INTRODUCTION

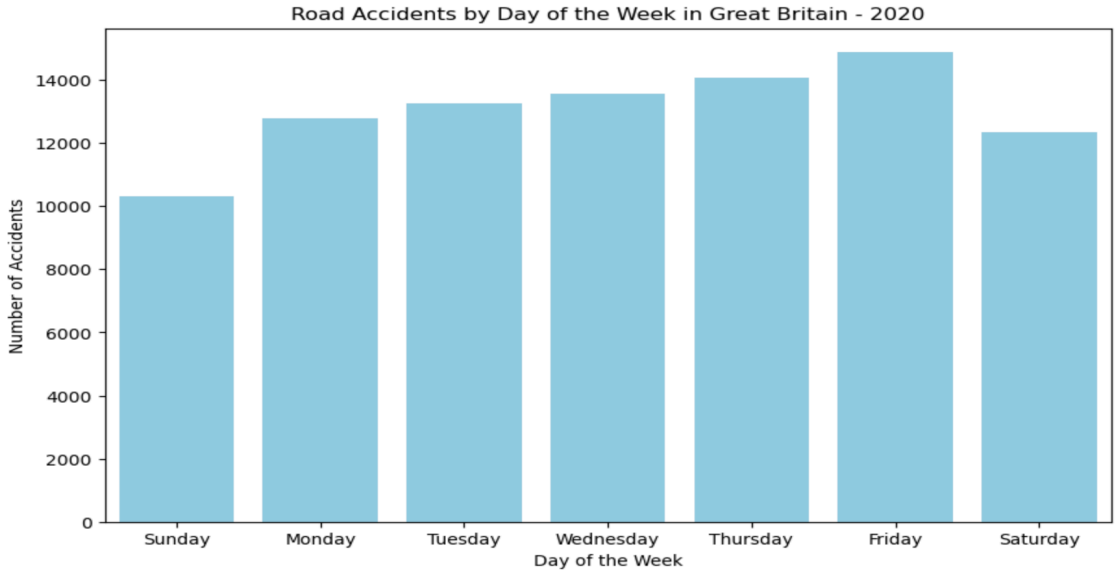
This report presents an analysis of road accidents that occurred in Great Britain during year 2020. The database is an SQL database consisting of tables containing information about accidents, vehicles, casualties, and LSOA for various years. The analysis focuses on patterns and trends of accidents that occurred. A classification model will be developed with the purpose of predicting fatal accidents to aid effective road safety strategies. Lastly, informed recommendations will be provided based on the insights garnered from the comprehensive analysis and observed trends.

ANALYSIS

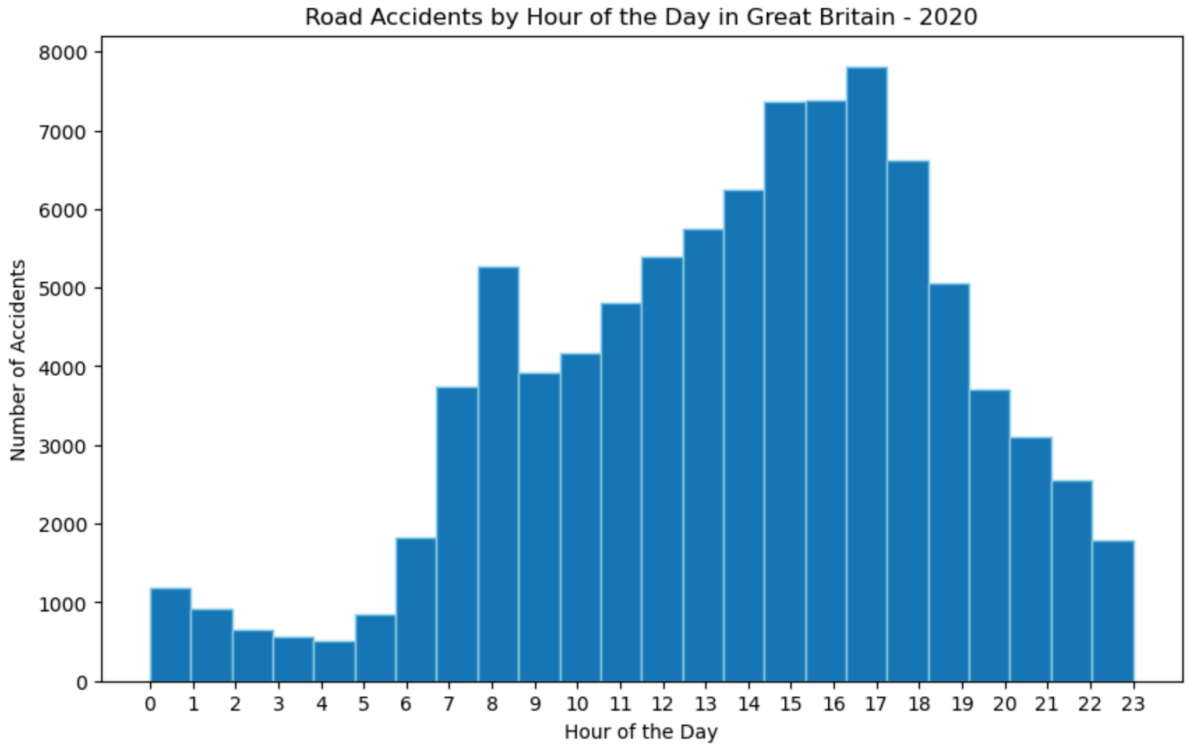
This will be carried out by looking for patterns in accident that occurred based on different conditions like time, place and vehicle involved. All the tables in the SQL database will be converted to Dataframe and filtered on just year 2020.

SIGNIFICANT HOURS OF THE DAY, AND DAYS OF THE WEEK, ON WHICH ACCIDENTS OCCUR

This approach involved creating a new dataframe named 'accident_dayTime' by extracting the 'time' and 'day of week' columns from 'accident_df'. The data contained days of the week represented with numbers, a mapping was applied to numbers with their corresponding days, beginning with Sunday. Hours was extracted from the time the accident was record. These newly derived features, 'hour_of_accident' and 'day_of_weekname', were subsequently appended as columns to the 'accident_dayTime' dataframe. The resulting dataset was then used for generating a histogram plots, visualizing the distribution of these two features as illustrated below.



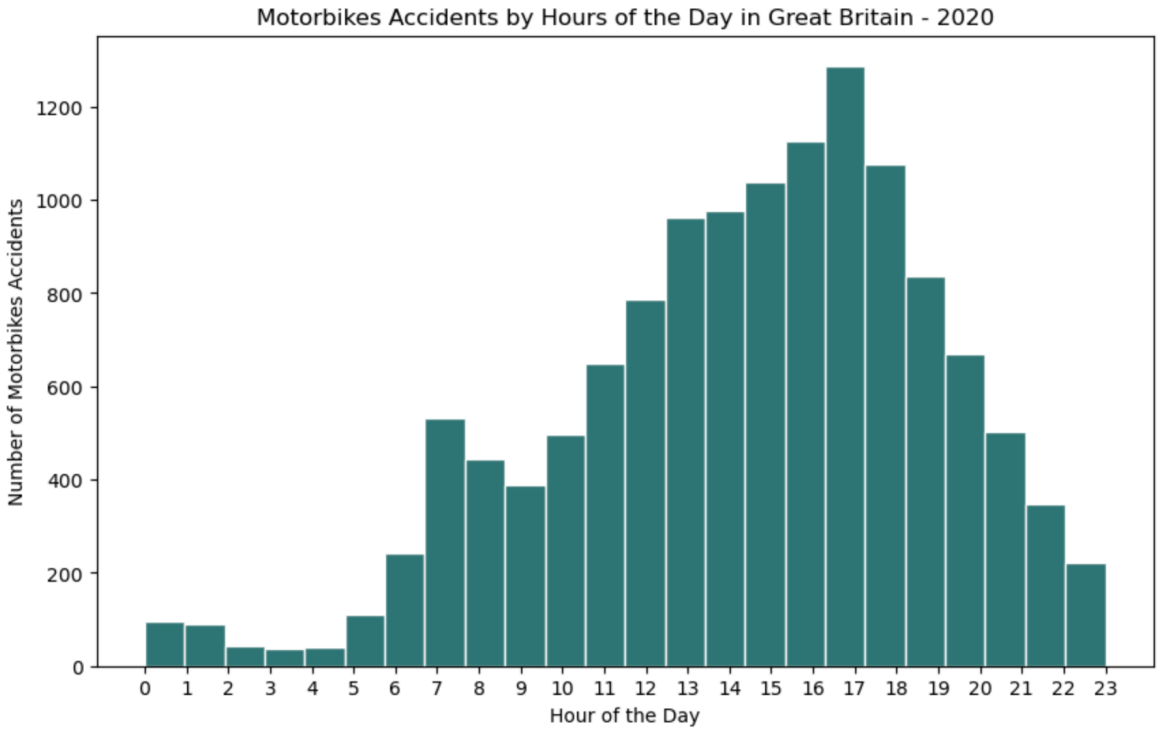
While progressing from Monday to Friday over the weekdays, there is a consistent increase. Friday registers the highest number of accidents in Great Britain, exceeding 14,000 cases. In contrast, weekends exhibit relatively lower accident counts in comparison to weekdays, with Sunday consistently reflecting the lowest count among all days of the week.



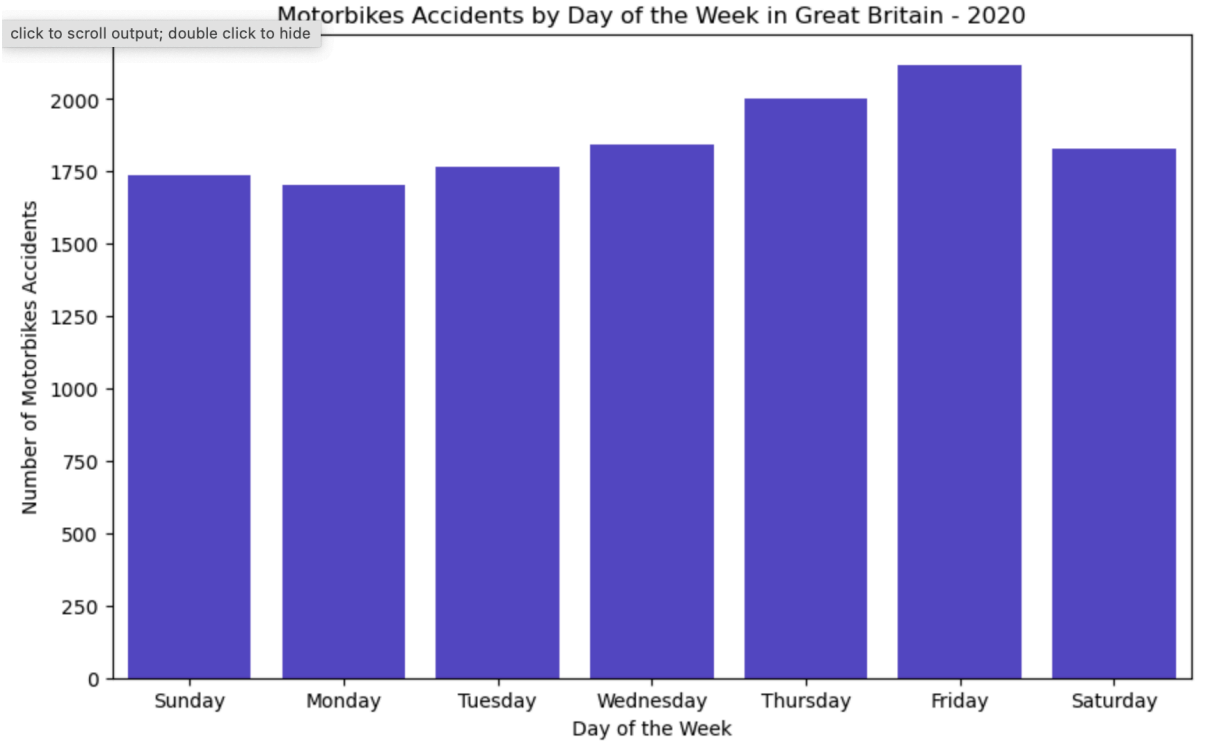
Accidents are highest in the morning in hours of 8am in Great Britain, it reduces and then steadily increases from 9am to 3pm. It peaks at 5pm and starts decreasing again. This pattern possibly indicates the morning rush hour and the evening commute hours when people are more likely to be on the road.

SIGNIFICANT HOURS OF THE DAY, AND DAYS OF THE WEEK, ON WHICH ACCIDENTS OCCUR FOR MOTORBIKES

The accident and vehicle dataframe were merge on accident index so the type of vehicle involved can be filtered by the motorbikes (Motorcycle 125cc and under, Motorcycle over 125cc and up to 500cc, and Motorcycle over 500cc) across all accidents in Great Britain which was used to create a new dataframe motorcycle_df. Extraction of the time and day of the week was carried out just like the analysis above. Histogram plots was also generated for these new extracted features.



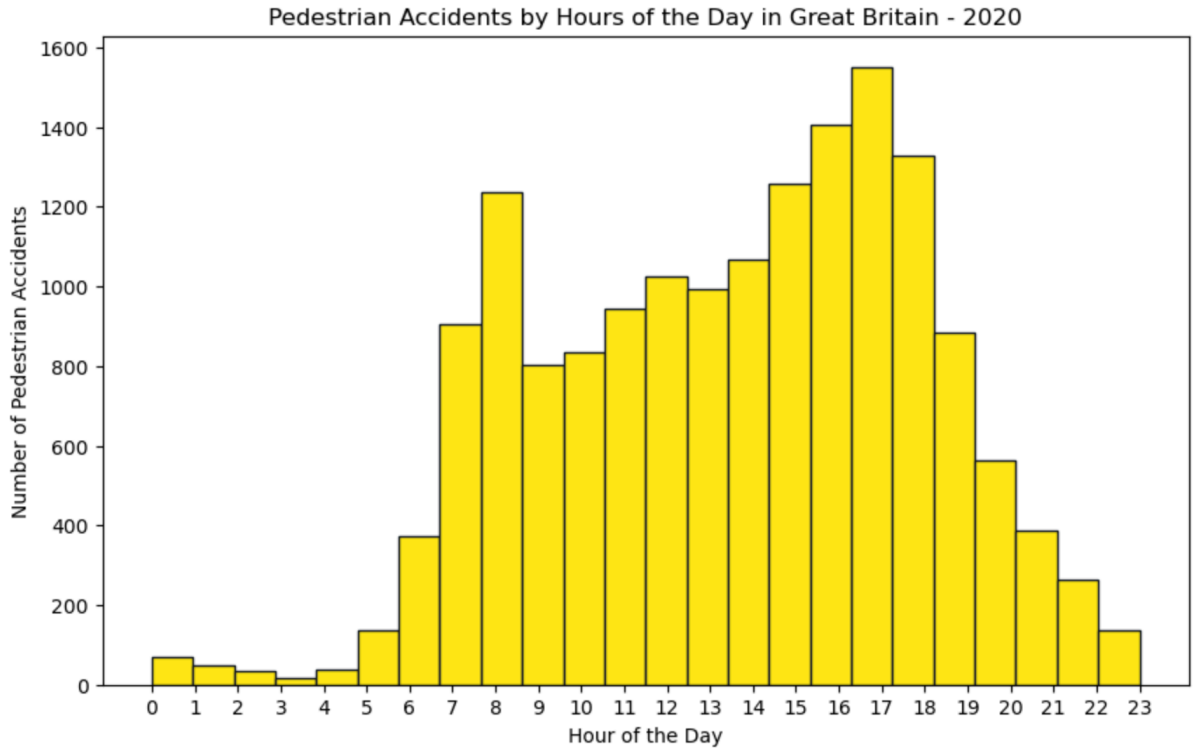
Among motorbike accidents in Great Britain, the morning hour of 7 am records the highest frequency. Following a brief decrease, accidents start increasing from 10 am and peaks at 5 pm, which registers as the hour with the maximum number of accidents (over 1200 accidents).



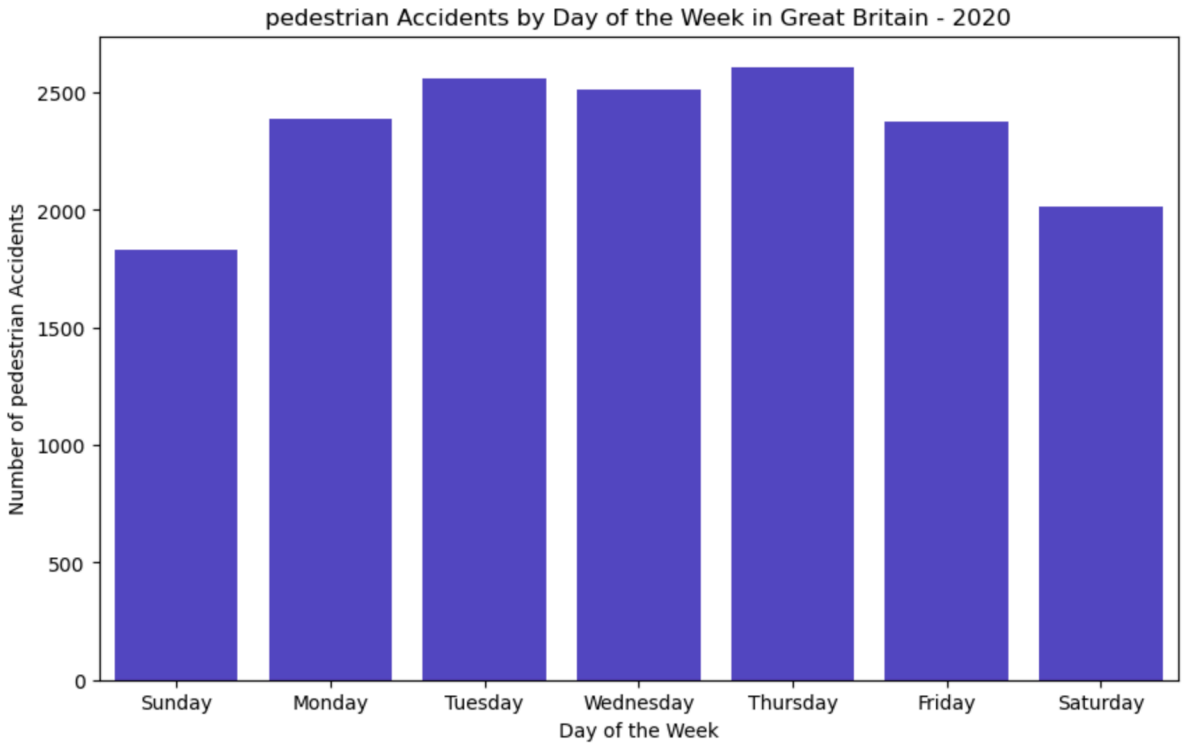
Trends for accidents for days of the week involving motorbikes in Great Britain shows Friday has the highest number of accidents. It can also be observed that as we approach weekend the accidents increases and peaks on Friday.

SIGNIFICANT HOURS OF THE DAY, AND DAYS OF THE WEEK, ON WHICH PEDESTRIANS ARE MORE LIKELY TO BE INVOLVED

The accident and casualty dataframe were merged on accident index filtered on casualty type pedestrian across all accidents in Great Britain. A new dataframe pedestrian_df was created. Extraction of the time and day of the week was carried out just like the two analysis above. Histogram plots was also generated for these new extracted features. Plots are shown below.



Regarding accidents involving pedestrians in Great Britain, the data indicates a peak occurrence during the morning hours at 8 am, likely corresponding to morning rush hours. Additionally, the hour of 5 pm emerges as the period with the highest number of accidents throughout the day. Interestingly, the timeframe between 3 pm and 6 pm witnesses elevated accident rates compared to other hours, potentially reflecting the rush associated with commuting home.



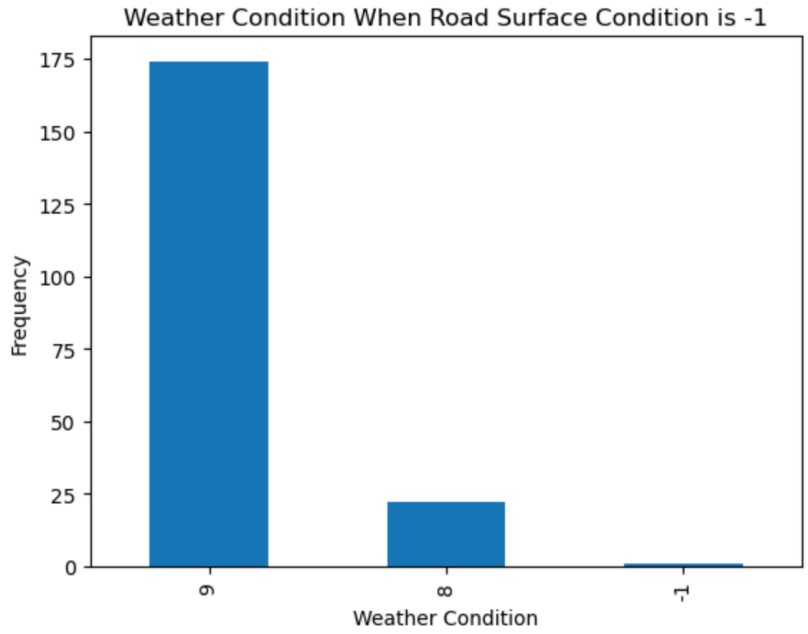
For days of the week, accident are highest on Thursday and lowest on Sunday. The low accident on Sunday reflects a day when most people don't work or go out. Accidents involving pedestrians are lowest on weekends. This pattern might suggest increased activity during weekdays, possibly due to higher traffic volume.

IMPACT OF SELECTED VARIABLES ON ACCIDENT SEVERITY

Using apriori algorithm on selected variables on accident severity I selected ROAD TYPE, LIGHT CONDITIONS and ROAD SURFACE CONDITION for accidents across Great Britain for this analysis. First the selected variables where cleaned

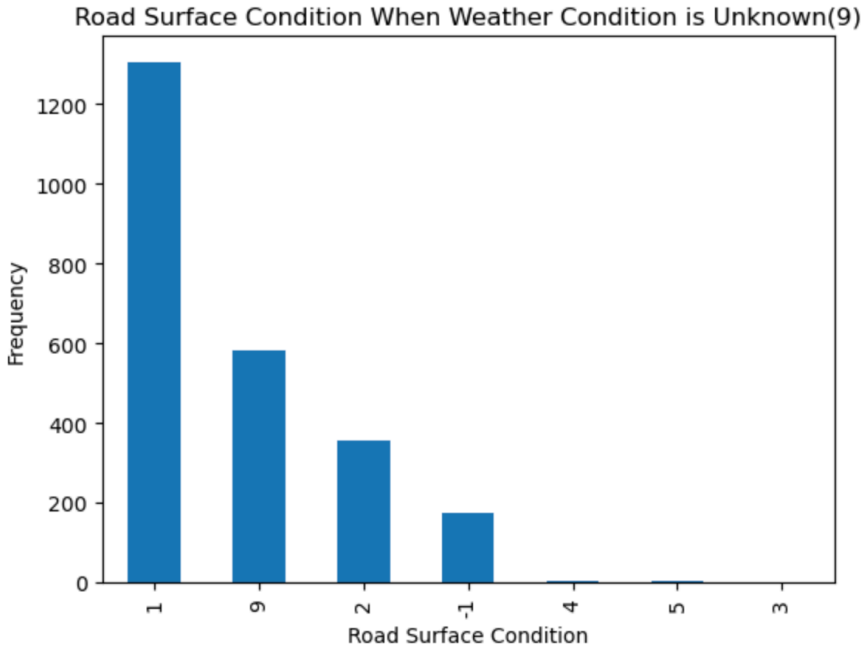
Data cleaning

I conducted an examination of all unique values within the variables to identify any instances deviating from the provided guidelines. The variable 'road_surface_conditions' contained instances of -1, indicating missing or out-of-range data. To address this, I cross-referenced the associated weather conditions to determine appropriate replacements for these instances. For example, instances characterized by both rain and high winds were assigned the 'wet/damp surface' road condition. Despite these adjustments, some rows still retained -1 values for road surface conditions. In response, I leveraged the predominant weather conditions associated with the remaining -1 instances to assign corresponding predominant road conditions. The outcomes of this process are depicted in the visualization below.



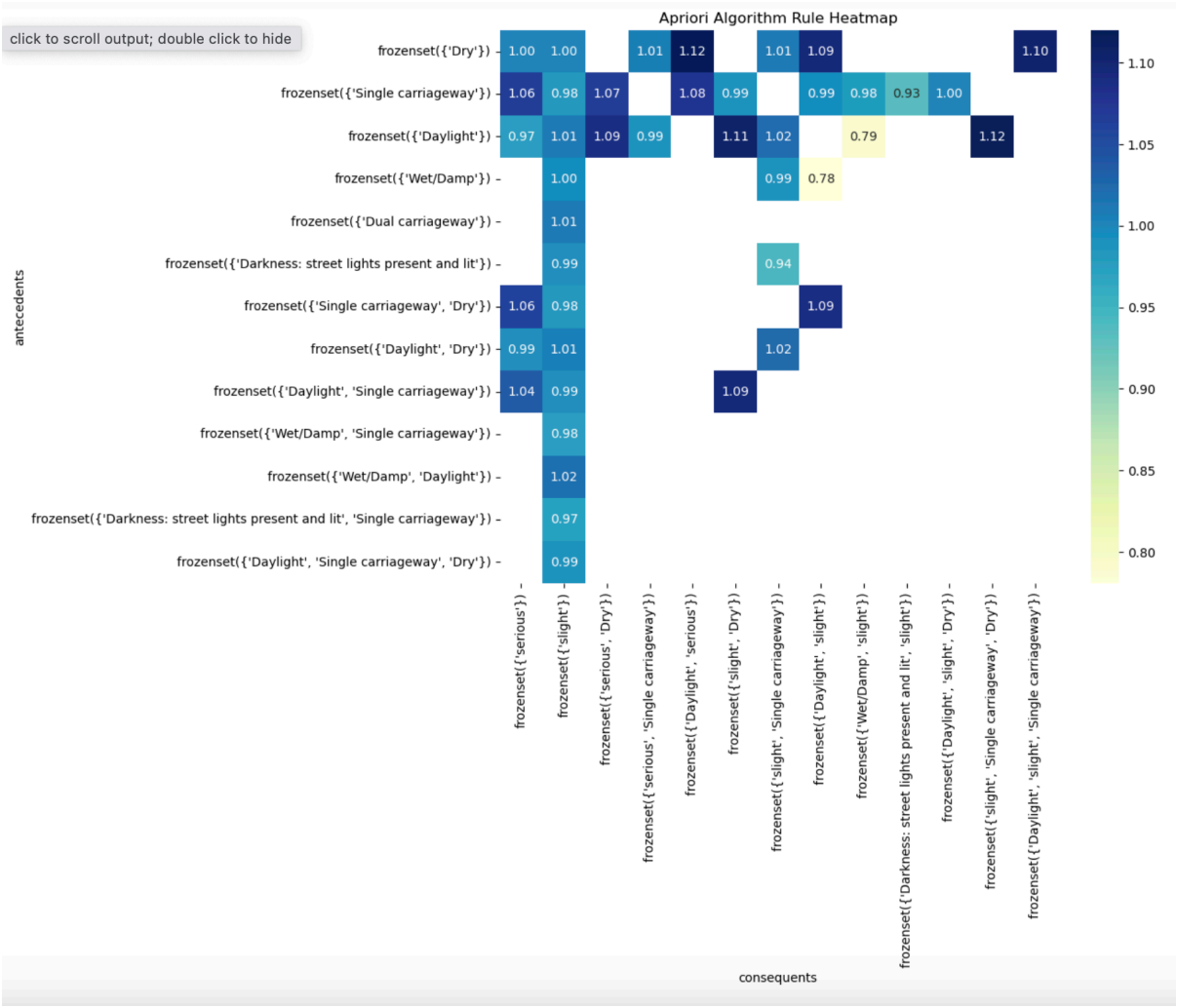
.

As evident from the figure above, Weather condition 9 holds dominance. I will proceed to examine the predominant road condition associated with Weather condition 9.



When Weather condition is 9, Road condition 1 constitutes nearly half of the instances. Consequently, for the remaining rows with -1 values, Road condition will be assigned as 1.

After completing the data cleaning process, I moved on to transform the individual variables using one-hot encoding and subsequently concatenating them. Following this, I applied the apriori algorithm to identify frequent itemsets with a support threshold of 10% (considering the dataset's size). I filtered these rules to display only rows associated with consequent accident severity. To visualize the relationships, I created a heatmap where cells are shaded according to the lift or confidence values of the respective rules.

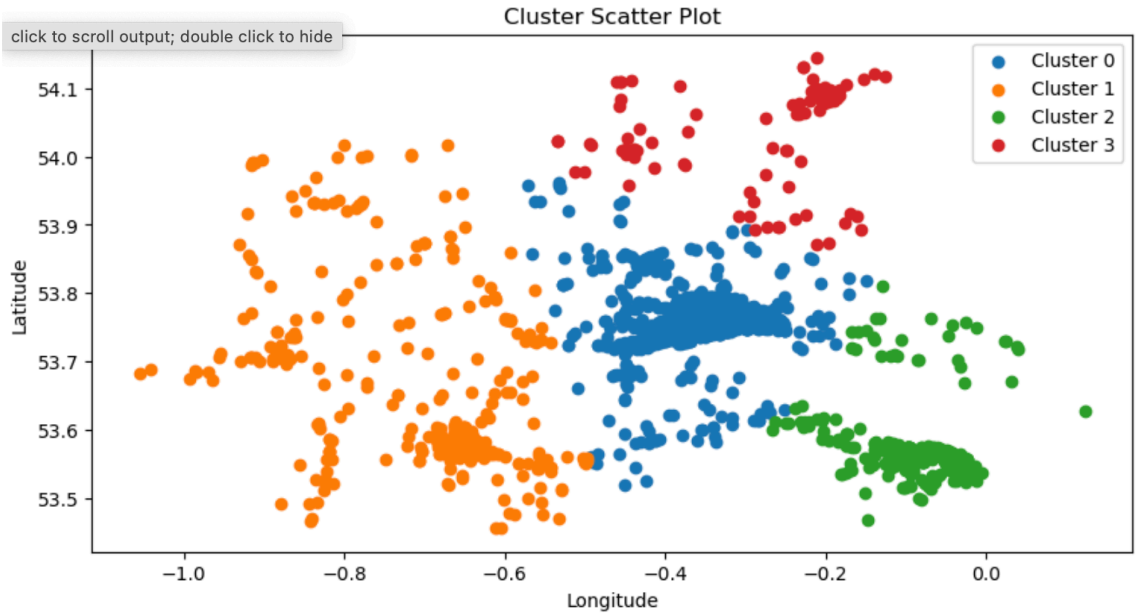


Upon analyzing the results presented above, it is evident that all lift values greater than 1 indicate a positive association between the items, whereas those less than 1 indicate a negative association.

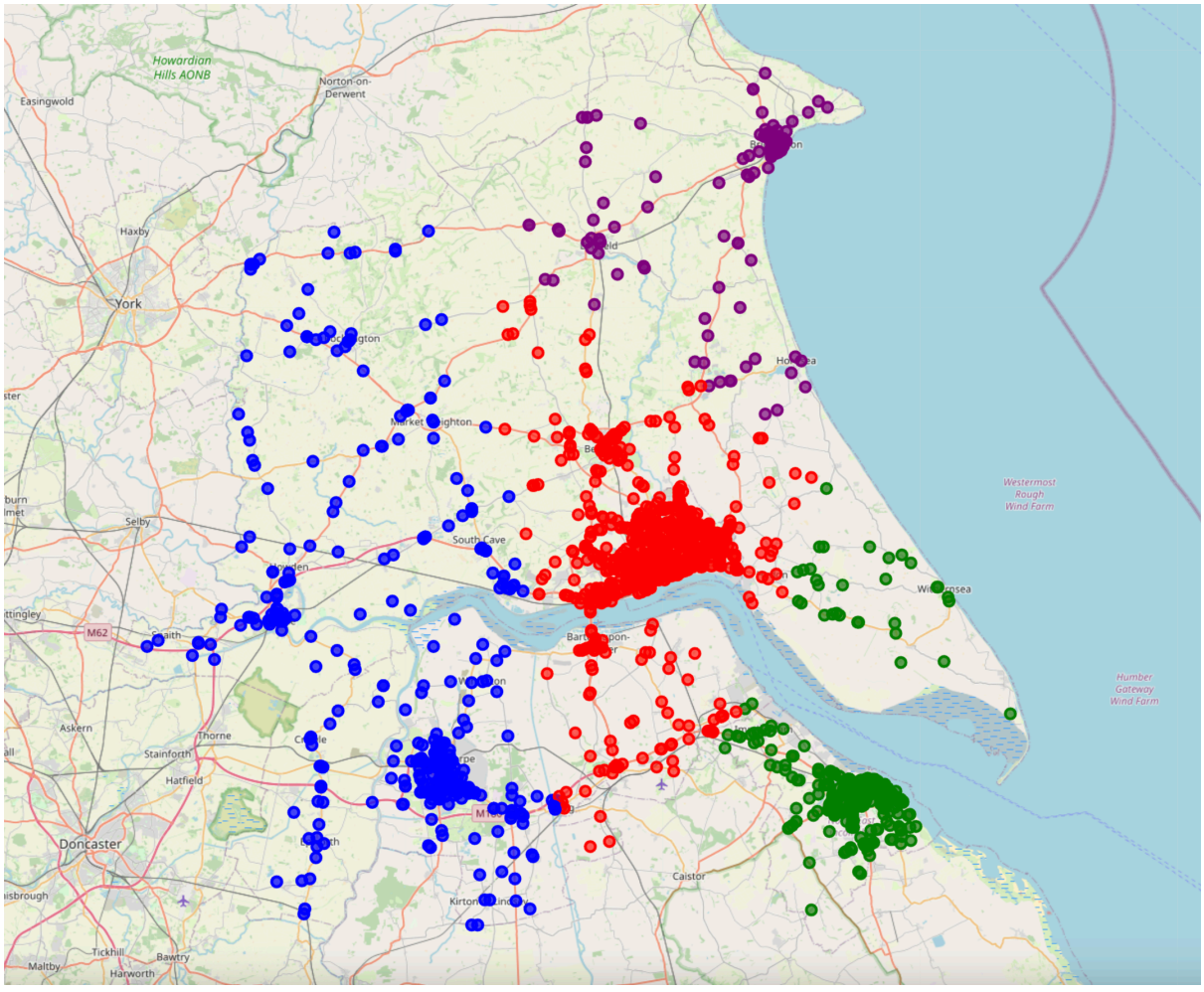
RUNNING CLUSTERING ON OUR REGION: KINGSTON UPON HULL, HUMBERSIDE, AND THE EAST RIDING OF YORKSHIRE ETC

My approach involved filtering the data by the police force 'Humberside', which covers both urban and rural areas including Kingston upon Hull, East Riding of Yorkshire, North Lincolnshire, and North East Lincolnshire. The filtered data was assigned to a dataframe named 'our_region'. To analyze the spatial distribution of accidents in this region, I employed k-means clustering. Specifically, I utilized longitude, latitude, and accident severity columns for clustering.

For determining the optimal number of clusters (k value), I used function for this. Subsequently, I created an elbow graph to visualize the most suitable k value, which turned out to be 4. I further plotted a scatter plot illustrating the distribution and naming of the clusters.

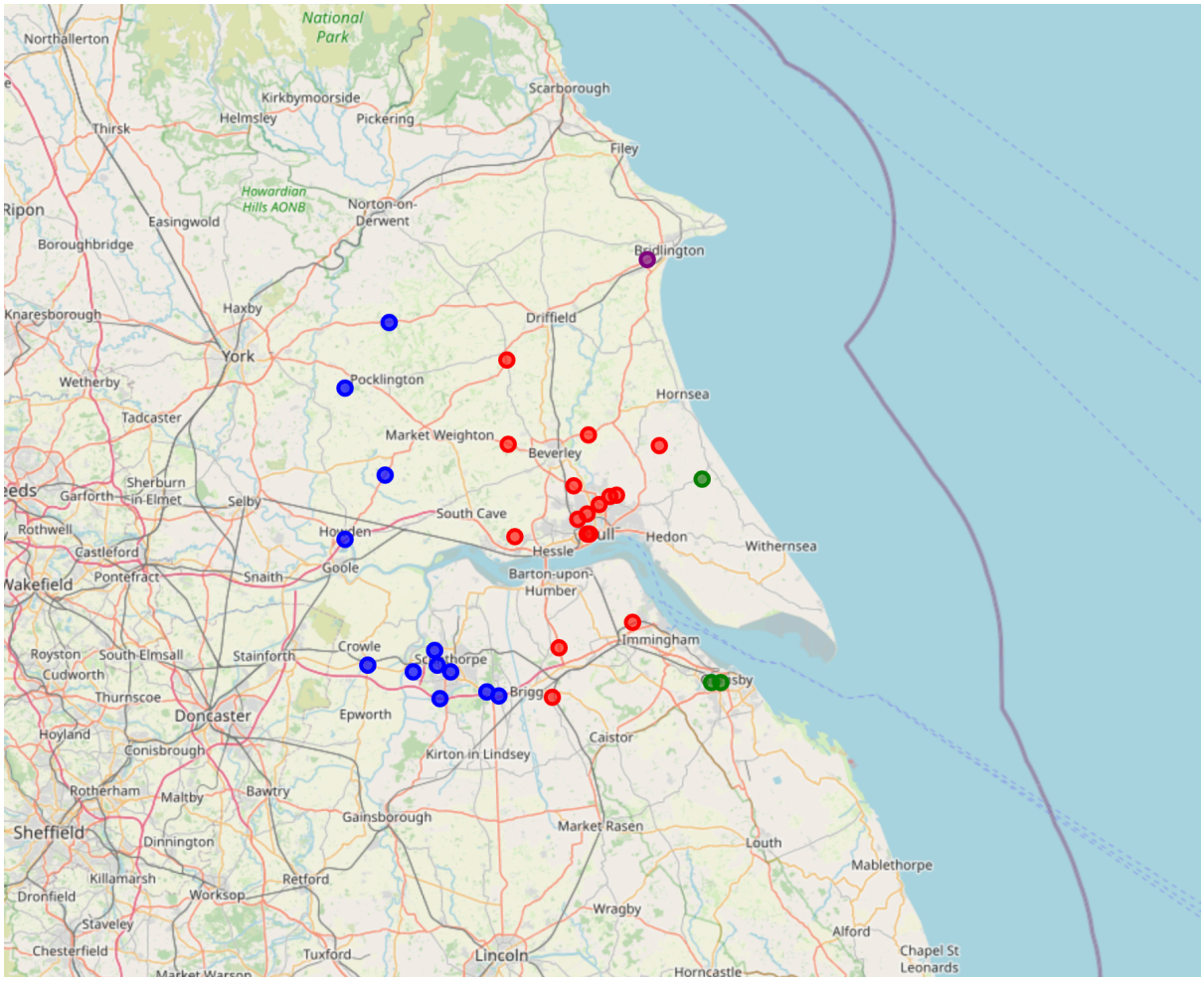


Also, I visualise how the clusters are distributed in a map below to get more information on the clusters.



The map above reveals a notable concentration of accidents in urban zones, such as Hull city center, Scunthorpe, and Grimsby. Additionally, I refined the map to exclusively display

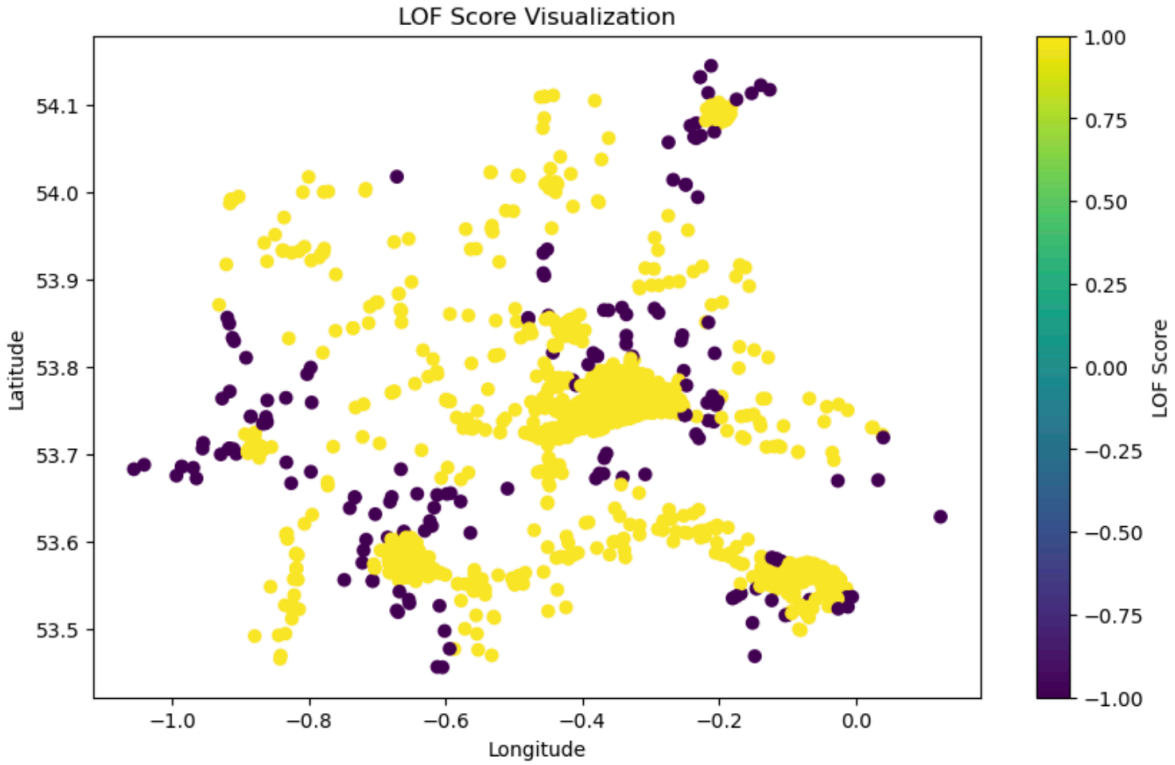
accidents categorized as 'fatal', which allows for a focused observation of areas with a higher frequency of fatal accidents.



The map clearly indicates a higher concentration of fatal accidents in urban areas like Hull and Scunthorpe. Notably, Bridlington exhibits a comparatively lower occurrence with only one recorded fatal accident. This shows the risk associated with fatal accidents in densely populated regions.

USING OUTLIER DETECTION METHODS, IDENTIFY UNUSUAL ENTRIES IN YOUR DATA SET

I conducted an outlier detection analysis specifically focusing on the longitude and latitude variables in our region under the Humberside police force. To achieve this, I employed the Local Outlier Factor (LOF) technique. I then created a scatter plot that show side by side the regular data points with the identified outliers. The visualization was generated by associating the LOF scores with the respective latitude and longitude values, providing a clear representation of potential anomalies within the dataset.



Maintaining outliers within the longitude and latitude data is essential due to their capacity to offer a comprehensive and detailed depiction of the geographical area under consideration. By retaining these outliers, the analysis gains a broader perspective and a better representation of the geographic landscape just like the map I visualized for accidents in our region.

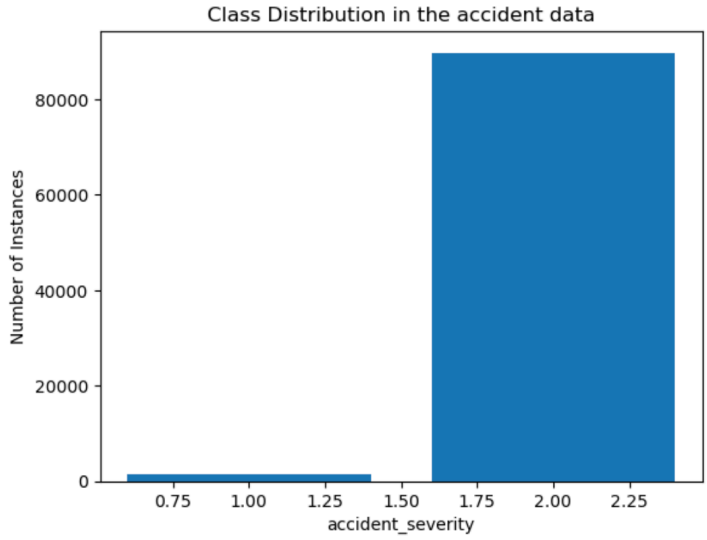
PREDICTION

In approaching this, my initial step involved data cleansing by addressing null values. Subsequently, I combined accident severity levels 2 and 3 into a unified severity level 1, essentially categorizing accidents as either fatal or non-fatal. This consolidation streamlined the severity classification.

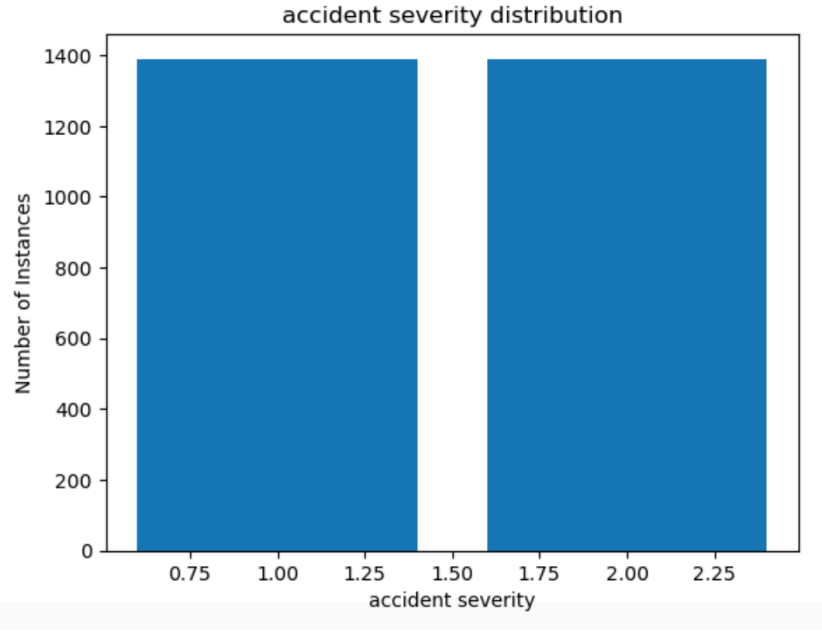
To ensure a balanced distribution of the target variable and avoid potential bias, I performed data balancing. This was crucial in preventing overrepresentation of a specific target outcome, which could compromise model effectiveness. Following this, I employed the K Nearest Neighbors algorithm to construct a predictive model that determines whether an accident is fatal or non-fatal.

For handling rows with values of -1, signifying missing or out-of-range data as per the guidelines, I employed a forward-fill strategy. This choice was driven by the prevalence of repeating values in sequential rows, which lent itself well to this approach. Additionally, I addressed null values in the longitude and latitude columns, as these parameters couldn't be deduced from other data columns.

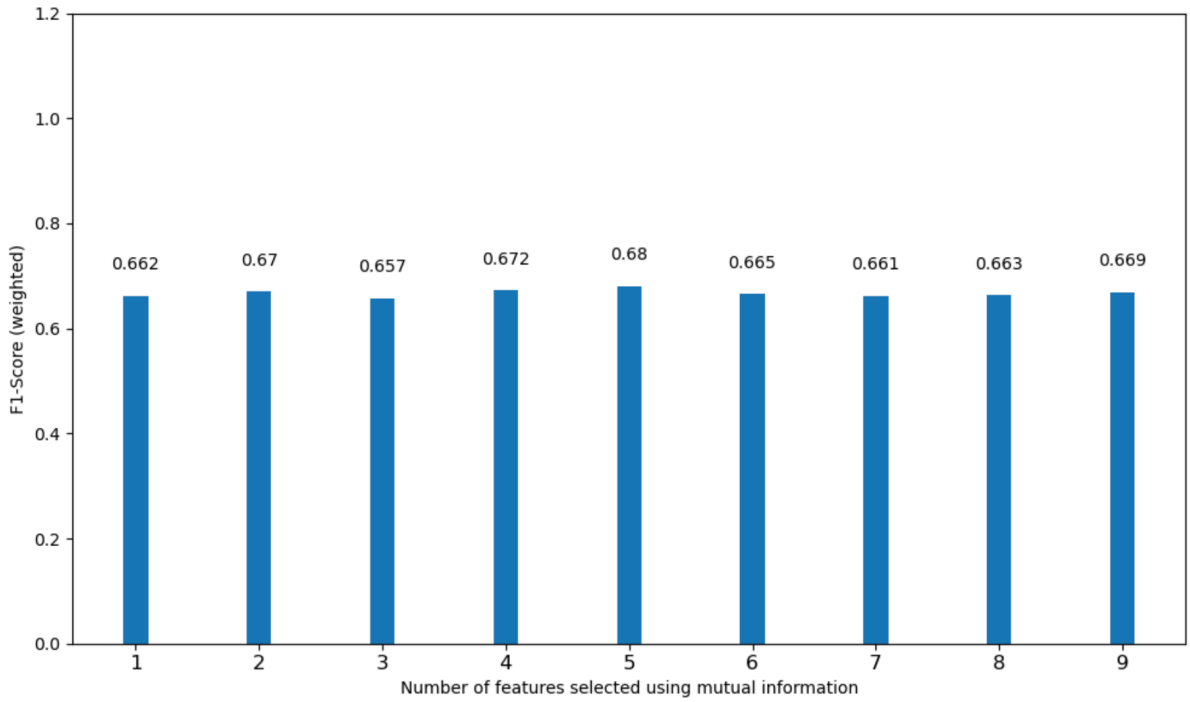
To address the issue of data imbalance, I examined the distribution of the target variable and identified its unevenness. This can be observed in the visualization provided below.



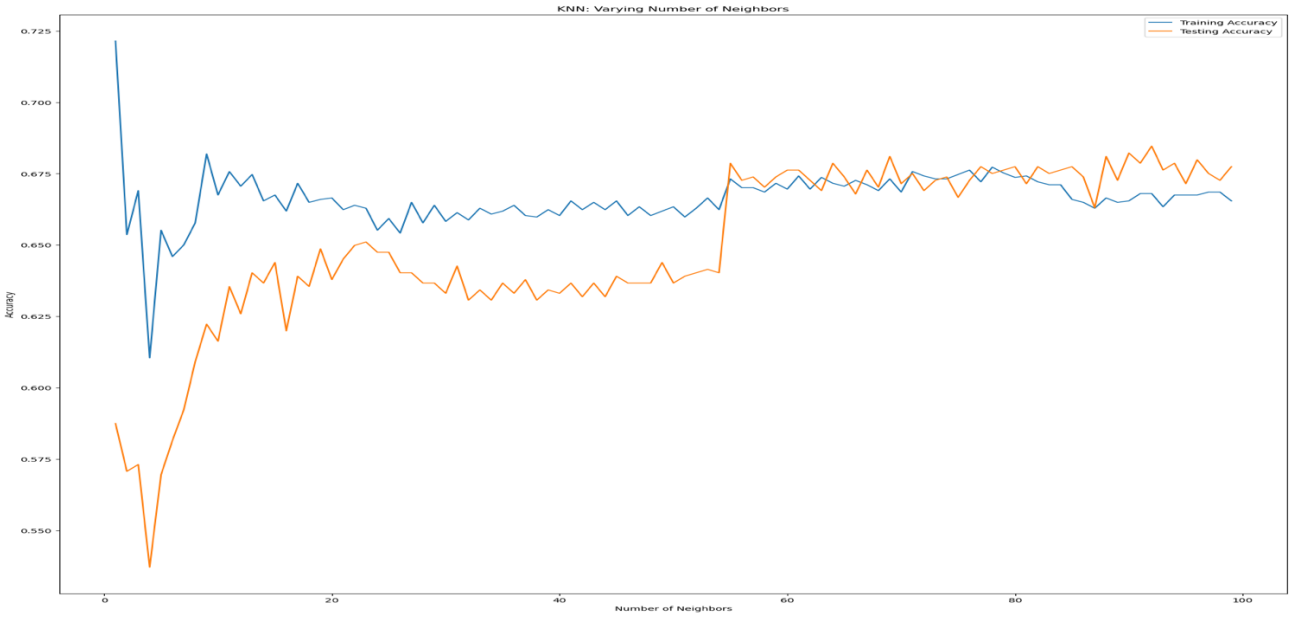
I then employed the "imblearn" library to perform undersampling, this balance the dataset by randomly reducing the occurrence of the severity 2 class. After applying this technique, a checked to confirm if this was successful as shown in the visual representation provided below.



For the data preprocessing, the dataset was divided into training and testing subsets. Feature engineering was performed to identify the most influential attributes for constructing the model. The initial performance of the model was assessed using the GradientBoostingClassifier, providing a baseline F1-score for reference. To optimize the model's performance further, the mutual_info_classif method was employed, varying the number of selected features (k) to enhance the F1-score. Through this process, it was determined that an optimal feature count of 5 yielded the highest F1-score of 0.68, as illustrated in the figure provided below.



The analysis subsequently revealed the optimal selection of features, namely 'speed_limit', 'junction_detail', 'junction_control', 'light_conditions', and 'weather_conditions'. This set of features was deemed the most influential for the model's predictive accuracy. Following this, I instantiated the K-Nearest Neighbors (KNN) classifier with n_neighbors set to 57. Subsequently, the model underwent an assessment for complexity, overfitting, and underfitting to determine the ideal number of neighbors that would ensure an optimal balance between these factors.



I visualized a confusion matrix to present an overview of the model's performance. Additionally, precision, recall, and F1-score to gain a more detailed understanding of the model's effectiveness to give deeper insights on model's capabilities and performance.

RECOMMENDATION

- Based on the analysis, it is recommended to implement reduced speed limits during peak traffic hours, such as 8am in the morning and 3pm to 6pm in the evening. This can help mitigate the risks associated with rush-hour congestion.

- Regarding feature selection for predicting accident severity, factors like speed limit, junction detail, junction control, light conditions, and weather conditions have shown significance. Implementing additional road safety measures aligned with these conditions is advisable.

- In urban areas like Hull and Scunthorpe, prioritizing the installation of pedestrian crosswalks and enhancing signage and road markings is crucial. This will improve pedestrian safety and reduce the risk of accidents.

- Particular attention should be given to single-carriage roads, as indicated by the heatmap generated from the Apriori algorithm. Strengthening road safety measures on such roads is essential to address the higher incidence of fatal accidents associated with them.

REFERENCE

Police and crime landscape online: https://www.humberside-pcc.gov.uk/Our-Work/PCC-Election-2021/Police-and-Crime-Landscape.aspx [Accessed 1/8/2023].

Department for Transport (2021) STATS19 forms and guidance. GOV.UK. Available online: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/995422/stats19.pdf [Accessed 15/7/2023].

Department for Transport (2011) STATS20 Instructions for the Completion of Road Accident Reports from non-CRASH Sources. GOV.UK. Available online: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/995423/stats20-2011.pdf [Accessed 15/7/2023].

K-Means Clustering Algorithm with Python Tutorial online: https://youtu.be/iNlZ3IU5Ffw?t=523 [Accessed 1/8/2023]