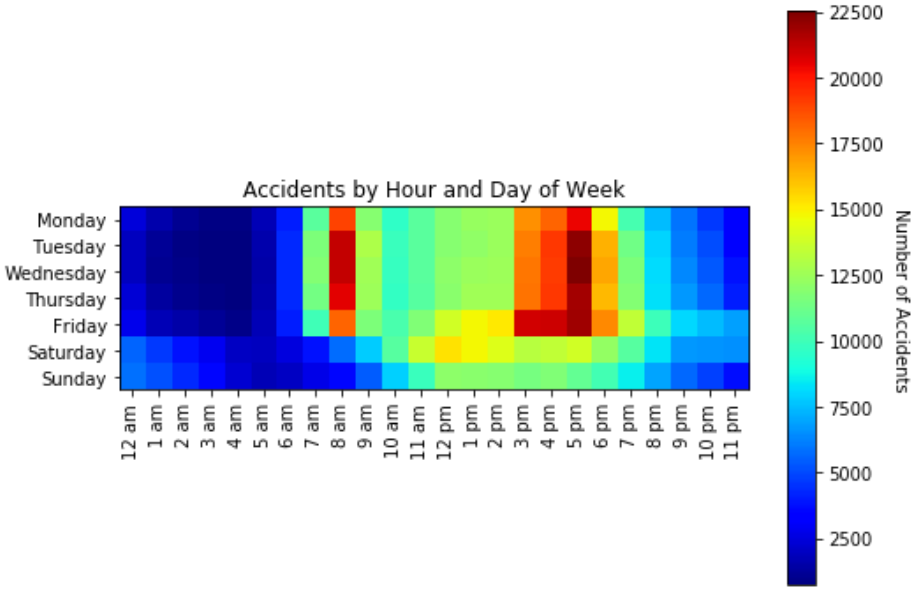
For months, October and November have the most accidents. This might be a combination of many factors. Firstly, lighting conditions get worse, because of reduced sunlight. Secondly, temperatures drop and rain increases causing the road to get wet and slippery. Summer also has a greater number of accidents, probably because of increased traffic from vacation trips.



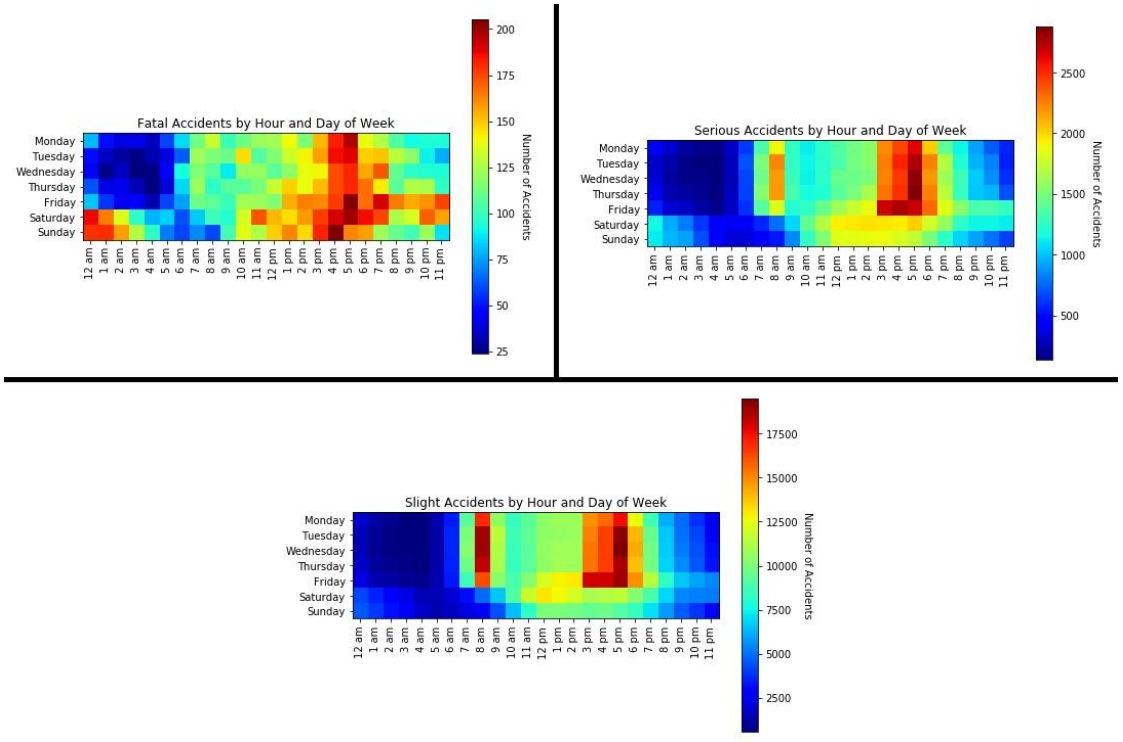
In the total accident graph the slight accidents probably dominate the data because of the disproportionate distribution. When we plot the three different severities independently, interesting patterns emerge. Fatal accidents seem to be concentrated on the weekends, Probably because of increased intoxication and general lackadaisical attitude at the wheel. When we move down on the severity scale we move more towards the day and the weekday workdays. People might be more aware of their surroundings and less under the influence, so accidents still happen due to greater traffic flow, but the severity of them are reduced.



From the time of day graph we can see that most accidents are concentrated on regular commute times in the morning with a longer window in the evening, maybe due to some people driving to leisure activities after work. We can see traffic decreasing on the weekends but with more accidents at night.



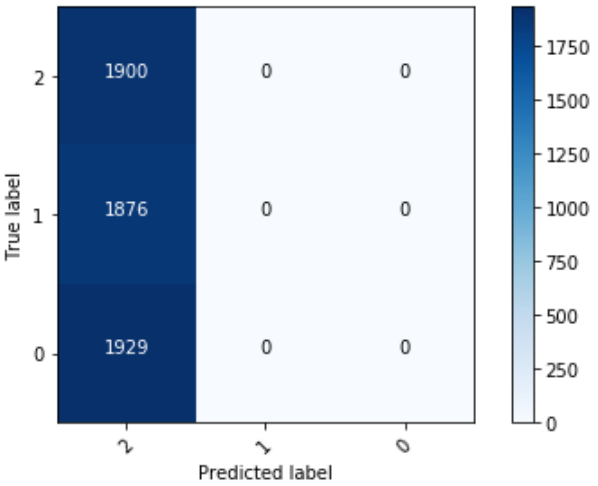
When we again split the graph into the three severity categories, we can again see some familiar patterns. Slight accidents graph looks very similar to the original, while the more severe the accident the more likely it is to happen at night and on a weekend. Fatal accidents are more rare so the data has more noise in it, but the general picture is still clear. An interesting note is that slight accidents seem to concentrate on the mornings.



Prediction data analysis.

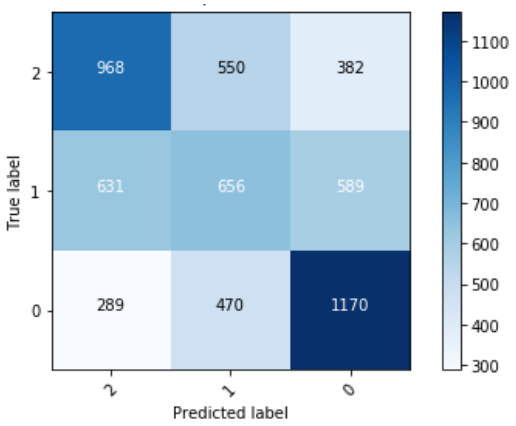
In order to apply the workflow presented in the fourth exercise to our dataset some changes were warranted due to high-level differences in the dataset structure in comparison to the one used in the exercise. Firstly, the dataset for this project contains three different classes where the other had only two. This also meant the column values had to be remapped as the model pipeline only accepts consecutive integer values starting from zero in the label column. Secondly the dataset needed to be resampled heavily as the classes were extremely imbalanced. We experimented with both downsampling the two more frequent classes to the level of the minority class and upsampling the smallest class up to the level of the middle class while again downsampling the majority class. It is noteworthy to point out that when the dataset is not balanced models can naively achieve great accuracies while being completely unable to distinguish between classes. This is because guessing only a single class for all samples when for example 90% of the samples are from that class will yield 90% accuracy for the total dataset but 0% for all classes other than the majority one.

After these preprocessing steps we proceeded with creating the feature vectors using Pyspark's own stringindexer and vectorassembler. This step handled the encodings of string columns to machine interpretable numeric values and packed values to form the vectors. After also creating the dataset partitions into train, test and validation sets the feature vectors were fed into the model. The results here were unfortunately poor as the model completely failed to distinguish classes from each other and simply assigned all samples to a single class as can be seen in the confusion matrix. We experimented extensively with different feature vector selections, model configurations and the above mentioned resampling schemes but the results were consistently unsatisfactory.

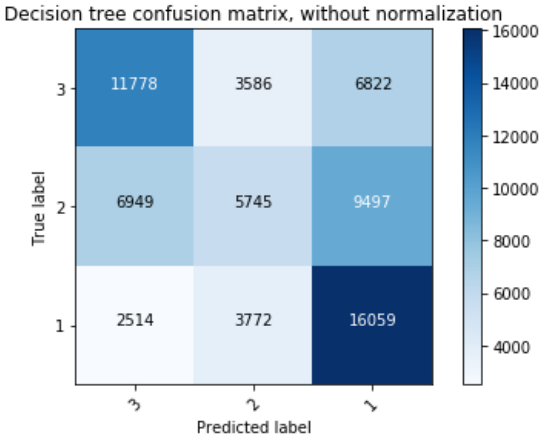


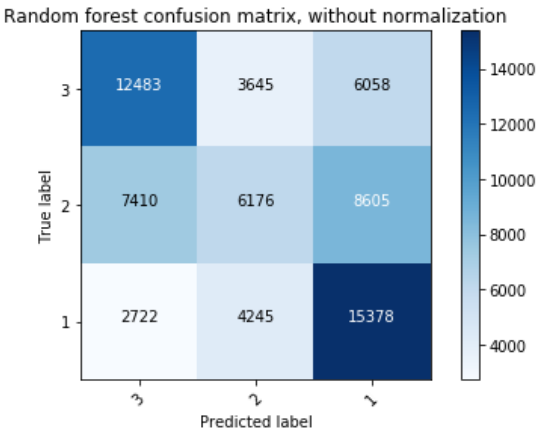
Since our approach with the stringindexer had failed we moved on to constructing an original pipeline for string column embedding and also column value normalization which was not really possible with stringindexer as the values were newer assigned to the dataframe but instead instantiated during the fitting process. The embedding process has two parts. First, we create a mapping in the form of a dictionary that links all unique string values present in a given column with an integer, with the most common string getting the highest value and the

least common getting the value zero. Secondly, we construct an accumulator that holds multiple when clauses that can then be asserted onto the given column changing all its values. Once this has been done on all string columns we normalize all rows to a specified range. We experimented with both 0 to 1 and -1 to 1. The label column 'Accident_Severity' is not normalized. For construction of the feature vectors we still utilize the exercise 4 pipeline but all columns are now handled as numeric columns. The fit on the model is better than before and we are actually able to distinguish between classes at a level better than chance. The model achieves ~50% accuracy which is somewhat impressive as the baseline accuracy for random guessing is ~33% for a three class problem. Confusion matrix below.



After achieving these results we again spent a lot of time tinkering with feature vector selection and different model parameters but the results never improved in a meaningful way. This ultimately led us to experiment with two additional classifiers: decision trees and random forests. We ran these classifiers with and without our own normalization and the resulting accuracies were around 50%. With the resulting confusion matrices being very similar to the confusion matrix which the multilayer perceptron classifier produced. We tried different combinations of features and all of them produced very similar results to each other regardless of classifier.





As a final note on the prediction model development we also attempted to simplify the problem by reducing the class count to two. This was done by removing the middle class

from the analysis altogether and proceeding as before otherwise. This resulted in around 70% accuracy which is decent compared to random guessing baseline which is 50%. Still the classification problem is by no means solved.

Results

Most accidents happen during the autumn on the weekdays, especially friday. Rush hours in the morning and in the evening are the most dangerous. The most frequent locations for the accidents are the large cities in the UK. London has the most accidents, as to be expected. Among the environmental conditions, the factors, which greatly increase the probability of an accident to be severe, are no street lights, wet road surface, high speed limit. Fatal accidents often happen on dual carriageways. Based on the distribution of fatal and non-fatal accidents across different weather conditions, it seems that weather does not have a significant impact on the severity of accidents. Fatal accidents are more frequent on weekends and during late night hours (12-1 am), in contrast to non-fatal accidents.

Our best prediction models we were able to produce had about 50% accuracy with the given three classes. This means that our models were better than a random guesser. We noticed that the models were better at classifying the most severe and mild cases while they were unable to predict the middle value accurately. The accuracy models for this data in other works range from 60 to 80 percent. This is the accuracy we had before balancing the data and our model guessed most if not all classes to the most frequent class which in this case means the most mild accidents. In one of the other works which had a considerably more accurate model the data imbalance seems to not have been taken into consideration. Our experiments with the three different classifiers and multiple prediction models suggest that this might be a common issue with the prediction models created from this data.

Conclusion

Our experiments with this data suggest that it can not be used to build a model for accurately classifying all the different severities. The different data points do not seem to correlate with the severity of the accident enough to make the severity predictable.

The dataset might need more accurate weather descriptions and new data points such as the speeds of the vehicles that were part of the accident and information about the drivers, were they sober or under influence. With these improvements the classifiers used in this study could be tried again to see if the predictability increases. Some new classifiers should also be tested to see if they perform better than the three we used.

Overall the project was educational both in terms of the new information we learned about the topic itself and also with regard to the technical execution aspects of operating in PySpark and the implementation of many data processing steps. The topic is likely to be the subject of continued research due to the large impact it still has on a vast amount of people around the world. Understanding the links between causes and effects in the context of traffic safety will stay relevant as the downsides of serious accidents will not stop accumulating. The sector is

likely to also go under meaningful changes as autonomous vehicles and low-noise electric cars begin to really gain market share. These new challenges will require researchers to keep innovating in the field of traffic safety, accident prediction and ultimately accident prevention.

Contribution report

Roni - Imported template, partially wrote related works, data description and methodology sections, methods and tools: prediction model and prediction model data analysis, did coding work on predictive model

Casimir - Wrote most of the project description, partially wrote method and tools, wrote paragraphs to results and conclusions, worked on building the prediction models; made and tested the decision tree and random forest models and wrote about them to data analysis

Danila - Related works, did coding work on raw data preprocessing, processed data analysis (graphs and plots), Made “RawDataProccessing.ipynb”. Other adjustments in the report (methods and tools, data analysis, results). Imported the dataset into a cluster.

Santtu - Related works, dataset description, data analysis

References

[1]https://www.brake.org.uk/get-involved/take-action/mybrake/knowledge-centre/uk-road-safety

[2] https://www.cdc.gov/injury/features/global-road-safety/index.html

[3]https://towardsdatascience.com/live-prediction-of-traffic-accident-risks-using-machine-learning-and-google-maps-d2eeffb9389e

[4]https://medium.com/rakesh-nain/big-data-analysis-of-road-crash-data-using-pyspark-with-pyspark-tutorial-c78ff2c35588

[5] https://www.sciencedirect.com/science/article/pii/S0960077922001977

[6] https://ieeexplore-ieee-org.pc124152.oulu.fi:9443/document/9588242

[7] https://www.rivas.ai/pdfs/bibb2021predicting.pdf

[8]https://beta.ukdataservice.ac.uk/datacatalogue/studies/study?id=7752&type=Data%20catalogue#!/documentation