Introduction:

The study delves into 2020 road traffic accidents in the United Kingdom with the goal of improving road safety. It examines the 2020 accident dataset provided by the UK government to reveal accident trends, risk factors, and patterns, guiding targeted interventions, and policies. The dataset is obtained by police at the place of an incident, or sometimes obtained online or reported at a police station by someone (Gov.UK, n.d). The study also investigates vehicle types, particularly motorcycles, as well as peak accident hours and accident distribution in specific regions using clustering and outlier identification. Finally, a classification algorithm estimates fatal injuries in car accidents to inform and improve road safety measures.

Methodology:

The approach used to address the issues with road safety incorporates several crucial steps:

- Data Preprocessing: To handle missing values, standardise features, and transform categorical variables into numerical representations, the 2020 traffic accident data is pre-processed.
- Time Analysis: Significant hours and days of the week with high accident incidence are identified using data visualisation tools to prioritise targeted safety measures during peak accident hours.
- Motorbike Analysis: To analyse accident trends for different motorbike categories, separate analyses are performed for motorbike accidents depending on engine capacity (125cc or less, 125cc to 500cc, and 500cc and more).
- Pedestrian Analysis: The danger of pedestrians being involved in accidents is analysed by reviewing main hours and days when pedestrian accidents are more common.
- Application of Apriori algorithm, clustering, outlier detections and training of predictive classification model.

Analysis:

The analysis of the 2020 accident data will be done following some prompt questions or guidelines.

Significant hours of the day on which accident occurs:

	time	number_of_accidents
1018	17:00	862
958	16:00	785
898	15:00	774
1048	17:30	746
1078	18:00	739
838	14:00	699
988	16:30	697
928	15:30	697
1108	18:30	629
778	13:00	605

Table 1.0 – Top 10 accident occurrence in hours

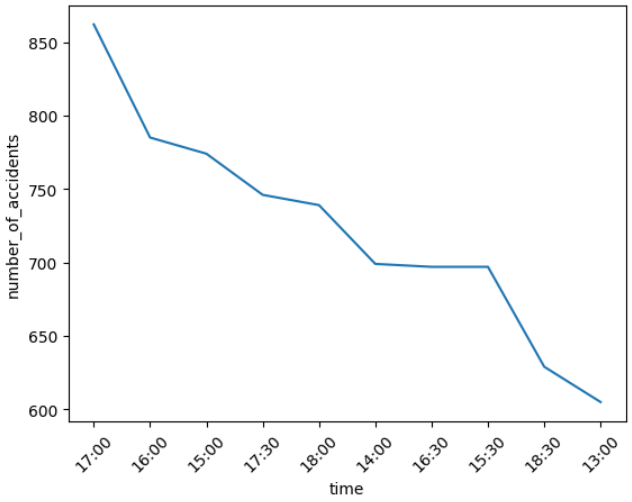

Figure 1.0 – representation of accident occurrences in hours

Accident time analysis entailed categorising data by time, counting accidents, and determining peak hours. Table 1 provides the top ten periods, and Figure 1.0 depicts this visually. Notably, significant accidents occurred between 16:00 and 17:00, with the peak being around 17:00 (862 events).

Significant days of the week on which accident occurs:

```
day_week
Friday       14889
Thursday     14056
Wednesday    13564
Tuesday      13267
Monday       12772
Saturday     12336
Sunday       10315
Name: accident_index, dtype: int64
```

Table 2.0 -Top accident occurrence days

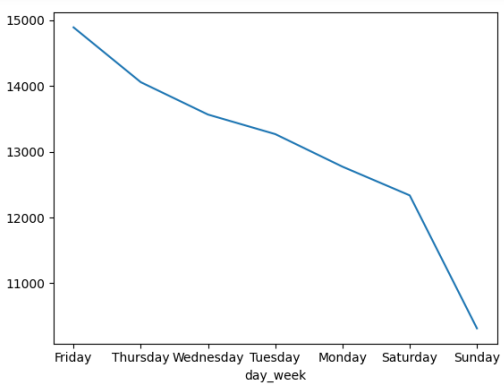

Figure 2.0 – representation of accident occurrence in days

Significant accident days were determined in a similar manner. Table 2.0 shows high accident rates on Thursdays and Fridays, with Friday leading the way with 14,889 accidents. The line graph supported these data, showing Thursdays and Fridays as important accident days.

Significant motorbikes accidents on hours of the day:

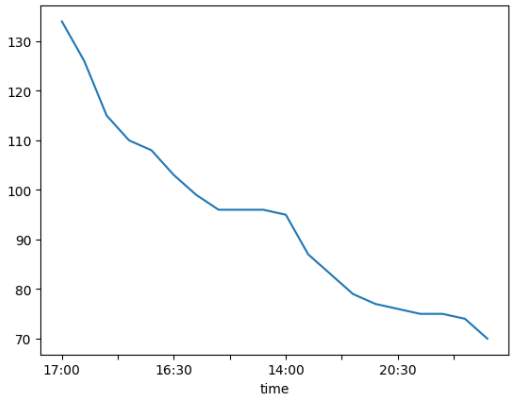

Figure 3.0 – Motorbikes accidents per hour of the day

I found important motorbike accidents that occurred throughout the day by choosing 3 classes of 3, 4, and 5 from accident form. I also combined accident and vehicle data. This produced **17.00** as the key hour for motorbike accidents. This is visualized in Figure 3.0.

Significant motorbikes accidents on weekdays:

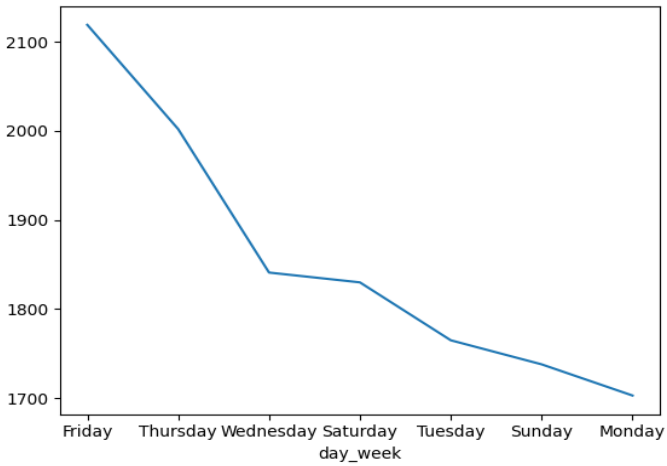

Figure 4.0 – Motorbikes accidents per days of the week

The data was aggregated by the day of the week, counting the number of accidents per day, and plotting it. The analysis revealed a high number of motorbike accidents on **Fridays** and highlighted that most accidents were caused by motorbikes of various engine capacities.

Significant Hours of Accident for pedestrians of the day:

```
time
15:30    188
15:00    164
16:00    153
18:00    152
17:00    150
         ...
03:46      1
02:11      1
19:34      1
06:53      1
01:43      1
Name: accident_index, Length: 1264, dtype: int64
```

Table 3.0 - Pedestrians accidents per hour

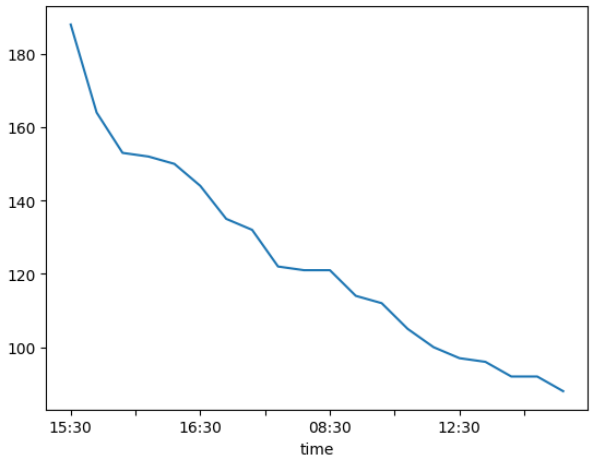

Figure 5.0 – Pedestrians accident per hour

Data in Table 3.0 and Figure 5.0 reveal that pedestrian accidents peaked between **15:30 and 16:00** in 2020, with **15:30** having the most events (**188**).

Significant Day of Accident for pedestrians of the week:

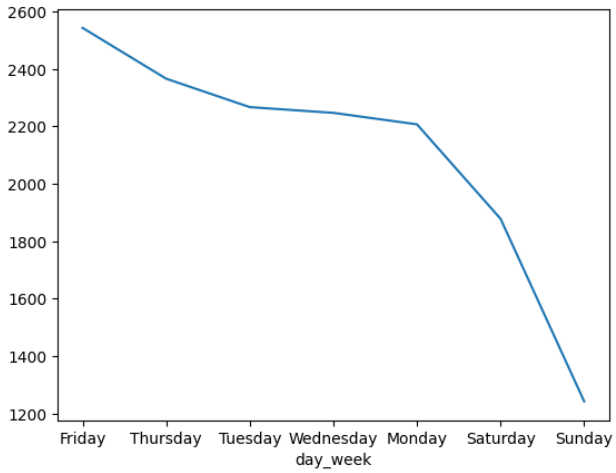

Figure 6.0 – Pedestrians accident occurrence of the week

Figure 6.0 showed accident with the highest spike of accident occurrence for the pedestrians. **Friday** has shown to be the days of the week that most accidents occurred for both motorbikes and **pedestrians**. Further insights and suggestions need to focus on the hours of 15.30 and Fridays for pedestrians.

Using the Apriori algorithm, explore the impact of selected variables on accident severity:

To explore the impact of the selected variable on accident_severity, I used Apriori algorithm which is a technique for determining underlying relationships between various variables (Chonyy, 2020). The Apriori method assessed the effects of variables' relationships with accident severity, using data from the accident, vehicle, and casualty tables. The top five variables with the strongest associations with accident severity were extracted and presented as association rules, along with additional explanations of their impact. Due to space constraints, only the first two columns of association rules are covered.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(number_of_vehicles)	(accident_severity)	0.867285	0.980806	0.851952	0.982320	1.001544	0.001313	1.085646	0.011615
43	(speed_limit, number_of_vehicles)	(accident_severity)	0.867267	0.980806	0.851934	0.982320	1.001543	0.001313	1.085623	0.011610
57	(age_of_casualty, number_of_vehicles)	(accident_severity)	0.846726	0.980806	0.831492	0.982009	1.001226	0.001018	1.066854	0.007991
263	(speed_limit, number_of_vehicles, age_of_casua...	(accident_severity)	0.846708	0.980806	0.831474	0.982009	1.001226	0.001018	1.066831	0.007988
52	(vehicle_type, number_of_vehicles)	(accident_severity)	0.792860	0.980806	0.778157	0.981456	1.000663	0.000515	1.035047	0.003197

Table 4.0 Apriori association of variables with accident severity

Antecedents: number_of_vehicles and Consequents: accident_severity

With 98.2% confidence, the association rule reveals that knowing the number of vehicles are part of an accident has a high predictive link with accident_severity. The conviction rate of 1.086 indicates that number_of_vehicles increase the risk of greater severity by 8.6%. Despite their frequent co-occurrence in 85.2% of data, the lift of approximately one indicates a minor relationship between the number of cars and greater severity. Overall, the number of cars is useful for predicting severity but is not strongly associated to it.

Antecedents: speed_limit, number_of_vehicles and Consequents: accident_severity

This rule suggests that the speed limit and the number of vehicles involved can be used to forecast accident severity with 98.2% accuracy, even though these variables are not substantially connected with creating increased severity (lift of 1.001543). Based on a conviction rate of 1.085623, the antecedents raise the severity probability by 8.6%. Their co-occurrence with accident severity, on the other hand, is highly prevalent (85.2% support). Overall, the antecedents are highly predictive but have a modest connection with accident severity.

Identify accidents distribution in our region: Kingston upon Hull, Humberside, and the East Riding of Yorkshire

To identify accident distribution in our various regions, I used Clustering method which can discover hidden relationships between data points based on their common properties (Datamites, 2022). Using data from accident, vehicle, Lsoa, and casualty tables, clustering was applied to three regions (Kingston upon Hull, East Riding of Yorkshire, and Humberside). The KMeans elbow curve method calculated the optimal number of clusters, settling on three for improved visualisation. For clarity and understanding, findings will be presented area by region.

Kingston Upon Hull:

	labels_Kmeans	Number of Accidents
0	0	334
1	1	508
2	2	437

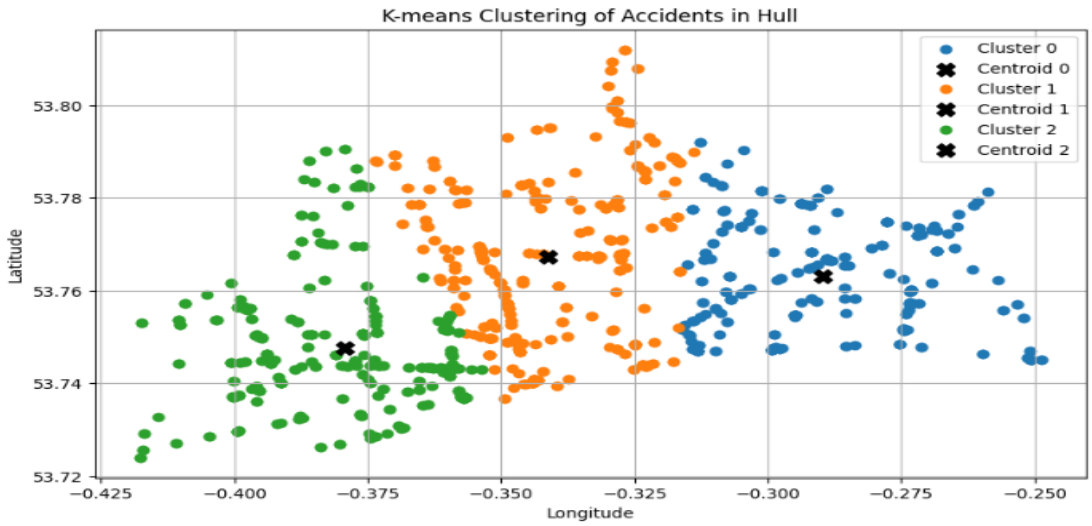

Figure 7.0 – KMeans Clustering of accidents in Hull

Figure 8.0 – Map showing the accident distribution in Hull.

In Hull, three clusters (0,1 and 2) were discovered, with accident distributions of 334, 508, and 437 respectively, for a total of 1279 incidents. Figure 8.0 depicts the accident distribution. Refer to the accompanying Jupyter notebook for comprehensive cluster information and map visualisations.

East Riding of Yorkshire:

	labels_Kmeans	Number of Accidents
0	0	321
1	1	524
2	2	343

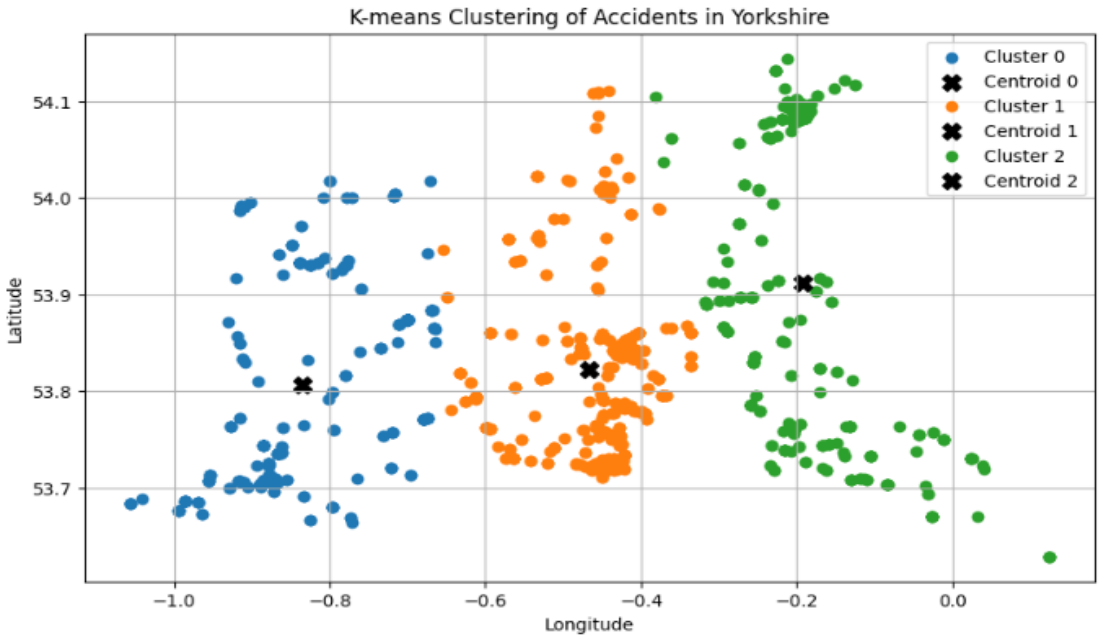

Figure 9.0 – accident distribution clustering of East Riding of Yorkshire

Figure 10.0 Map representation of accident distribution for East Riding of Yorkshire

Three clusters (0,1 and 3) were discovered in East Riding of Yorkshire, with accident distributions of 321, 524, and 343 totalling 1188 accidents. The accident distribution is depicted on the map in Figure 10.0.

Humberside:

	labels_Kmeans	Number of Accidents
0	0	699
1	1	580
2	2	239

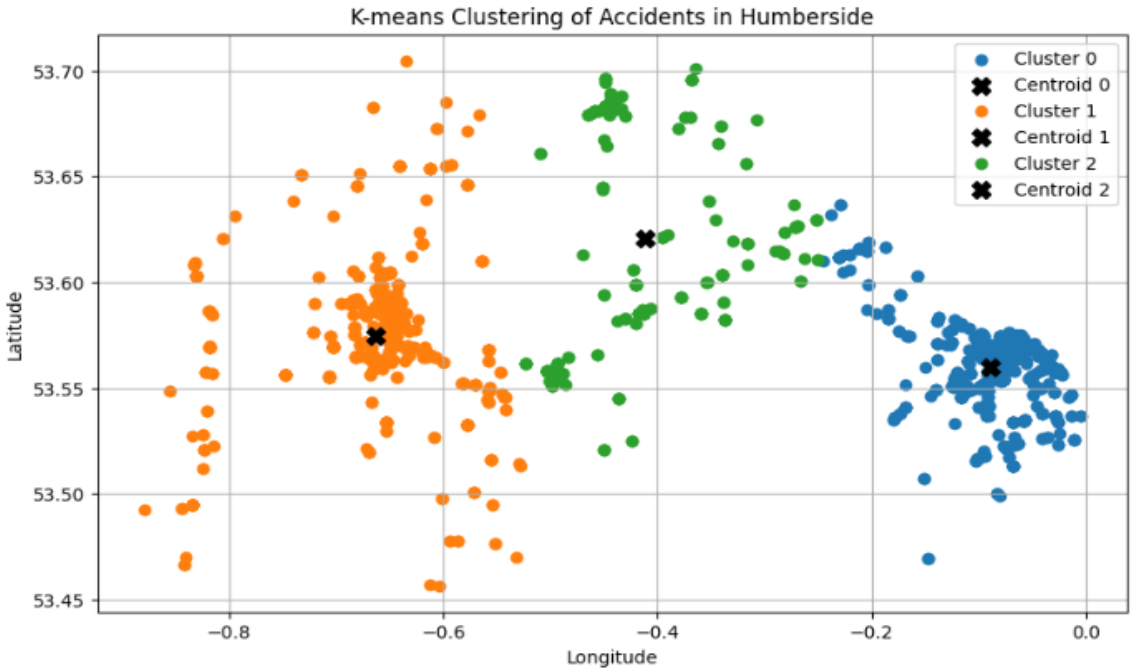

Figure 11.0 accident distribution of Humberside

Figure 12.0 map distribution of Humberside

In Humberside, three clusters (0,1 and 2) were detected, with accident distributions of 699, 500, and 239 respectively, for a total of 1518 incidents. Figure 12.0 depicts the accident distribution map.

To summarize the accident distribution in the 3 regions, the map below shows the accident distribution in the 3 regions. Refer to the accompanying Jupyter notebook for comprehensive cluster information and map visualizations.

Figure 13.0 map distribution of Hull, Yorkshire, and Humberside

Figure 13.0 depicts the accident distribution in Hull, Yorkshire, and Humberside for 2020. Hull had 1279 accidents, Yorkshire had 1188, and Humberside had 1518, for a total of 3985 events throughout the three regions.

Outlier detection methods:

To detect unusual entries in my dataset, I applied Outlier detection which is a type of data analysis that identifies anomalous findings in a dataset (Bush, 2020). I followed the same pattern used in my clustering which is region by region. Two outliers' detection methods which are Local outlier Factors and IsolationForest were used for Hull region while IsolationForest was used for Yorkshire and Humberside. The details of the outliers are as shown below:

Hull:

Outlier Detection using Local Outlier Factors (LOF)

	longitude	latitude
83567	-0.321773	53.773447
83568	-0.321773	53.773447
83591	-0.328032	53.777858
83592	-0.328032	53.777858
83958	-0.344794	53.768032
...	...	...
87381	-0.365360	53.740502
87382	-0.365360	53.740502
87415	-0.410842	53.726991
87416	-0.410842	53.726991
87425	-0.261624	53.764536

128 rows × 2 columns

Table 5.0 – LOF outliers

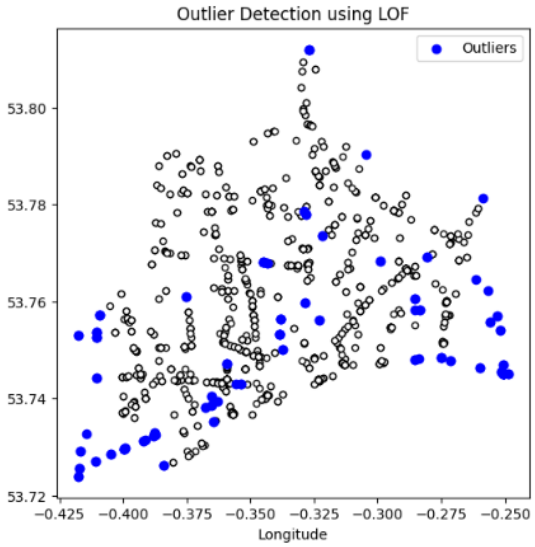

Figure 14.0 outlier detection using LOF.

Table 5.0 shows the location of the outliers detected in the dataset of Hull. The blue markers in figure 14.0 are the representation of outliers identified in the Hull Region. A total of 128 outliers were detected.

Outlier Detection Using IsolationForest (ISF):

	longitude	latitude
166	-0.324394	53.807867
167	-0.324394	53.807867
168	-0.328268	53.800963
169	-0.328268	53.800963
184	-0.250524	53.745048
...	...	...
1199	-0.253136	53.757024
1200	-0.252017	53.754032
1201	-0.252017	53.754032
1257	-0.410842	53.726991
1258	-0.410842	53.726991

63 rows × 2 columns

Table 6.0 ISF Outliers

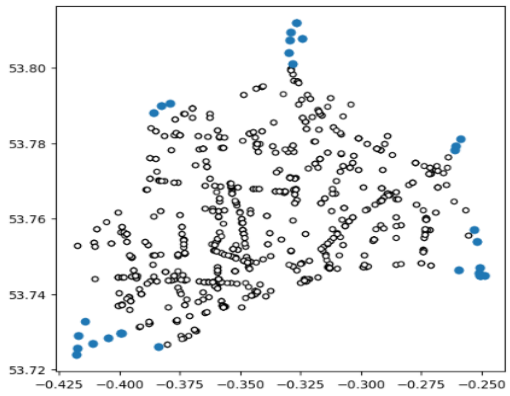

Figure 15.0 Outlier detection using IsolationForest.

The Isolation-Forest showed a total 63 outliers in the dataset of Hull as shown in the table 6.0 and visualized in figure 15.0.

Yorkshire region:

Outlier Detection using Local Outlier Factors (LOF)

	longitude	latitude
83671	-0.361794	53.864510
83672	-0.361794	53.864510
83673	-0.368461	53.864753
83674	-0.368461	53.864753
83971	-0.832815	53.690712
...	...	...
87346	-0.139122	54.121760
87428	-1.055770	53.682905
87429	-1.055770	53.682905
87430	-1.055770	53.682905
87431	-1.055770	53.682905

119 rows × 2 columns

Table 7.0 LOF Outliers

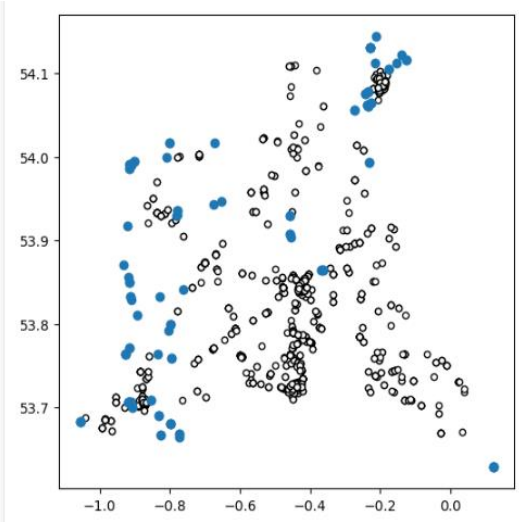

Figure 16.0 Outlier detection using LOF.

Table 7.0 shows the location of the outliers detected in the dataset of Yorkshire. The blue markers in figure 16.0 are the representation of outliers identified in the Hull Region. A total of 119 outliers were detected.

Outlier Detection using Local IsolationForest (ISF)

```
First 5 rows:
     longitude   latitude
107   0.023263  53.730483
108   0.023263  53.730483
109   0.023263  53.730483
110   0.023263  53.730483
111   0.023263  53.730483

Last 5 rows:
      longitude   latitude
1167  -1.055770  53.682905
1168  -1.055770  53.682905
1169  -1.055770  53.682905
1170  -1.055770  53.682905
1173  -1.040787  53.687955

Total number of rows: 56
Total number of columns: 2
```

Table 8.0 – IsolationForest outlier detection

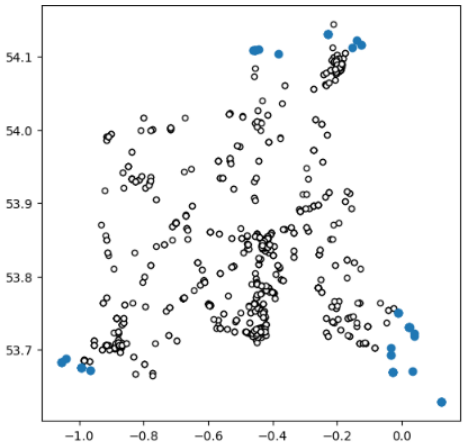

Figure 17.0 ISF outlier detection

A total of 56 outliers were detected for Yorkshire region as shown on table 8.0. Figure 17.0 is visual representation of the outliers.

Humberside will be evaluated using Isolation-Forest. This is because it has the advantage of computationally efficient, with a low memory demand, linear time complexity, and a low constant. As a result, it handles big data sets well. It does not presume normal distribution and can detect outliers at multiple dimensions (Yoon, 2022).

Humberside:

	longitude	latitude
44	-0.806313	53.620510
45	-0.806313	53.620510
186	-0.364669	53.700887
187	-0.367902	53.695609
188	-0.367902	53.695609
...	...	...
1487	-0.577565	53.671189
1514	-0.823599	53.520707
1515	-0.823599	53.520707
1516	-0.823599	53.520707
1517	-0.823599	53.520707

75 rows × 2 columns

Table 9.0 – IsolationForest outlier detection

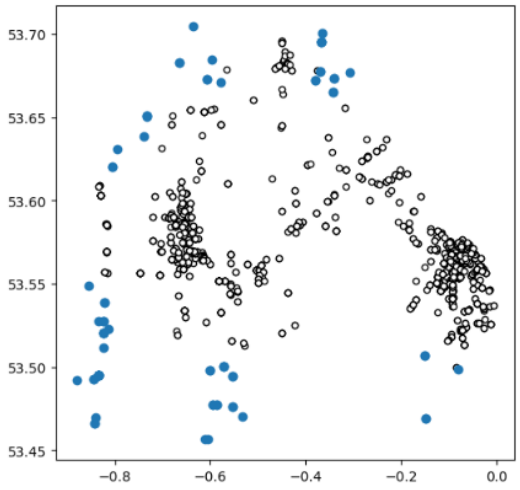

Figure 18.0 ISF outlier detection

A total of 75 outliers were detected as shown on table 9.0.

Outliers won't be dropped; they will be cleaned up during data cleaning process and used together with other datapoints in my dataset to build predictive classification model.

Predictions:

This involves building a model that predicts fatal injuries sustained in road traffic accidents. The building of the model involved 6 steps which are Data cleaning, Data preprocessing, Model training, Data balancing Model retraining and Evaluation metrics.

Data Cleaning: The data was explored where lots of negative values which are unusual in the dataset were discorvered. Most of these unusual values also comes as outliers and in the negative values. These steps were used to clean up the dataset before building the model.

- Use NAN to replace all the negative values -1
- Fill the NAN values with their medians.

Duplicates were also checked which reviewed a total of 52 rows were duplicated. The duplicated rows were dropped.

Data Preprocessing: The accident severity class was visualized to understand the target dataset been used which showed that the data is imbalance.

```
3    3091
2     768
1      74
Name: accident_severity, dtype: int64
```

Table 8.0 – unbalanced data

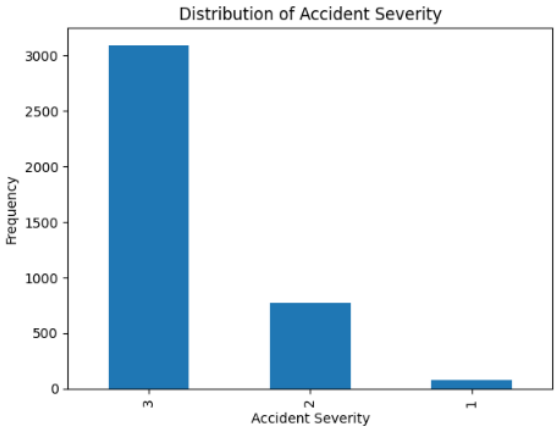

Figure 19.0 Data imbalance visualization

Table 8.0 showed that the accident severity is not balanced with class 1 which is Fatal showing a dataset of 74, class 2 showing 768 and class 3 showing 3091. Figure 19.0 visualized the data imbalance.

The next step in the Data preprocessing is the feature selection where the non-required columns like police force and region were dropped. Data splitting was also implemented in this section.

Model Building and Evaluation: The model was first trained with Gradient-boosting algorithm without the data been balanced. The classification report as shown below was used to analyse the result.

```
Classification Report:
              precision    recall  f1-score   support

       Fatal       0.67      0.36      0.47        22
     Serious       0.76      0.31      0.44       231
      Slight       0.84      0.98      0.90       927

    accuracy                           0.83      1180
   macro avg       0.76      0.55      0.60      1180
weighted avg       0.82      0.83      0.80      1180
```

Table 9.0 Classification report of the first model without data balancing

Classification report showed disparities with ‘Fatal’ class having precision of 67%, f1-score 47%, recall 36%, while Slight class had precision of 84%, recall 98%, f1-score 90%. Imbalance data made "Fatal" class results.

Data balancing: SMOTE algorithm was used to balance the data and that produced the result shown in the table 10.0 below and visualized as shown Figure 20.0.

```
3    3091
2    3091
1    3091
Name: accident_severity, dtype: int64
```

Table 10.0 Balanced data.

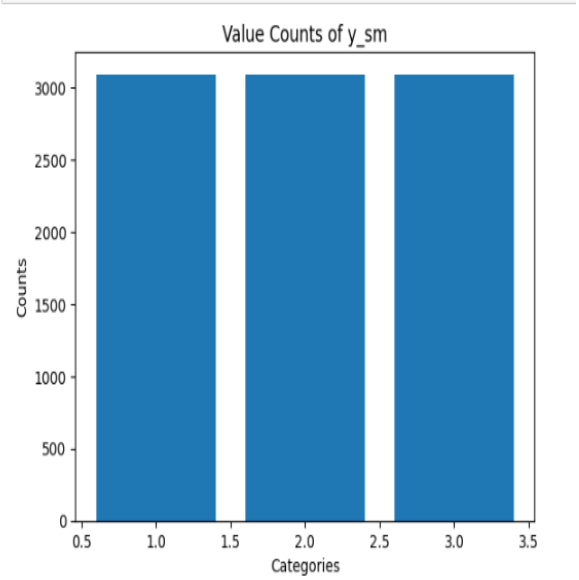

Figure 20.0 – Visualization of the balanced data

Rebuilding the Model with the balanced Data: The model was retrained with the balanced data with evaluation carried out to determine the performance. Gradient boosting algorithm was also used to train the model. Find below the classification report for the rebuilt model.

```
Classification Report:
              precision    recall  f1-score   support

       Fatal       0.92      0.94      0.93       955
     Serious       0.88      0.72      0.79       918
      Slight       0.79      0.91      0.85       909

    accuracy                           0.86      2782
   macro avg       0.86      0.86      0.86      2782
weighted avg       0.86      0.86      0.86      2782
```

Table 11.0 – Classification report of the rebuilt Model

The rebuilt model achieved accuracy of 86%, 92% precision, 94% recall, and a 93% f1-score. It excelled in predicting fatal injuries, which will aid road safety measures.

Feature of Importance: As shown in Figure 20.0, variable importance analysis confirmed that all specified variables contributed to model training and predictions. Because no variables were found to have an insignificant contribution, they were all kept.

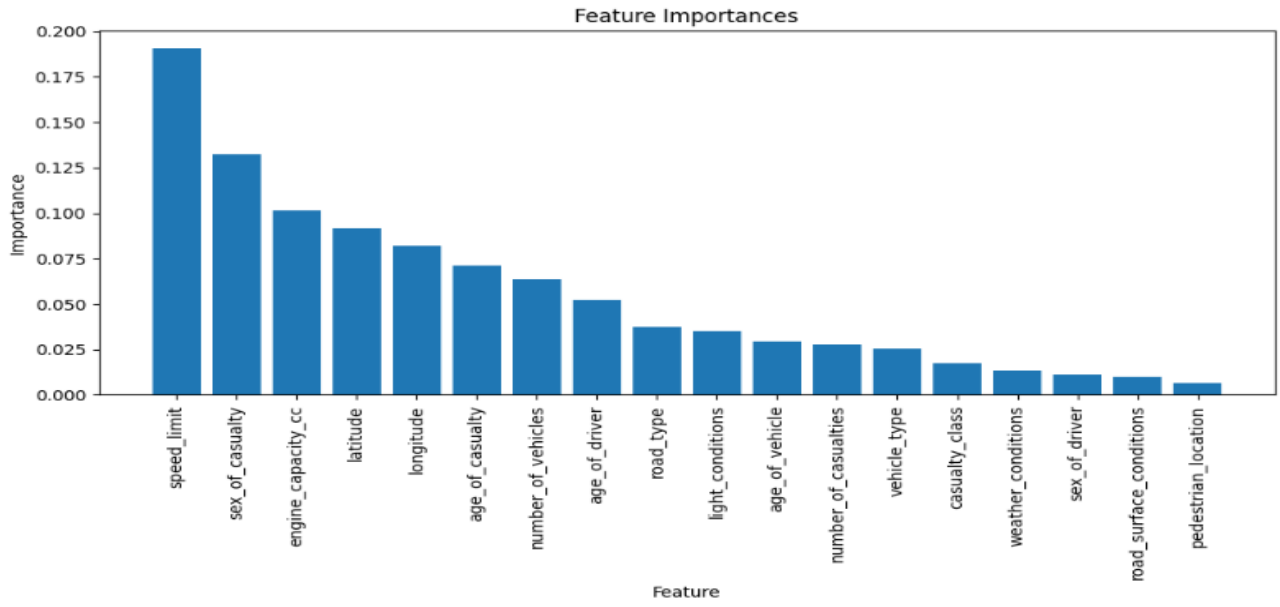

Figure 21.0 Feature of importance.

On confirmation of the good performance of the model and the contributions of the features, the rebuilt model was saved as **accident_prediction.pkl**

Recommendations:

- Increase police patrols and enforcement on Fridays and between 4-5 p.m., when accident rates are highest. Target speeding and intoxicated driving.
- Implement motorcycle safety campaigns on Fridays and at 5 p.m., concentrating on helmet use and safe driving.
- Improve pedestrian safety measures such as crosswalks and signals near schools, super malls & stores, trains, and bus parks where the risk is greatest about 3:30 p.m.
- Use the accident severity prediction model to identify high-risk regions and conditions. Target investments on safety countermeasures.
- Use clustering analysis to gain a regional understanding of accident patterns. Create localised preventative measures.
- Use outlier analysis findings to address data issues and uncommon high-risk accident types.

Reference list:

BUSH, T. (2020) *Outlier Analysis: Definition, Techniques, How-To, and More*. pestleanalysis.com. Available online: https://pestleanalysis.com/outlier-analysis/.

Chonyy (2020) *Apriori: Association Rule Mining In-depth Explanation and Python Implementation*. Medium. Available online: https://towardsdatascience.com/apriori-association-rule-mining-explanation-and-python-implementation-290b42afdfc6.

Datamites, D.S.T. (2022) *What is Clustering in Machine Learning - Importance & Types*. Datamites Global Institute of Data Science. Available online: https://datamites.com/blog/clustering-in-machine-learning/#:~:text=Clustering%20is%20a%20widely%20used [Accessed 8 Aug. 2023].

GOV.UK (n.d.) *Reported Road casualties in Great Britain: notes, definitions, symbols and conventions*. GOV.UK. Available online: https://www.gov.uk/government/publications/road-accidents-and-safety-statistics-notes-and-definitions/reported-road-casualties-in-great-britain-notes-definitions-symbols-and-conventions.

Yoon, Y. (2022) *Isolation Forest Anomaly Detection — Identify Outliers*. Medium. Available online: https://medium.com/@y.s.yoon/isolation-forest-anomaly-detection-identify-outliers-101123a9ff63 [Accessed 7 Aug. 2023].