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

Chandramani
College of Mathematics and Natural Science
University of Maryland College Park
chandramani05@gmail.com

***Abstract*—The project centers on the analysis of a comprehensive dataset encompassing traffic accident data in the UK, employing the PySpark environment. This dataset, compiled by the UK government from 2000 to 2016, comprises over 1.6 million recorded accidents. PySpark is essential for efficient analysis, considering the dataset's size and complexity.**

To ensure a rigorous analysis, the dataset undergoes thorough preprocessing and cleaning. This step aims to enhance the data quality, facilitating the generation of meaningful and visually interpretable results, likely presented in graphical formats. These visualizations may illustrate correlations between various dataset features, such as the relationship between accident severity and factors like time of day, speed limit, or weather conditions.

The generated graphs serve the purpose of identifying trends and patterns in accidents over time, shedding light on the evolving nature of accidents throughout the years. The project's ultimate goal is to create graphs that not only highlight these trends but also enable the prediction of accident severity under specific conditions. This predictive capability holds potential applications, such as adjusting speed limits based on factors like adverse weather, road conditions, lighting, or junction control

***Index Terms*—Data Analytics, Spark, Machine Learning, Traffic Accident Severity**

I. Introduction

In this project, we are exploring new ways to better predict how severe traffic accidents are in the United Kingdom. We will be working with a dataset that was put together by the UK government with over 1.6 million detailed traffic accident records from 2000 to 2016. We want to build machine learning models that can classify accidents as fatal, serious or slight, focusing our analysis on factors like weather, road conditions, time of day and location that may impact accident severity.

A key part of our approach involves taking a close look at which factors seem to have the biggest influence on accident severity outcomes. We hope this report sheds light on what dynamics and patterns in the data help explain which factors led some accidents end up being more severe than others. We also assess how well our models are actually able to make accurate predictions on accident severity categories, using several performance measurements as benchmarks. Our goal is to fine-tune and optimize our models as much as possible to get the best prediction capabilities.

II. Related Work

There are lots of research papers published on the topic of traffic accident prediction in the UK and the world at large, emphasizing the global significance of this topic due to its profound economic and societal impacts. In the UK alone, statistics from 2021 reveal over 1,600 fatalities and 26,000 severe injuries attributed to traffic accidents [1]. Globally, an alarming 1.35 million lives are lost on roads annually, with pedestrians, cyclists, and motorcyclists comprising more than half of the fatalities [2]. Acknowledging human responsibility in designing traffic systems worldwide, it becomes evident that informed decisions and meticulous planning can potentially mitigate the substantial negative repercussions of traffic accidents.

An exemplary work in big data analysis for traffic prediction is ”Live Prediction of Traffic Accident Risks Using Machine Learning and Google Maps” by Meraldo Antonio . Leveraging Kaggle's dataset supplemented with DarkSky's more precise weather conditions, the author visually presents accident locations on Google Maps. Despite the absence of full code disclosure, this work serves as a benchmark for comparison, particularly in addressing Kaggle's dataset's limitations in representing varying weather conditions throughout the day. [3]

For comprehensive guidance on big data analysis using PySpark, ”Big Data Analysis of Road Crash Data using PySpark with PySpark Tutorial” by Rakesh Nain [4] proves invaluable. Focused on road crashes in South Australia, the article provides detailed insights into PySpark analysis methodologies, offering a reference for the present project. Additionally, it contributes to the visual design approach for presenting analysis results.

In another instance, ”Analysis and visualization of accidents severity based on LightGBM-TPE” [5] delves into building a prediction model for UK traffic accidents using LightGBM-TPE algorithms and a Kaggle dataset from 2017. The authors visually represent their findings, unveiling insights such as the highest fatal accident ratio in high-latitude regions peaking at 4:00 am. A comparative analysis with the 2005-2014 dataset could offer valuable insights into the evolving landscape of UK traffic accidents.

Numerous research papers underscore the efficacy of machine learning methods in accident severity prediction, achieving over 90% accuracy with models utilizing support vector machines, k-nearest neighbors, and multilayer perceptron networks . Notably, the choice of model architecture influences the impact of features on predictive performance, emphasizing the need to identify significant factors for the model employed in this project.

Advanced methodologies, such as using autoencoders to reduce data dimensionality while retaining patterns [6], demonstrate the evolution of techniques. This involves encoding input vectors to a smaller size for efficient training of neural networks, showcasing improved speed without compromising performance.

III. Dataset Description

This project utilizes a dataset sourced from UK government records on traffic incidents. The dataset spans accidents occurring between 2005-2007 and 2009-2014, alongside average daily traffic flow for all UK roads during the same periods. Comprising four CSV files— one for Annual Average Daily Traffic (AADT) values and three detailing accidents—the data's total size is 465 MB. The AADT file categorizes traffic flow for various vehicle classes.

Accident data, stored across three files, comprises 33 columns, encompassing spatial details like location, road, weather, and lighting conditions, and temporal aspects such as time of day, day of the week, and date. Additional information covers speed limits, junction details, and pedestrian involvement, with label descriptions provided in separate documentation.

The dataset facilitates the analysis of multiple factors influencing traffic accident severity. Spatial and temporal elements can be scrutinized independently to identify trends. Other factors, including road conditions, traffic volume, and junction specifics, can also be examined separately to pinpoint the most impactful influences on accident severity.

IV. Method and Tools

We'll use SparkSession from PySpark [7] to analyze our dataset, ensuring efficient processing of the large dataset. To keep up with the data as it's processed, we'll employ PySpark's Spark Streaming. Our focus is on understanding the relationships between various factors and accident severity, depicted through graphs. This analysis aims to uncover trends in the data. Moving forward, we'll transition to predictive analysis using a multilayer perceptron network. This network takes a vector of values as input, representing different features in the dataset, and produces an output of three sizes, corresponding to the three severity levels in our dataset. All this work will be compiled in a Jupyter notebook for easy experimentation and modification.

A. Preprocessing

We started by cleaning up the raw data to get it ready for analysis. We focused on specific features related to "Accident Severity," such as where the accident happened (longitude, latitude, location code), when it occurred (time, date, day of the week), and environmental conditions like light, weather, road surface, speed limit, and road type.

After cleaning, we organized the data for better understanding. We created a new variable called "Month" based on the date and another called "Hour" based on the time. We also changed the numerical representation of "Day_of_Week" into a more understandable string format. For "Accident Severity," we transformed numerical values into categories like "Fatal," "Serious," and "Slight."

To grasp the data's patterns more easily, we used heatmaps, which are visual tools displaying information in a color-coded way. This helped us uncover insights and trends in the dataset.

B. 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.

I experimented with two models:

- Gradient Boosting [8]
- Deep Neural Networks (DNN). [7]

Regarding the DNN architecture, the model consists of several layers. It starts with an input layer of 128 neurons using the ReLU activation function, followed by dropout layers to prevent overfitting. The subsequent layers have 64, 32, and 16 neurons, respectively, with decreasing dropout rates. The final layer has a single neuron, signifying the output for 'accident severity.' The choice of these configurations aims to capture complex patterns in the data.

For Gradient Boosting Trees, it is an ensemble learning method that builds a predictive model in a stage-wise fashion by combining the predictions of multiple weak learners, often decision trees. Each tree corrects the errors of its predecessor, gradually improving the overall predictive accuracy. Gradient Boosting is widely used for regression and classification tasks due to its ability to handle complex relationships in data and resist overfitting.

V. 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. [22]

When we looked at how often different levels of accident severity happen, we noticed that fatal accidents are quite rare in the data. Most of the accidents fall into the category of slight accidents. Because of this, we decided to study fatal accidents and non-fatal accidents separately, focusing on the various environmental conditions that might be influencing them.[3]

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

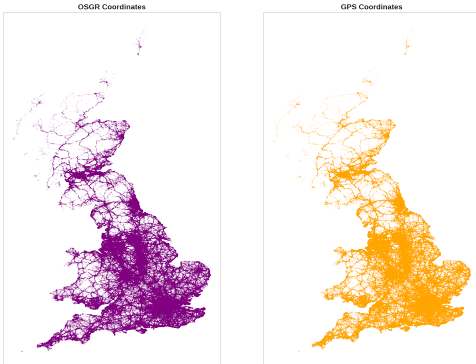

Fig. 1. Map of accidents by geographical location

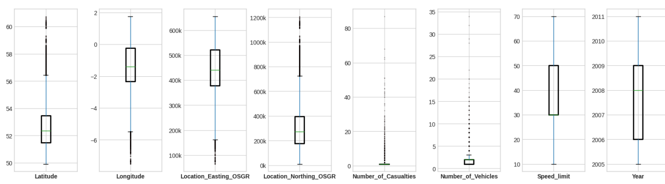

Fig. 2. Box Plot Shows no Outlier in the Data

appearing in the data from the fatal and non-fatal data. "Total_%" shows the percentage from the whole data.

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.

The majority of accidents, both serious and less severe, occur when the weather is clear and there's no strong wind. This is likely because clear weather is the most common condition throughout the year. Interestingly, there doesn't seem to be a specific weather condition that significantly influences how severe an accident might be.[4]

In light conditions, fatal accidents are more likely to occur when street lights are not present [5]

Interestingly, there have been some fatal accidents reported

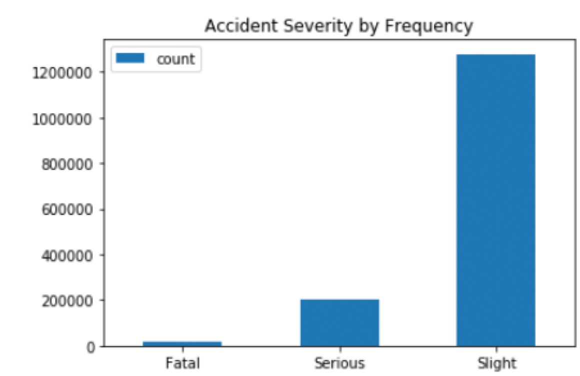

Fig. 3. Accident count divided into 3 severities

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

Fig. 4. Fatal and non-fatal accidents by weather conditions

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

Fig. 5. Fatal and non-fatal accidents by lighting condition

in areas with small speed limits.[7] Accidents are more likely to be fatal on a dual carriageway, and fatal accidents happen on roundabouts less often than non-fatal accidents

We created heatmaps to show the number of accidents on different days and months. We compared our maps to those in an article by Rakesh Nain. Even though the article used a smaller amount of data, the general trends we observed were similar for big data as well. From our maps, we noticed that Sunday has the fewest accidents, likely because it's a less busy day on the roads. In contrast, Friday has the most accidents among weekdays. As the first day of the weekend, there's more traffic and people driving back from parties, possibly in a less alert state, making the risk of accidents higher.

In October and November, we observe more accidents, and there are likely a few reasons behind this. First, the days get shorter, and there's less sunlight, making driving conditions more challenging. Second, as temperatures drop, and there's more rain, the roads can become wet and slippery. Additionally, summer tends to see more accidents, possibly due to increased traffic from people going on vacation trips. [8]

Looking at the overall accident graph, it's likely that slight accidents make up the majority of the data due to their higher frequency. However, when we examine each severity level separately, we notice some interesting trends. Fatal accidents appear to be more common on weekends, possibly because of increased alcohol consumption and a more relaxed attitude while driving. As we move to less severe accidents, like serious and slight, we see a shift toward weekdays and regular workdays. This suggests that on weekdays, people tend to be

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

Fig. 6. Fatal and non-fatal accidents by road surface conditions

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

Fig. 7. Fatal and non-fatal accidents by speed limit (mph)

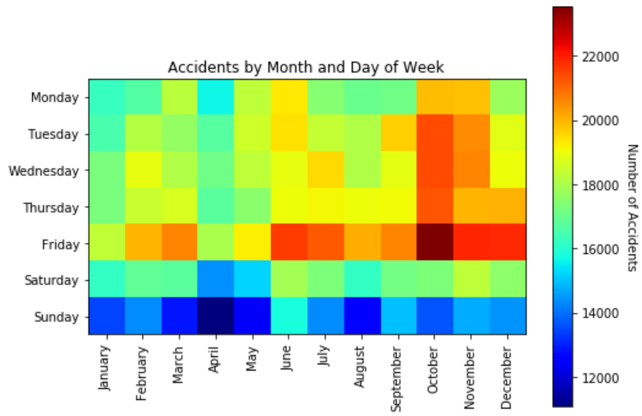

Fig. 8. Accident by Month and Day of week

more attentive, less influenced by substances, and although accidents still occur due to increased traffic, their severity is generally lower.[9]

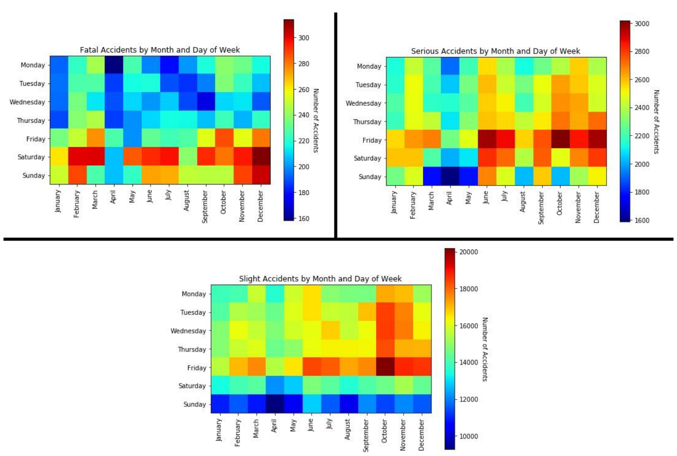

Fig. 9. Action Severity over months and day of week

Looking at the graph showing the time of day, we notice that most accidents happen during the usual morning commute times, and there's a broader period in the evening, possibly because people are driving to activities after work. On weekends, there's less traffic, but there are more accidents at night.[10]

When we divide the graph into the three severity categories, we notice some familiar patterns. The graph for slight accidents looks much like the original one. However, as accidents become more severe, they are more likely to occur at night and on weekends. Fatal accidents are less common, so there is more variability in the data, but the overall trend is still apparent. It's interesting to observe that slight accidents seem to be more concentrated in the mornings.[11]

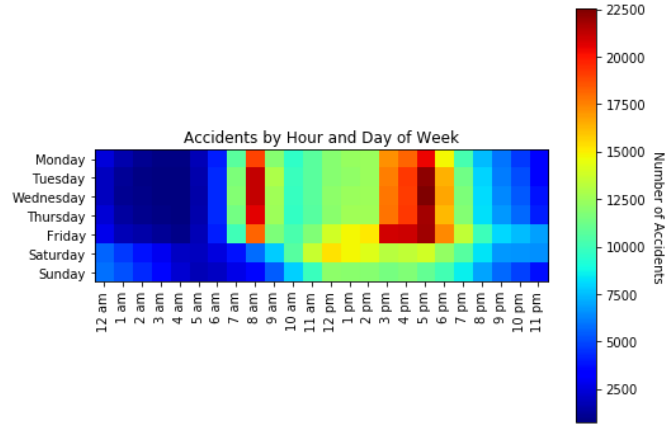

Fig. 10. Accident by hour and day of week

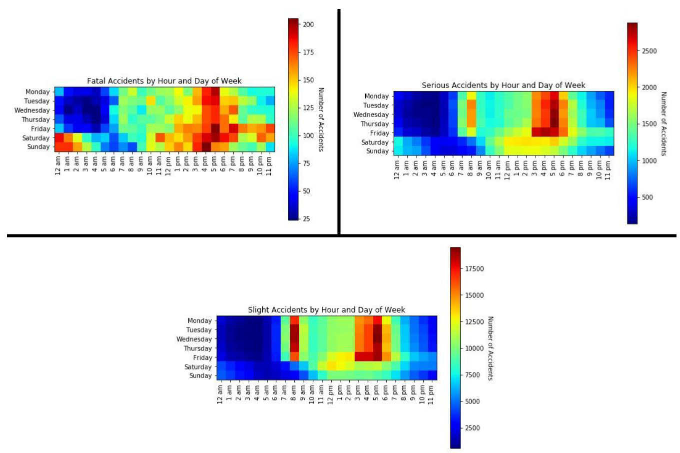

Fig. 11. Accident Severity by hour and day of week

VI. Prediction Analysis

A. Preprocessing

Sampling : In the preprocessing steps, we tackled the issue of imbalanced classes in our dataset. We tried different methods, like reducing the more common classes and increasing the less common ones, to make things more balanced. It's important to be careful with imbalanced datasets because models might seem really accurate by just guessing the most common class. We pointed out that it's crucial to evaluate how well the model performs for all classes, not just the majority one. 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

B. Feature Engineering

After completing the initial data preparation steps, we moved on to creating feature vectors using Pyspark's tools. This process involved converting text-based information into numbers that the machine could understand and then combining these values into vectors.

Despite dividing the dataset into training, testing, and validation sets, the performance of the model was disappointing. It struggled to differentiate between different classes, often assigning all samples to a single class, as shown in the confusion matrix [12]. We explored various options, including different feature selections, model configurations, and resampling methods, but the results remained consistently unsatisfactory.

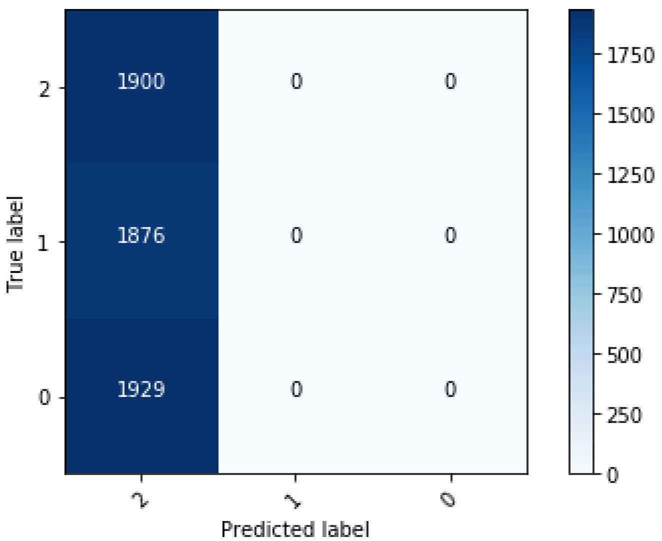

Fig. 12. Confusion Matrix where model falied to differentiate between classes

Recognizing the limitations of our initial approach, we shifted to building a custom pipeline for embedding text columns and normalizing column values. Unlike the previous method, this approach allowed us to assign values during the fitting process. The embedding process involved creating a dictionary that linked unique text values to integers based on their frequency. An accumulator was then constructed with multiple conditions to be applied to the column, changing its values accordingly.

After applying these changes to all text columns, we normalized all rows within a specified range, experimenting with both 0 to 1 and -1 to 1. It's important to note that the label column 'Accident_Severity' was not normalized. While we still utilized the pipeline from exercise 4 for constructing feature vectors, all columns were now treated as numeric columns.

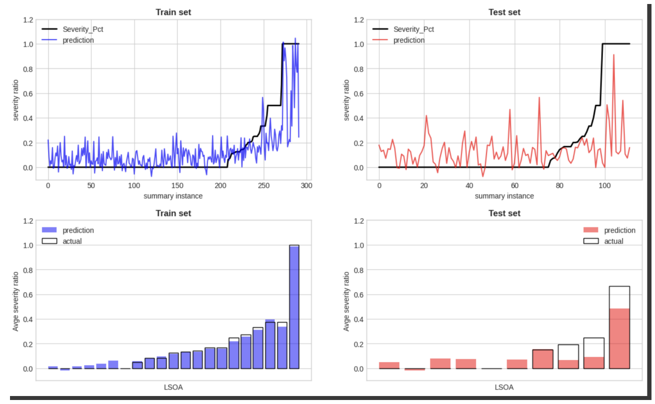

Fig. 13. Gradient Boosting Tree Training

C. Training Process

1) Gradient Bossting Tree (GBT): In the training process, we employed a Gradient Boosting Tree (GBT) model using PySpark's MLlib library. The categorical features were then transformed into index columns, and a VectorAssembler was utilized to combine these features into a single "features" column. The GBTRegressor, a powerful ensemble learning algorithm, was chosen as the regression model for predicting the response variable, which, in this case, is the severity of traffic accidents. The training pipeline, including the indexing, vectorization, and GBT model stages, was constructed and fitted on the training data. The model was trained with specific hyperparameters such as maxIter, maxDepth, and maxBins. After training, the model's performance was evaluated on both the training and testing sets using the Root Mean Squared Error (RMSE) as a metric to assess predictive accuracy. The training process aimed to create a robust GBT model capable of accurately predicting accident severity based on various input features.

2) Deep Neural Network: In the training process, we utilized a deep neural network (DNN) model with multiple hidden layers to predict the severity of traffic accidents based on various features. The architecture of the DNN model was configured with several dense layers, each employing the ReLU activation function. To enhance the model's ability to generalize, dropout layers were strategically inserted to prevent overfitting during training. The model was compiled using the Adam optimizer and mean squared error as the loss function. The training process involved iterating through epochs, and the learning curves were visualized to monitor the model's performance on both the training and validation sets. The training time and performance metrics, such as root mean squared error (RMSE), were displayed for evaluation. The custom LossHistory callback was implemented to store and track the training and validation losses during the training process. Additionally, an evaluation function was employed to calculate the RMSE on the provided dataset, providing insights into the predictive accuracy of the trained DNN model.

Hyperparameters : The hyperparameters for the DNN model include the number of neurons in each layer, dropout rates, the optimizer (Adam), and the loss function (mean squared error).

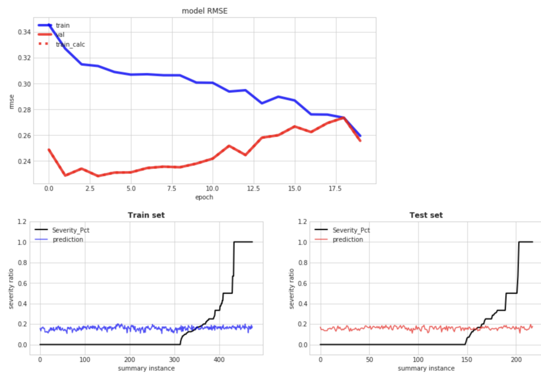

Fig. 14. Deep Neural Network Training

VII. Results

A. GBT Results

The Gradient Boosting Trees (GBT) model exhibited promising performance on the training data, achieving a Root Mean Squared Error (RMSE) of 0.22 and an R2 (explained variance ratio) of 0.26. However, its performance on the test data was less impressive, with an RMSE of 0.28 and a negative R2 of -0.04. The accuracy of the GBT model was 73%.

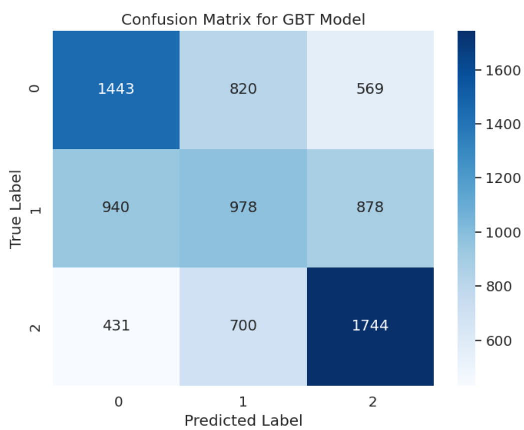

Fig. 15. Confusion Matrix for GBT

B. DNN Results

The Deep Neural Network (DNN) model, comprising both small and large networks, demonstrated better RMSE metrics compared to the GBT regressor. On the training data, the RMSE was 0.29, while on the test data, it was 0.27. The accuracy of the DNN model was 75

Despite the improved RMSE metrics, both the small and large deep networks struggled to capture the intricate dynamics of the dataset, merely identifying the best average outcome.

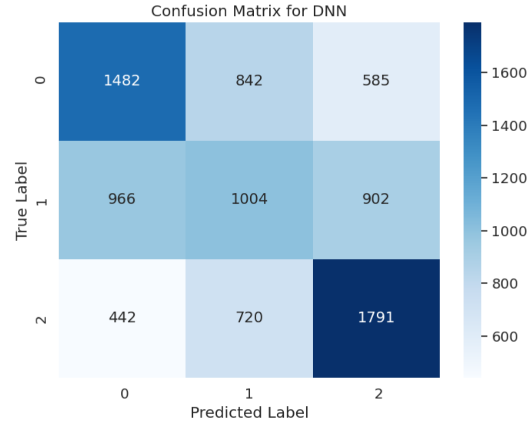

Fig. 16. Confusion Matrix for DNN

C. Accident Patterns Analysis

Analyzing accident patterns revealed intriguing insights. Most accidents occurred during autumn weekdays, particularly on Fridays. The morning and evening rush hours were identified as the most perilous times. Large cities, especially London, recorded the highest accident frequencies. Environmental conditions such as the absence of street lights, wet road surfaces, and high speed limits significantly increased the likelihood of severe accidents, especially on dual carriageways.

Contrary to expectations, weather conditions seemed to have a minimal impact on accident severity. Fatal accidents were more frequent on weekends and during late-night hours (12-1 am) compared to non-fatal accidents.

The best prediction model, the DNN, achieved a 75% accuracy with the three severity classes. Notably, the model excelled in classifying the most severe and mild cases but struggled with accurate predictions for the middle severity. Before balancing the data, the model tended to guess most, if not all, classes as the most frequent (mild) accidents. This highlights the importance of addressing data imbalance in predictive modeling. The findings suggest that data imbalance might be a common challenge in prediction models derived from this dataset.

D. Model Evaluation

In the evaluation phase, I conducted an assessment of the Gradient Boosting Trees (GBT) model. The primary focus was on predicting accident severity across the geographic landscape of the United Kingdom, specifically within each Output Area (LSOA). This evaluation involved considering various factors, including the date and time of the accident, as well as the prevailing driving conditions at the given locations.

VIII. Conclusion

In conclusion, our endeavors with this dataset indicate that building a model for accurately classifying different accident

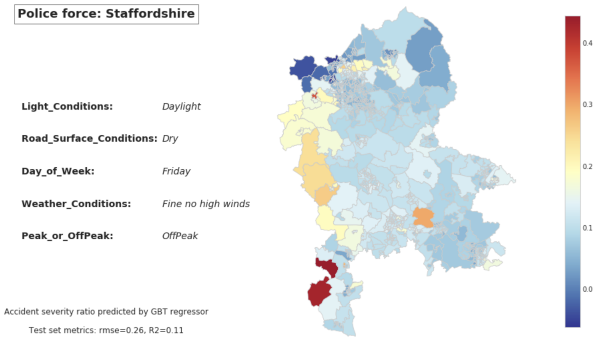

Fig. 17. Predict accident severity at each output area (LSOA) given datetime and driving conditions

severities is challenging. The existing data points don't exhibit sufficient correlation with accident severity to make reliable predictions. To enhance predictability, the dataset might benefit from more precise weather descriptions and additional data, such as vehicle speeds and driver information (e.g., sobriety). With these improvements, it would be worthwhile to reevaluate the classifiers used in this study and explore new classifiers for potential better performance.

From an educational standpoint, this project provided valuable insights into both the subject matter and the technical intricacies of working with PySpark and implementing various data processing steps. The topic of traffic safety remains pertinent, considering its significant impact on people worldwide. As the landscape evolves with the rise of autonomous vehicles and low-noise electric cars, ongoing research in traffic safety, accident prediction, and prevention becomes crucial. Understanding the relationships between causes and effects in the realm of traffic safety will continue to be relevant, addressing the ongoing challenges posed by serious accidents.

REFERENCES

[1] UK Collision and Statistics, year = 2015, url = https://www.brake.org.uk/get-involved/take-action/mybrake/knowledge-centre/uk-road-safety,.
[2] CDC Statistics. CDC Statistics, year = 2018, url = https://www.cdc.gov/injury/features/global-road-safety/index.html,.
[3] Kaggle Dataset, year = 2017, url = https://www.kaggle.com/datasets/tsiaras/uk-road-safety-accidents-and-vehicles/data,.
[4] Medium Article. Big data Analysis of Road Crash Data using PySpark with PySpark Tutorial., year = 2021, url = https://medium.com/rakesh-nain/big-data-analysis-of-road-crash-data-using-pyspark-with-pyspark-tutorial-c78ff2c35588,.
[5] Kun Li, Haocheng Xu, and Xiao Liu. Analysis and visualization of accidents severity based on lightgbm-tpe. *Chaos, Solitons & Fractals*, 157:111987, 2022.
[6] Meghan Bibb, Pablo Rivas, and Mahee Tayba. Predicting traffic accident severity with deep neural networks.
[7] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael J. Franklin, Reza Zadeh, Matei Zaharia, and Ameet Talwalkar. Mllib: Machine learning in apache spark, 2015.
[8] Zhiyuan He, Danchen Lin, Thomas Lau, and Mike Wu. Gradient boosting machine: A survey, 2019.

APPENDIX

```
root
 |-- Accident_Index: string (nullable = true)
 |-- 1st_Road_Class: string (nullable = true)
 |-- 1st_Road_Number: string (nullable = true)
 |-- 2nd_Road_Class: string (nullable = true)
 |-- 2nd_Road_Number: string (nullable = true)
 |-- Accident_Severity: string (nullable = true)
 |-- Carriageway_Hazards: string (nullable = true)
 |-- Date: string (nullable = true)
 |-- Day_of_Week: string (nullable = true)
 |-- Did_Police_Officer_Attend_Scene_of_Accident: string (nullable = true)
 |-- Junction_Control: string (nullable = true)
 |-- Junction_Detail: string (nullable = true)
 |-- Latitude: double (nullable = true)
 |-- Light_Conditions: string (nullable = true)
 |-- Local_Authority_District: string (nullable = true)
 |-- Local_Authority_Highway: string (nullable = true)
 |-- Location_Easting_OSGR: integer (nullable = true)
 |-- Location_Northing_OSGR: integer (nullable = true)
 |-- Longitude: double (nullable = true)
 |-- LSOA_of_Accident_Location: string (nullable = true)
 |-- Number_of_Casualties: integer (nullable = true)
 |-- Number_of_Vehicles: integer (nullable = true)
 |-- Pedestrian_Crossing_Human_Control: string (nullable = true)
 |-- Pedestrian_Crossing_Physical_Facilities: string (nullable = true)
 |-- Police_Force: string (nullable = true)
 |-- Road_Surface_Conditions: string (nullable = true)
 |-- Road_Type: string (nullable = true)
 |-- Special_Conditions_at_Site: string (nullable = true)
 |-- Speed_limit: integer (nullable = true)
 |-- Time: string (nullable = true)
 |-- Urban_or_Rural_Area: string (nullable = true)
 |-- Weather_Conditions: string (nullable = true)
 |-- Year: integer (nullable = true)
 |-- InScotland: string (nullable = true)

number of records: 1917274
```

Fig. 18. Road Types and Class

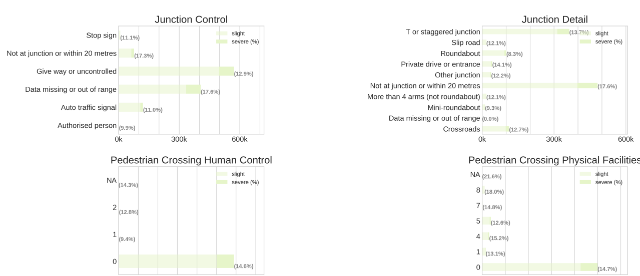

Fig. 19. The majority of accidents occur away from a junction Also the presence of a junction do not increase accident severity

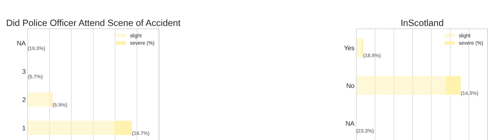

Fig. 20. Did Police Officer Attend Scene of Accident

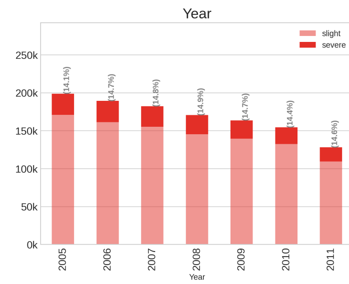

Fig. 21. Accident Per Year

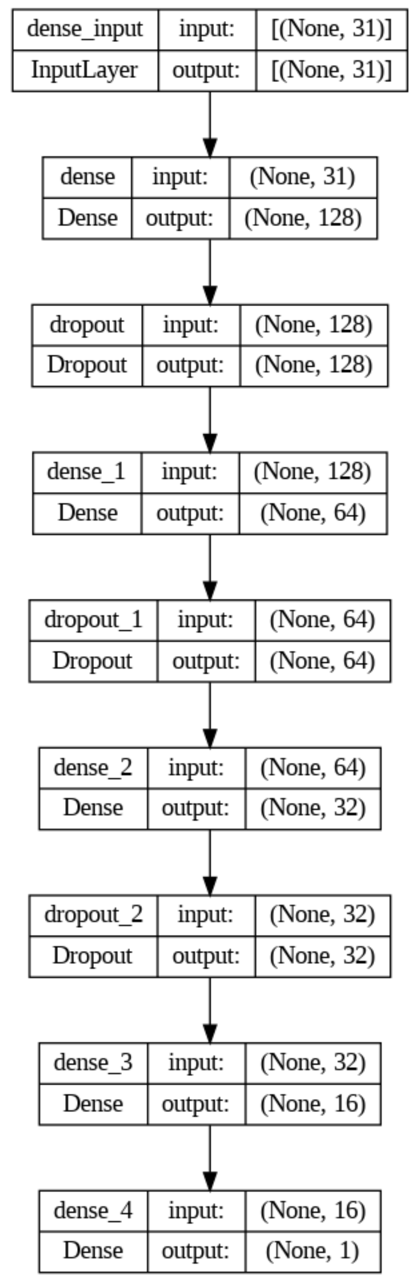

Fig. 22. Deep Neural Network Architecture