DATA MINING AND TEXT ANALYTICS (MANM529) - SEMR2

Individual Assignment

Exploring Road Traffic Accident Data and Text Analytics Insights

Dina Usama Ismail

Student ID: 6904690

Main Objective

This study analyzes UK road accident data (Surrey 2021), beginning with data cleaning, quality assessment, and exploratory analysis to identify key features for predicting accident severity. Additionally, text analytics on related tweets will supplement findings and address gaps in primary dataset.

Task 1: *Data Exploration, Cleaning, and Feature Selection*

Based on exploring the dataset through excel and the data quality report created using SAS Viya 'Discover Information Asset' app from where a snippet is shown in Figure 1, a data cleaning strategy has been devised.

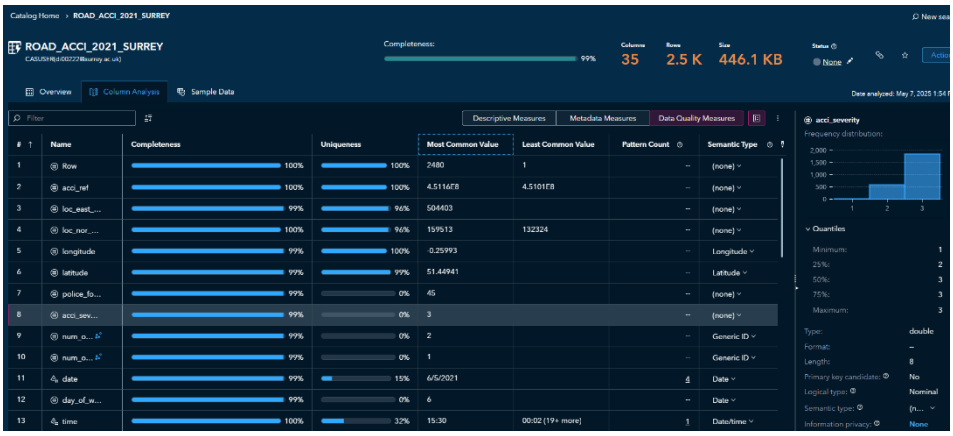


Fig. 1 Snippet from Data Quality Report by Discover Info Asset SAS Viya

According to the dataset guidelines, there are two groups found in most of the variables: **unclassified/unknown** and **missing**. This is critical to distinguish because the **-1** values indicate missing data that will be imputed while unclassified values will be treated as a category in analysis.

The **acci_ref** variable contains 2480 unique values for 2021 with no duplicates. The **police_force** is constant at 45 (Surrey), so any missing values will be set to 45. The **local_auth_distr_all** variable uses a deprecated code and will be excluded from study.

The data guidelines state that a -1 in **second_road_number** indicates an unclassified first road, but in the actual data it represents unclassified second roads, showing an inconsistency. Moreover, both **junc_control** and **sec_road_number** have more than 1300 missing entries accounting to more than half of the data; thus, these variables will be dropped from the study.

Data Cleaning Stage 1

Details of the pipeline using Build Model App: ***Task 1_Road_2021_Data_Cleaning (Stage 1)_Pipeline (Build Model App).pdf***
Output of this pipeline = ***Road_Accident_Cleaned_Final_Dataset.csv***

This stage addresses missing values (blank and -1) using Build Model app. As shown in figures 2 and 3, the pipeline integrates both imputation nodes and custom SAS code for advanced imputation.

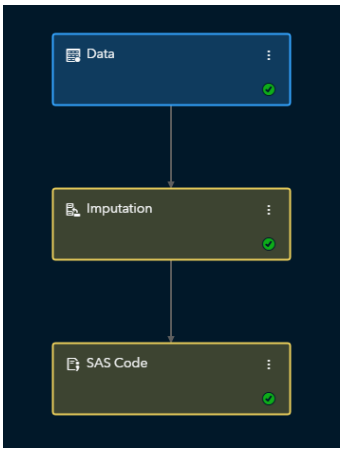


Fig. 3 Data Cleaning Stage1 Pipeline

```
data new_dataset;
    if loc_auth_highw ne "E10000030" then loc_auth_highw= "E10000030" ;/* replace missing variable with this value (cnstant value)

/* Remove rows where did_poli_att not equal to 1, 2, or 3 'var_name' equals 'value' */
    if did_poli_offi_att not in (1 ,2 ,3, -1) then delete;

 /* Remove rows where longitude, latitude, loc_east_osgr, or loc_nor_osgr are missing' */

    if cmiss(latitude,longitude,loc_nor_osgr,loc_east_osgr) > 0 then delete;

 /* Adds random date to emplty date field with the condition of keeping the year 2021 */
    if missing(date) then do; /* Replace 'date_var' with your variable name */

        /* Define start and end dates for 2021 */
        start_date = '01JAN2021'd; /* SAS numeric date value for 01/01/2021 */
        end_date = '31DEC2021'd;   /* SAS numeric date value for 12/31/2021 */

        /* Generate a random date between start_date and end_date */
        random_days = floor((end_date - start_date + 1) * rand('uniform')); /* Days between 0 and 364 */
        random_date_num = start_date + random_days; /* Numeric SAS date */

        /* Convert numeric date to character with mm/dd/yyyy format */
        date = put(random_date_num, mmddyy10.); /* Format as mm/dd/yyyy */
    end;

    /* Ensure the variable remains character type */
    format date $10.; /* Explicitly set length to match mmddyy10. format */
```

Fig. 2 Snippet from SAS code node in Data Cleaning Stage 1

Below are variables that require cleaning;

- **police_force**: missing values replaced with 45 using the SAS code.
- **acci_severity**: One missing value, row removed using SAS code
- **num_of_vehi** and **num_of_cas:** one missing value from each, imputed by "count" using Imputation Node
- **date:** all dates correspond to 2021; missing values randomly generated ensuring the year is 2021 using SAS code
- **day_of_week:** missing values will be generated from the corresponding random date generated to ensure consistency using SAS code
- **loc_auth_highw**: one missing value, replaced by E10000030 which is constant using SAS code
- **did_poli_off_att** rows with values not defined in data guidelines were removed using SAS code
- **Carri_haz, Spec_con_site,** and **Road_surf_con:** -1 values replaced with mode using SAS code, as they are few
- **Latitude, longitude loc_east_osgr**: spatial variables can't be imputed with a mean; typically, clustered mean is used. However, since one row is missing it was removed using SAS code

Data Cleaning Stage 2

Details of flow using Develop Flow and Code SAS Viya: ***Task 1_Road_2021_Data_Cleaning_Preprocessing Flow(Stage2)(Develop Codes and Flow App).pdf***

Output file = ***Cleaned_Road_Accident_2021_Final.csv***

This stage is for final tweaking of data using the flow in figure 4. Below are the steps done:

1) Creation of new time and date variables using SAS program (snippet in figure 5):
    - hour_of_day
    - time_category
    - months
2) Dropping of unnecessary columns
    - dm_index
    - Police_force: constant value
    - local_auth_distr: constant deprecated value
    - loc_auth_highw: constant value E10000030
    - junc_control and second_road_number
3) Renaming variables:
    - renaming the columns to remove IMP prefix
4) Reordering the columns in a more logical order

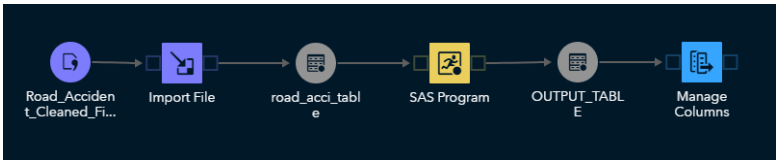


Fig. 4 Data Cleaning Stage 2 Flow Process

```
data work.output_table;
set road_acci_table;
length _months_ $3;

length time 8;
format time time8.;

/* Convert the date variable to SAS date value */
sas_date = mdy(month(date), day(date), year(date));
/* Check month ranges and assign labels accordingly */
if ('01JAN2021'd <= sas_date <= '31JAN2021'd) then _months_= 'jan';
else if ('01FEB2021'd <= sas_date <= '28FEB2021'd) then _months_= 'feb';
else if ('01MAR2021'd <= sas_date <= '31MAR2021'd) then _months_ =
'mar';
else if ('01APR2021'd <= sas_date <= '30APR2021'd) then _months_= 'apr';
else if ('01MAY2021'd <= sas_date <= '31MAY2021'd) then _months_=
'may';
else if ('01JUN2021'd <= sas_date <= '30JUN2021'd) then _months_= 'jun';
else if ('01JUL2021'd <= sas_date <= '31JUL2021'd) then _months_= 'jul';
else if ('01AUG2021'd <= sas_date <= '31AUG2021'd) then _months_= 'aug';
else if ('01SEP2021'd <= sas_date <= '30SEP2021'd) then _months_= 'sep';
else if ('01OCT2021'd <= sas_date <= '31OCT2021'd) then _months_= 'oct';
else if ('01NOV2021'd <= sas_date <= '30NOV2021'd) then _months_= 'nov';
else if ('01DEC2021'd <= sas_date <= '31DEC2021'd) then _months_= 'dec';
else _months_= 'other';
format sas_date date9.; /* Optional: Set the format for display purposes */

/* Extract the hour from the time variable */
hour_of_day = hour(time);
/* Categorize time into different periods */
if 0 <= hour_of_day < 6 then time_category = 'night';
else if 6 <= hour_of_day < 12 then time_category = 'morning';
else if 12 <= hour_of_day < 18 then time_category = 'afternoon';
else if 18 <= hour_of_day <= 23 then time_category = 'evening';
else time_category = 'other';
```

Fig. 5 Code Snippet from SAS Program in Data Cleaning Flow Process Stage 2

Data Analysis and Features Selection

Stage 1

Details of flow using Develop Flow and Code SAS Viya: ***Task 1_Data Analysis and Feature Selection _Road_Accident_SQLMerge_Descirption Flow(Stage 1)(Develop Code and Flows app).pdf***

Data after using SQL queries and joins = ***Cleaned_Road_Accident_2021_MergedwithDescription.csv***

Data visualization through charts and statistics will be conducted to determine the important variables and select the most important features to be fed into the predictive model. For the results to be easily interpreted, SQL queries are created to merge some of the variables that are encoded with their corresponding descriptions in the data guidelines using key variables. This stage accomplishes this using the Develop Flows app.

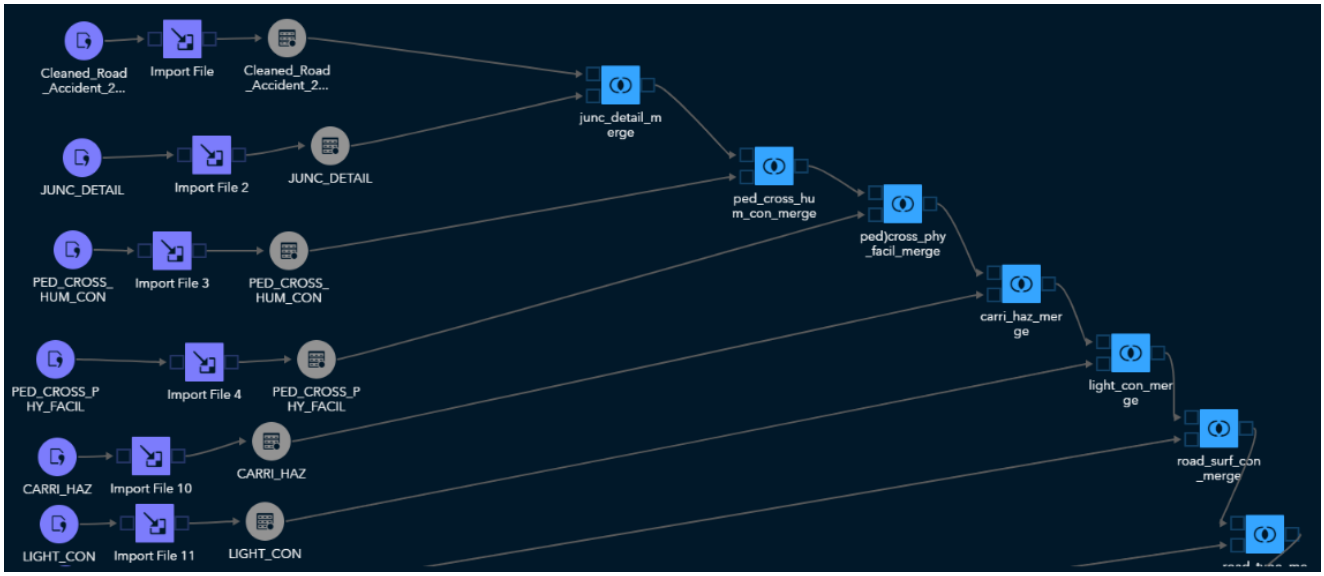


Fig. 6 Snippet from SQL Merge

Stage 2

*Details for all visuals using Explore and Visualize SAS Viya****: Task 1_Road_Accidents_Reporting (Exploratory Data Analysis using Explore and Visualize App).pdf***

This stage focuses on variable analysis and feature selection. As shown in figure 7, the data is imbalanced with “slight” severity dominating. While this limits the ability to assess variables relationships for each severity category, the analysis can still offer insights about which attributes impact accidents overall.

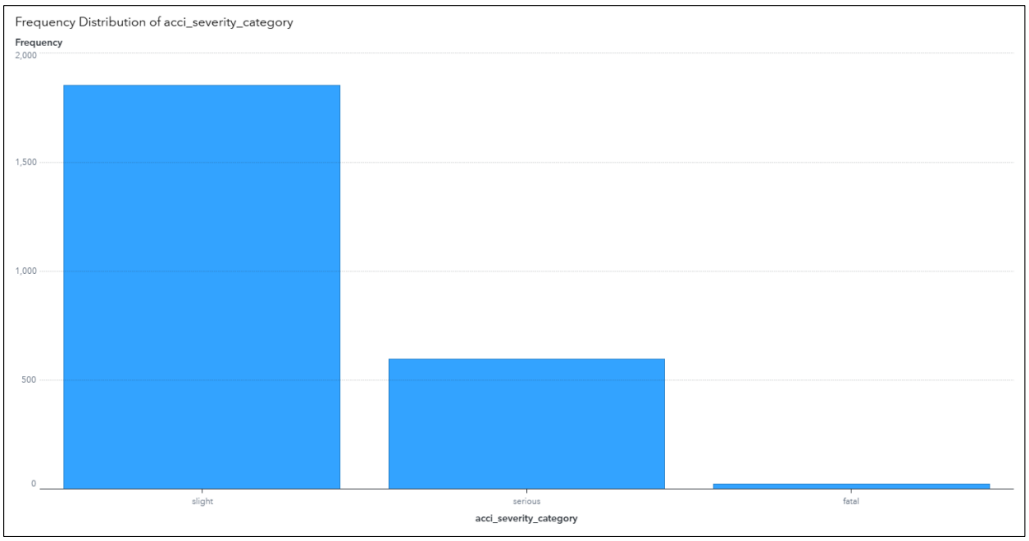


Fig. 7 Accident Severity Category Frequency Chart

Attributes are grouped into road/environmental, time, accident circumstances, and location factors. Since most variables are categorical, relationships will be explored using visual charts rather than Pearson correlation matrix.

Variables Classification			
Road and environmental variables	**Time Variables**	**Accident Circumstances**	**Location Variables**
weath_con	day_of_week	num_of_casu	longitude
road_type	time_category	num_of_vehi	latitude
road_surf_con	_months	did_poli_offi_att	lsoa
spec_con_site			loc_auth_on_distric
junc_detail			loc_nor_osgr
light_con			loc_east_osgr
ped_cross_hum_con_desc			
tru_road_flag			
ped_cross_phy_facil_desc			
Cari_haz			
urban_or_rural			
first_road_class			
second_road_class			

Table 1 Variables Classification

Road and environmental variables

Starting with weather conditions, as shown in figure 8, all accidents took place in fine no high wind weather. It seems that the different weather does not have an impact on the frequency of the accidents; this could be due to data collection issues. However, factually weather is a critical factor, and it can have an impact on severity therefore it will be included.

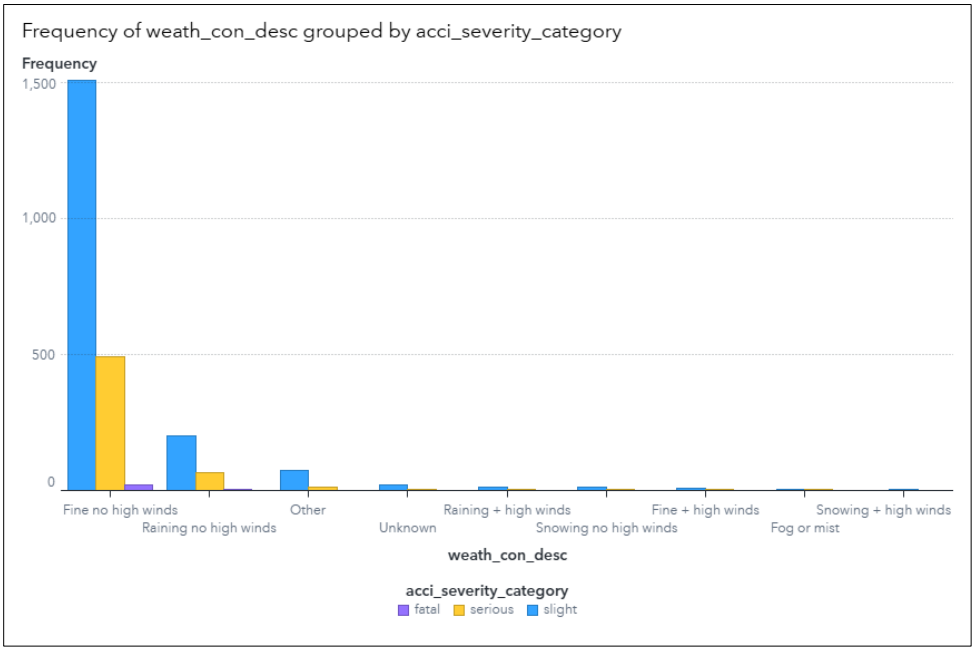


Fig. 8 Frequency of accidents in different weather conditions

Moving on to road types and road surface conditions in figures 9 and 10, there is some variation across different categories where single carriageway roads that are dry have the highest number of accidents. Hence, both variables will be included.

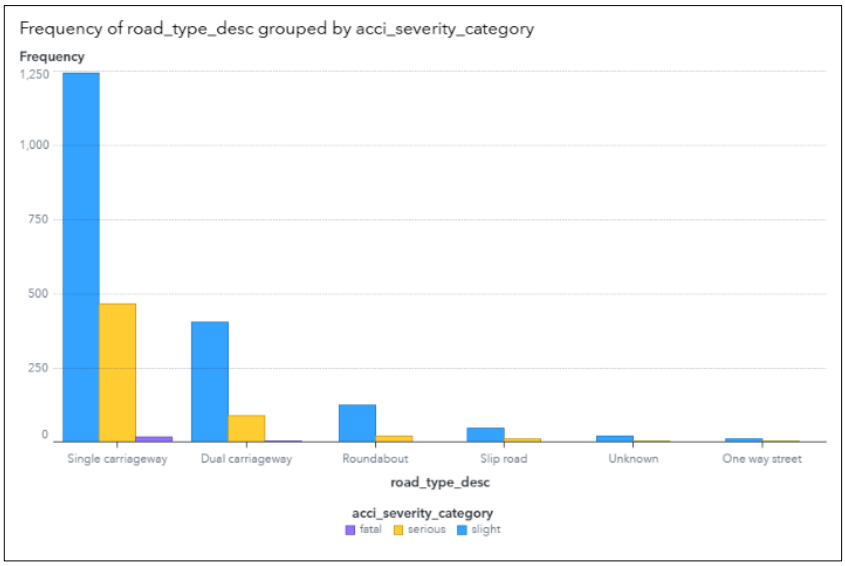


Fig. 9 Frequency of accidents in different road types

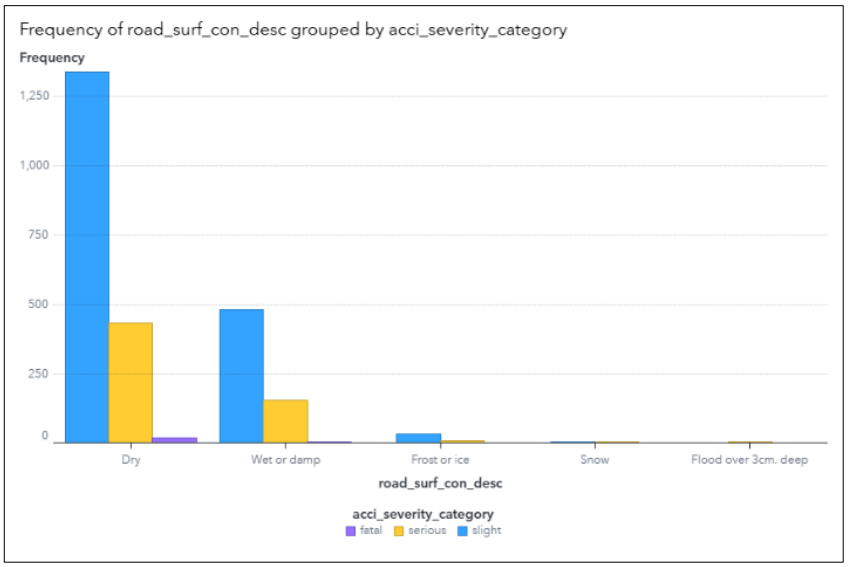


Fig. 10 Frequency of accidents in different road surface conditions

As shown in figures 11 and 12, there is variation in the accident frequency across different junction details and light conditions; these variables should be considered.

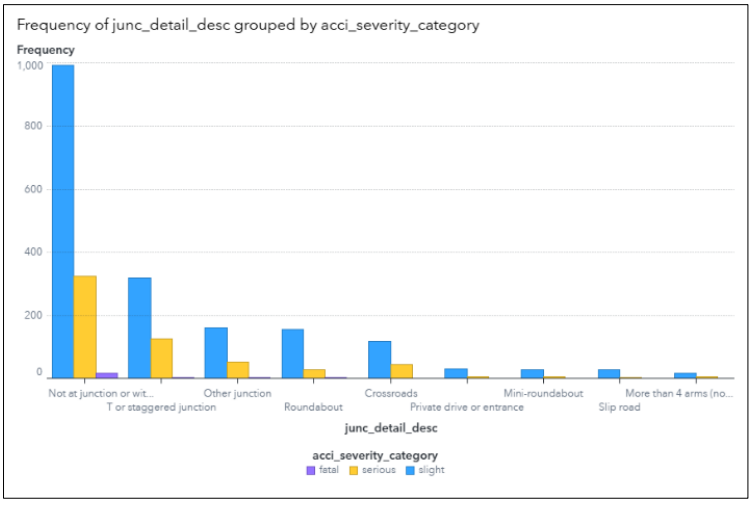


Fig. 11 Frequency of accidents in different junctions

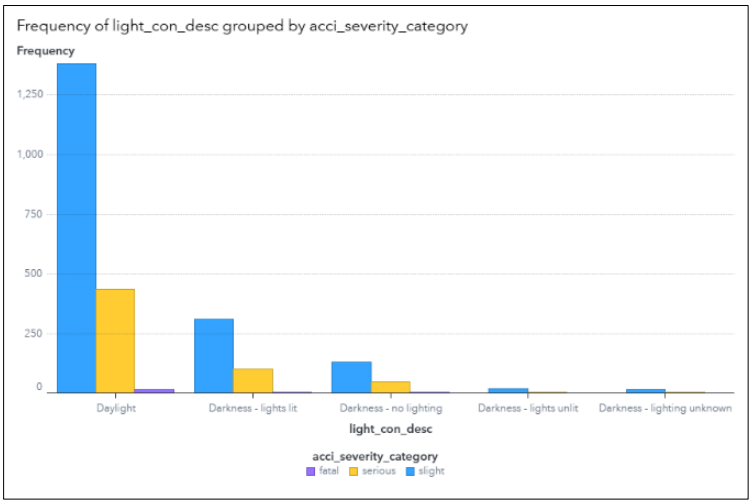


Fig. 12 Frequency of accidents in different light conditions

Urban or rural and whether the road is a trunk road (major road in UK that are key routes for long-distance traffic strategic) or not are factors that have an impact on accidents as shown in the below figures 14 and 13, and they will be considered.

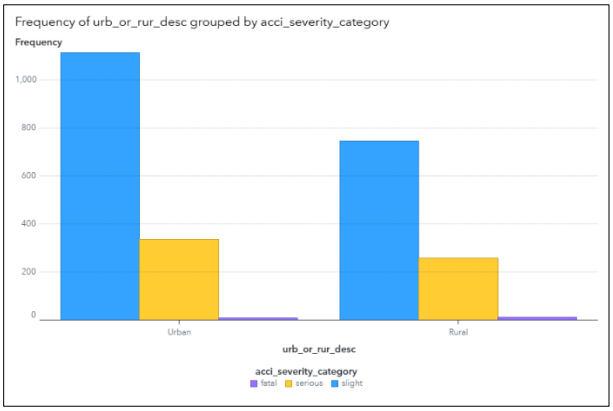


Fig. 14 Frequency of accidents in rural and urban areas

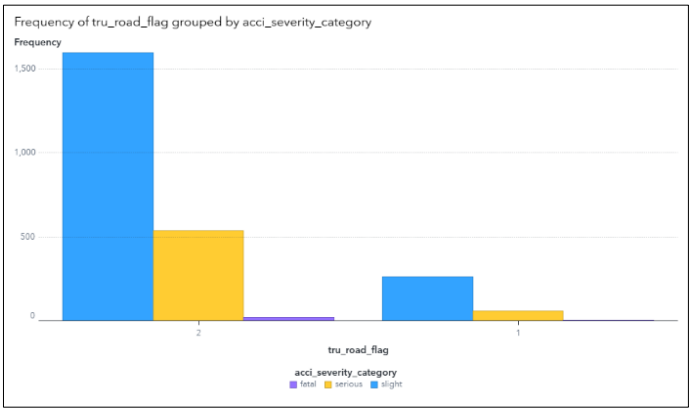


Fig. 13 Frequency of accidents in trunk and non-trunk roads

There is variation in the accident frequency across different first roads (which is the main road where the accident occurred) and second road types (which is the intersecting road at junction if applicable); these variables will be considered.

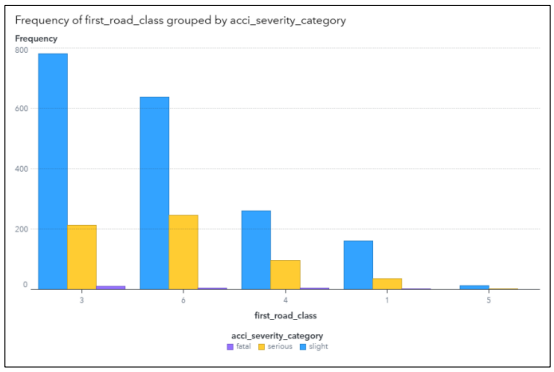


Fig. 15 Frequency of accidents in different first road classes

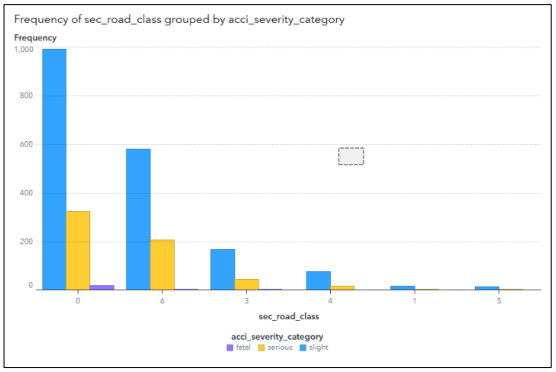


Fig. 16 Frequency of accidents in different second road classes

Carriageway hazards do not seem to have an impact on the accidents as shown below because almost all the accidents take place in places without carriageway hazards; this variable will be dropped.

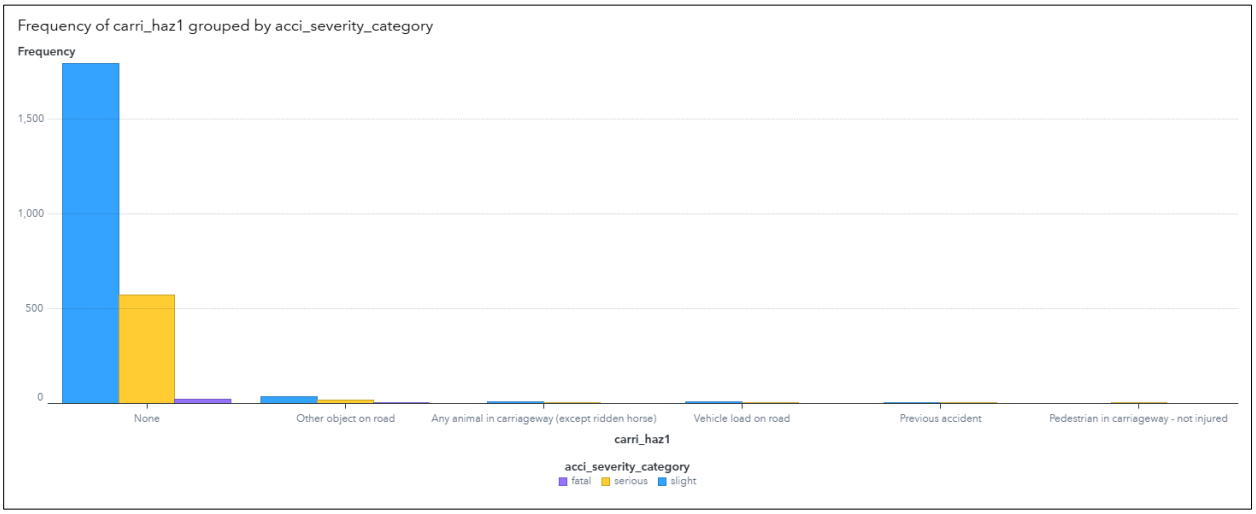


Fig. 17 Frequency of accidents across different carriageway hazards

Special construction sites do not seem to have an impact on the accidents because almost all the accidents took place in places without any construction as shown in figure 11. This variable will be dropped.

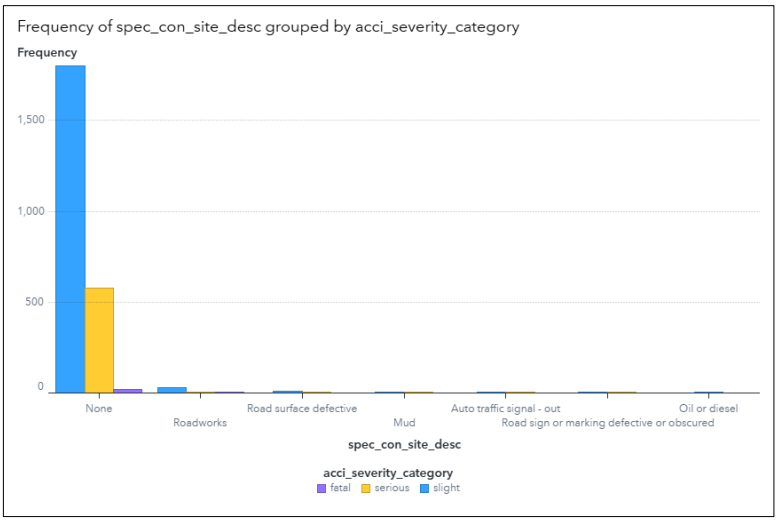


Fig. 18 Frequency of accidents in different construction sites

In figures 19 and 20 below, there are minor variations across different pedestrian crossing facilities and pedestrian crossing control measures with majority of the accidents taking place when there is no physical crossing and no control. Therefore, both these variables can be dropped.

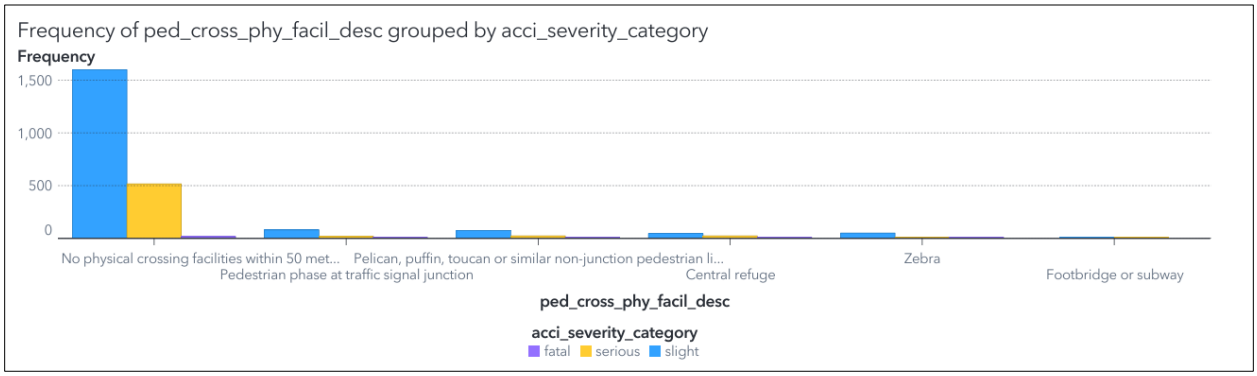


Fig. 19 Frequency of accidents in different pedestrian crossing facilities

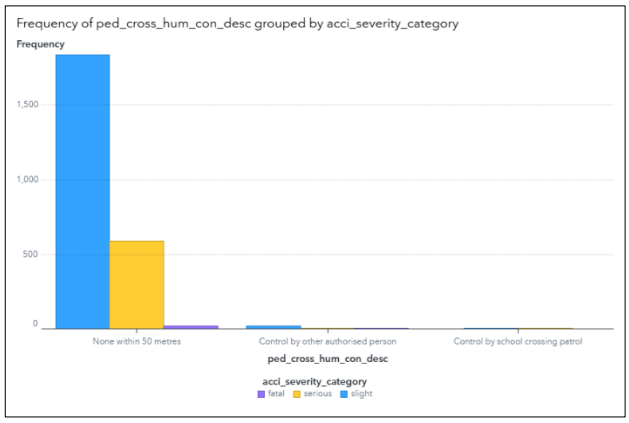


Fig. 20 Frequency of accidents in different pedestrian human control

Time Factors

There is stronger variation in the accident frequency across different weekdays (1-7 represents sun-sat) compared to the timing of the day as shown in figures 21 and 22. Also, the frequency of accidents is affected by the months. In the time series chart (figure 23) , accident occurrences drop between November and February; July and August whereas it increases between February to June and August till November. This is intuitive given these are seasonal holidays (less traffic congestion; no work, people travel). All three are deemed important and should be considered.

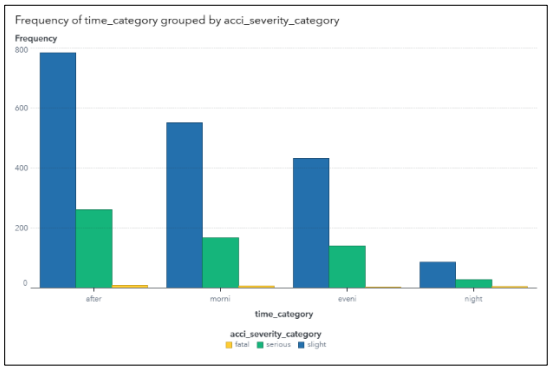


Fig. 21 Frequency of accidents in different times of the day

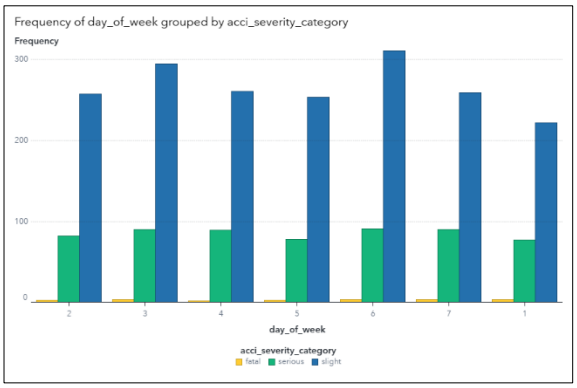


Fig. 22 Frequency of accidents across the week

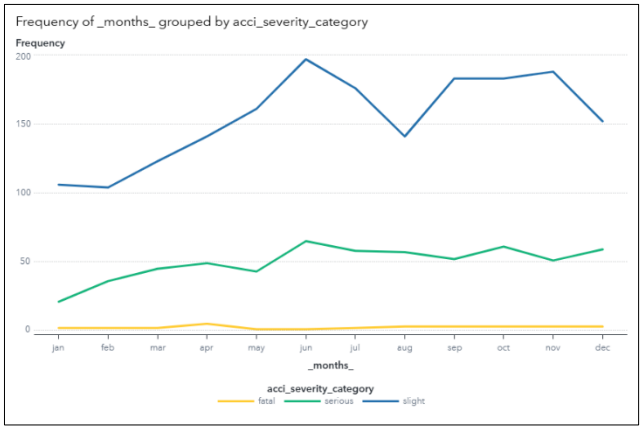


Fig. 23 Frequency of accidents across months

Accident Circumstances

Variables include number of casualties, number of vehicles, and speed limit. Given the numeric nature, dispersion of data can be analyzed across severity levels. Nevertheless, better insights can be made if data were balanced. In figure 24, half of fatal accidents involve more than one causality whereas other levels involve one. Moving on to speed limits in figure 25, top 25% of fatal accidents occurred when speed limits were 60 to 70 mph. and half were linked to higher speed ranging from 30 to 60mph compared to serious and slight where average was 35mph. Regarding the number of vehicles, top 25% of fatal accidents involve 2 to 5 vehicles; serious and slights accidents range between 2 to 8 as shown in figure 26. This is intuitive since fatal accidents involve a single high-speeding vehicle. For severity modeling, we will focus on number of vehicles and speed limit, excluding the number of causalities since by notion it is an outcome rather than a predictor. Did_police_attend is a problematic variable because in a sense it could be an outcome of severity; it is correlated to severity due to reporting rather than a causal predictor. Including it might result in bias and overfitting; therefore, it will be dropped**.**

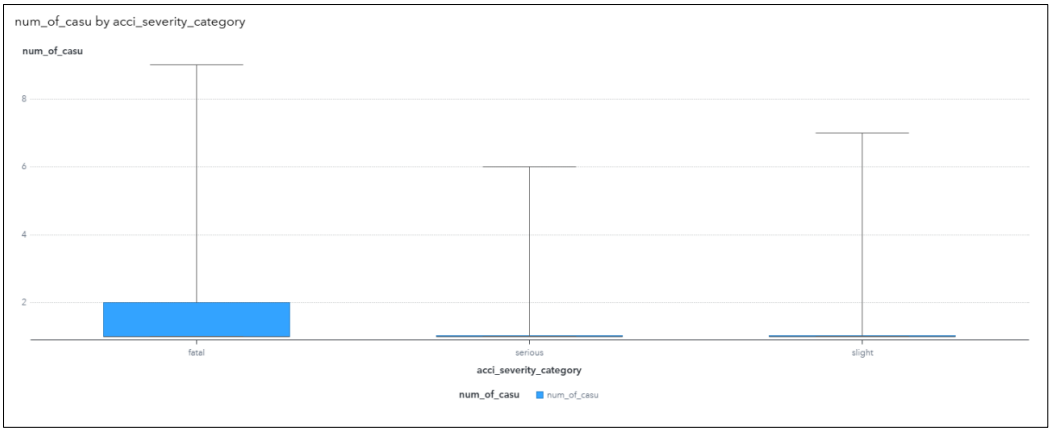


Fig. 24 Number of Casualties Box Plot

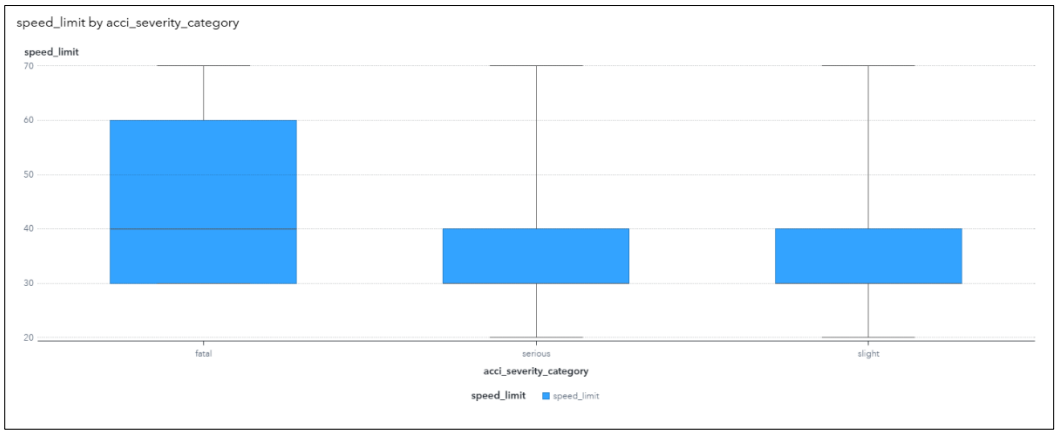


Fig. 25 Speed Limit Box Plot

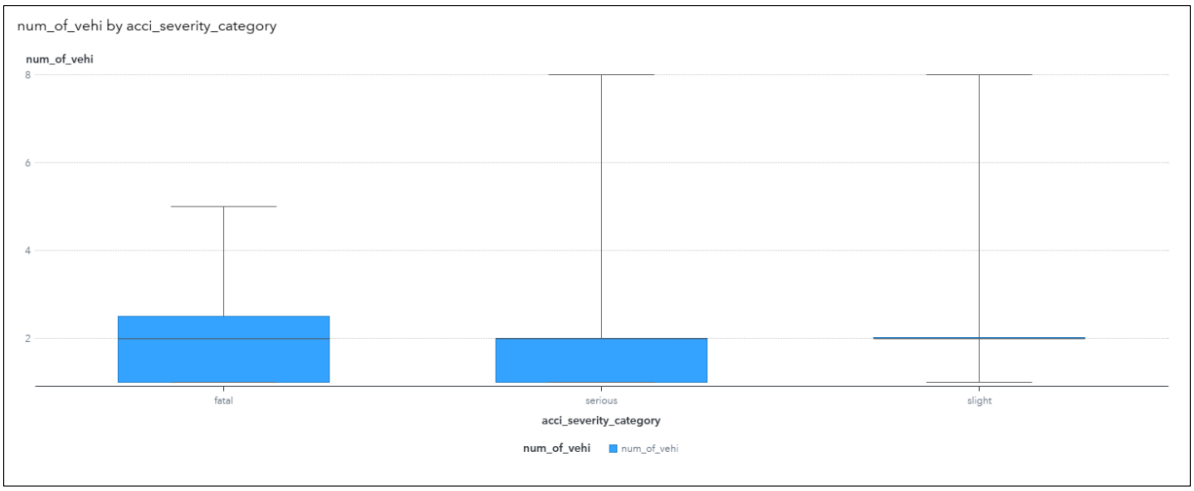


Fig. 26 Number of Vehicles Box Plot

Location Variables

The dataset contains many variable describing locations. Latitude and longitude are precise coordinates allowing us to infer many geographic details. The LSOA variable is less precise as many accidents occur within the same code, similarly, loc_auth_ons is a code for office of national statistics administrative district where accident happened; these areas are even broader compared to LSOA. Both LSOA and loc_auth_ons can be derived from latitude and longitude and add redundancy; hence they will be dropped. Additionally, northing and easting are alternative coordinate systems and are unnecessary thus will be dropped. As shown in figure 27, mapping coordinates show clear accident clusters highlighting the criticality of Lat. and Long. in predicting severity though data is unbalanced.

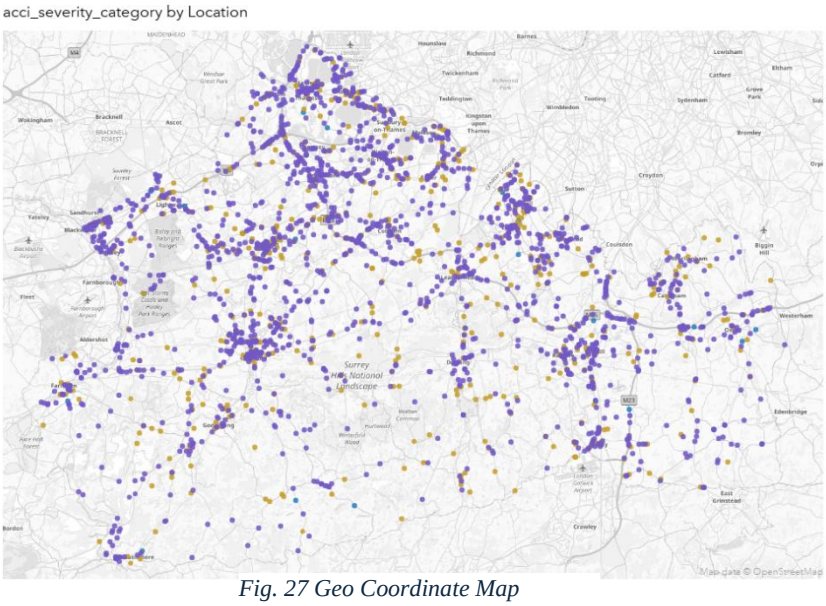


Fig. 27 Geo Coordinate Map

Based on the above analysis, the below variables are identified as potential predictors. Further refinement will be conducted to select the top 10 to11 variables.

Potential ML Predictors based on EDA			
Road and environmental variables	**Time Variables**	**Accident Circumstances**	**Location Variables**
Weath_con	day_of_week	num_of_vehi	longitude
road_type	time_category	speed_limit	latitude
road_surf_con	_months_		
junc_detail			
light_con			
ped_cross_hum_con_desc			
tru_road_flag			
ped_cross_phy_facil_desc			
Urban_or_rural			
First_road_class			
Second_road_class			

Table 2 Potential ML Predictors based on EDA

Variabes Refinement

Given that correlation matrices are unsuitable, AutoML (SAS Viya) and logical reasoning were combined to shortlist predictors instead. AutoML's output (see Figure 28) was cautiously interpreted due to data imbalance. Although did_poli_att was highly ranked by Auto ML, it was excluded due to bias. Conversely, light_con was included despite its low ranking as it is intuitively important and supported by findings from Ma et.al (2020). The time_category was dropped to avoid redundancy with light_con. ***road_surf_con*** , ***Second_road_class*** , ***ped_cross_hum_con_desc***, ***ped_cross_phy_facil_desc*** , ***Urban_or_rural***, , *and* ***tru_road_flag*** were dropped due to low importance as per AutoML results.

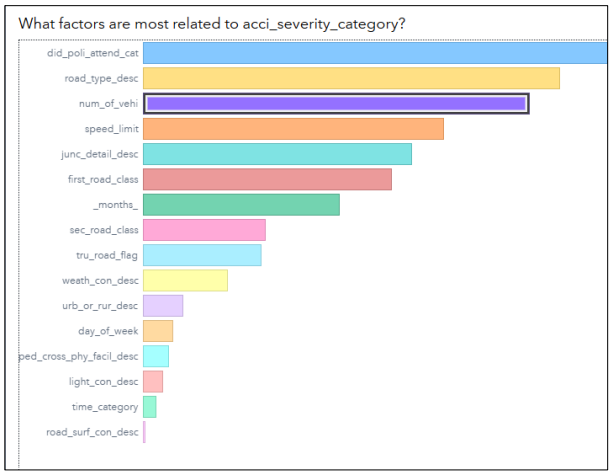


Fig. 28 AutoML Output

The final selection below aligns with ma et al. (2020) which identified geographical location, road type, and daytime conditions as critical predictors for accident severity.

Final Variables Selection			
Road and environmental variables	**Time Variables**	**Accident Circumstances**	**Location Variables**
Weath_con	day_of_week	num_of_vehi	longitude
road_type	_months_	speed_limit	latitude
junc_detail			
light_con			
First_road_class			

Table 3 Final Variables Selection as Predictors

Task 2: *Predicting Accident Severity*

Stage 1 Data Balancing

SAS Develop Flow and Code Flow : **Task 2*_Predicting Accident Severity_Cleaned_Road_Accident_2021__Data_Balancing(Stage1) SAS Program.pdf***
Data after balancing = ***ROAD_ACCIDENT_BALANCED.csv***

In this stage, to overcome the data imbalance issue, the dataset has been balanced using the provided code (a snippet is shown in figure 29). The balancing method in this code uses PROC SURVEYSELECT to oversample minority classes ("fatal") via bootstrap sampling with replacement.

Code

```
%let NumSamples1 = 70;
%let NumSamples2 = 3;
%let NumSamples3 = 1;
/* Sort the dataset by acci_severity */
proc sort data=WORK.IMPORT;
by acci_severity;
run;
/* Use PROC SURVEYSELECT to make the dataset balanced */
proc surveyselect data=WORK.IMPORT NOPRINT out=BalancedData1
method=urs
seed=12345
samprate=(1 0 0); /* Adjust the samprate to balance the strata */
strata acci_severity;
run;
/* Create a variable with a constant value for each observation */
data BalancedData_with_constant1;
set BalancedData1;
constant = 1;
run;
/* Use PROC SURVEYSELECT to perform bootstrap sampling based on
acci_severity */
proc surveyselect data=BalancedData_with_constant1 NOPRINT seed=1
method=urs
samprate=1
OUTHITS
reps=&NumSamples1(repname=row)
out=BootSamp1;
strata acci_severity;
```

Fig. 29 Snippet from Data Balancing Bootstrap Code

Stage 2 Comparative Analysis of Predictive Models

Details about all pipelines using Build App Model = ***Task 2_Predictive Models (All models combined).pdf***

Multiple machine learning models were developed and evaluated to identify the optimum one for predicting accident severity. Final variable predictors were selected based on EDA discussed earlier which highlighted a scenario where fatal accidents might take place in T or staggered junctions on single/dual carriageway roads in the non-holiday season due to a high-speeding vehicle. For all the models, dataset is partitioned as 70% training and 30% validation with fatal as the “target level event”.

Logistic Regression (LR)

LR's pipeline in figure 30 starts with transformation nodes that perform binning to continuous or high-cardinality categorical variables into discrete levels followed by WoE encoding which creates a linear relationship with log-odds. The transformed dataset is then passed to the LR model.

Fig. 30 Logistic Regression Pipeline

Model shows no ability to distinguish between severity classes, as evidenced by KS score of zero and a straight ROC curve (figure 32). Sensitivity increases at the same rate as specificity implying random guessing of model. Failure of this model could be attributed to the bootstrapping technique where duplicates of minority class are created causing LR to overfit and predict same probability for all observations. Additionally, LR assumes linearity between predictors and target; however, severity involves non-linear complex interactions that regression cannot capture.

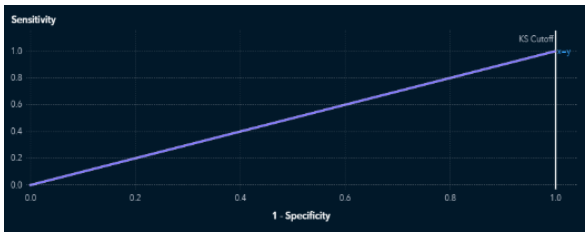


Fig. 32 Logistic Regression ROC

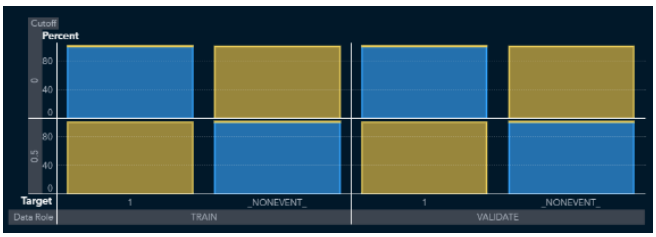


Fig. 31 Logistic Regression Event Classification

Neural Network

The model's pipeline is shown in figure 33 and it starts with a Variable Selection node that selected all variables and excluded only longitude.

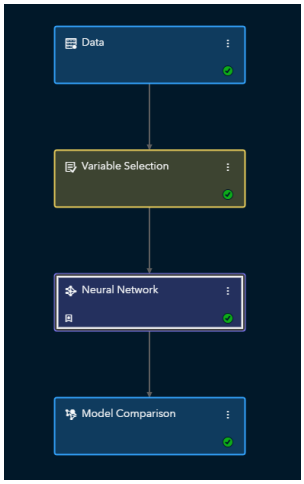


Fig. 33 Neural Network Pipeline

The model initially overfit due to balancing technique as it memorized the created duplicates (including noise) instead of general patterns resulting in a 90-degree ROC curve not only for training but also validation set which was drawn from the same duplicated data; a classic data leakage issue.

After some hyperparameter tuning (seen in table 4), performance slightly improved as shown below but metrics remain suspiciously strong (KS=88.7%, AUC = 95.38%) making the model high-risk for practical use.

Hyperparameters After Tuning	
Number of neurons per hidden layer	10
Hidden layer activation function	tanh
Number of hidden layers	1

Table 4 Neural Network Hyperparameters After Tuning

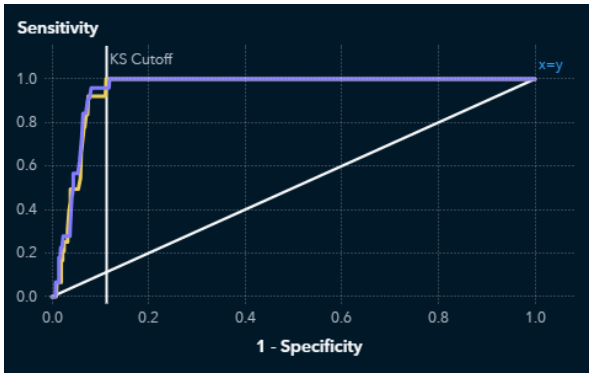


Fig. 35 Neural Network ROC Curve

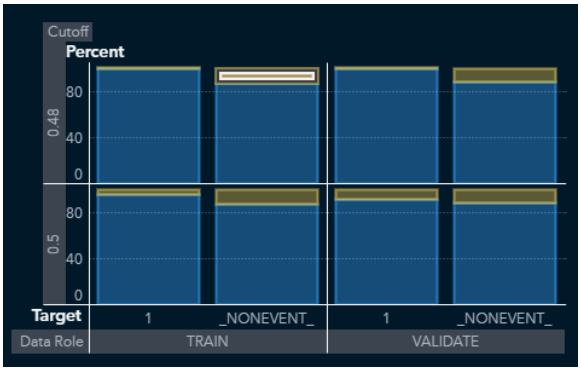


Fig. 34 Neural Network Event Classification

Neural Network Fit Statistics								
Target Name	**Data Role**	**Number of Observations**	**Average Squared Error**	**Divisor for ASE**	**Root Average Squared Error**	**KS (Youden)**	**Area Under ROC**	**KS Cutoff**
acci_severity	TRAIN	2,464	0.1409	2,464	0.3754	0.8805	0.9538	0.4800
acci_severity	VALIDATE	1,055	0.1450	1,055	0.3807	0.8877	0.9504	0.4800

Table 5 Neural Network Fit Statistics

Decision Trees

The third model being evaluated is the decision tree. After hyperparameter tuning; below are the values that resulted in the most optimum results:

Decision Tree Hyperparameters After Tuning	
Class Target Criterion	Information Gain Ratio
Max Number of branches	2
Max depth	15
Min leaf size	7

Table 6 Decision Tree Hyperparameters After Tuning

This model shows strong discriminatory power to separate between severity classes with a KS score of 72.5%. and an AUC of 89.66% indicating an excellent ability to rank fatal cases over non-fatal ones. The average square error is also low (0.1517) showing that predicted probabilities closely match actual values. This model outperforms the others because it naturally handles non-linear relationships and models complex spatial data without manual feature engineering which is crucial because of the latitude and longitude variables included. More importantly, the

algorithm is robust to the bootstrapped data because the splits are based on feature distribution not weights.

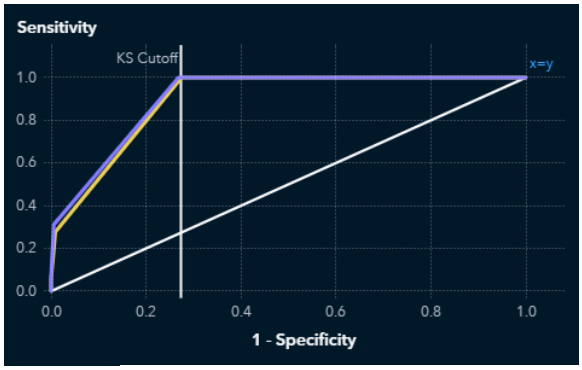


Fig. 36 Decision Tree ROC Curve

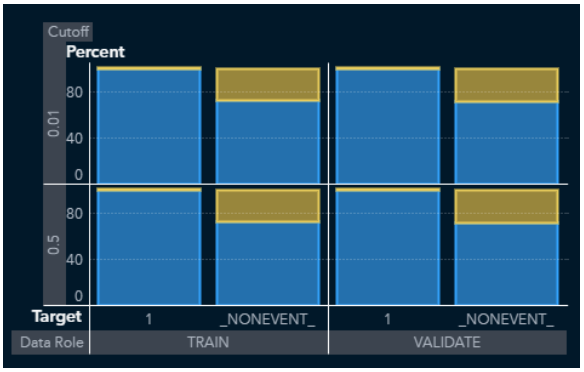


Fig. 37 Decision Tree Event Classification

Decision Tree Fit Statistics								
Target Name	**Data Role**	**Number of Observations**	**Average Squared Error**	**Divisor for ASE**	**Root Average Squared Error**	**KS (Youden)**	**Area Under ROC**	**KS Cutoff**
acci_severity	TRAIN	2,464	0.1282	2,464	0.3580	0.7320	0.9051	0.0100
acci_severity	VALIDATE	1,055	0.1517	1,055	0.3894	0.7253	0.8966	0.0100

Table 7 Decision Tree Fit Statistics

As shown below, pruning produced 68-leaf decision tree that balances accuracy and generalizability, though a 33.7 misspecification rate suggests room for improvement. To further enhance performance, a random forest model will be implemented next.

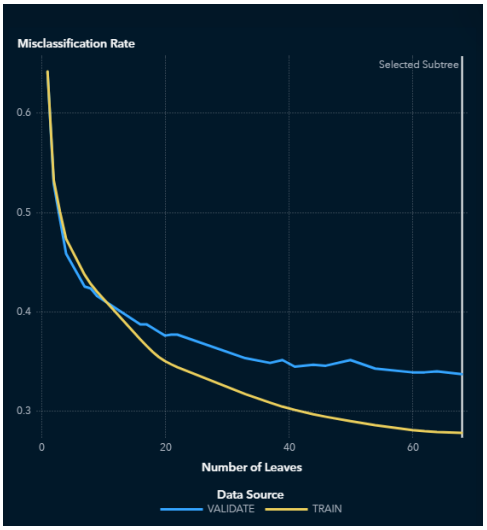


Fig. 38 Decision Tree Pruning Error Plot

Random Forest

Random forest, an ensemble of decision trees, has an outstanding performance relative to prior models (as shown in below figures) achieving a KS score of 83% indicating its strength in separating different classes. Morevoer, it is highly reliable for prioritizing fatal accidents as evidenced by its AUC score of 96%. Its F1 score is around 0.774 demonstrating a good balance between recall and precision implying model will reliably identify true positives (fatal accidents) without overwhelming users with false alarms.

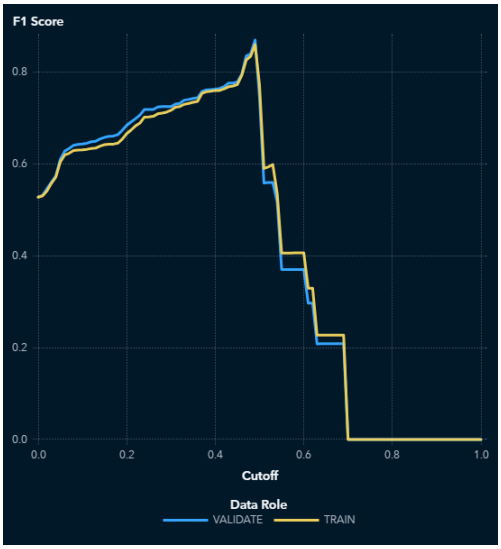


Fig. 40 Random Forest F1 Score

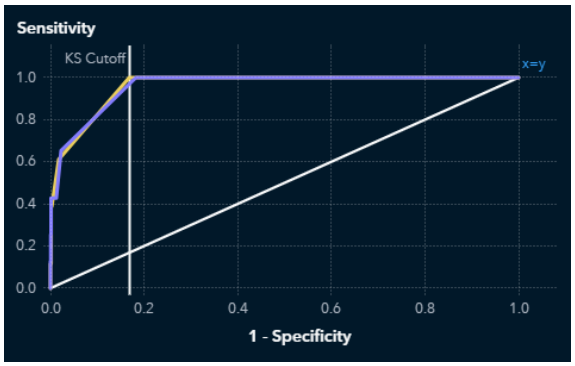


Fig. 39 Random Forest ROC Curve

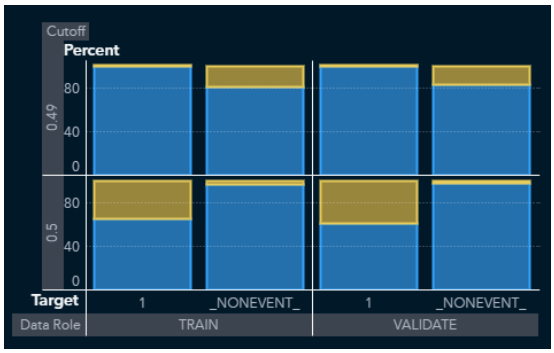


Fig. 41 Random Forest Event Classification

Random Forest Fit Statistics								
Target Name	**Data Role**	**Number of Observations**	**Average Squared Error**	**Divisor for ASE**	**Root Average Squared Error**	**KS (Youden)**	**Area Under ROC**	**KS Cutoff**
acci_severity	**TRAIN**	**2,464**	**0.1632**	**2,464**	**0.4040**	**0.8173**	**0.9601**	**0.4900**
acci_severity	**VALIDATE**	**1,055**	**0.1676**	**1,055**	**0.4094**	**0.8316**	**0.9611**	**0.4900**

Table 8 Random Forest Fit Statistics

Figure 42 below displays the importance of each feature used in the model. As expected, the most influential variables include months, location, and junction detail along with weather and road conditions.

Role	Variable Name	Relative Importance
INPUT	_months_	1
INPUT	latitude	0.3816
INPUT	longitude	0.3060
INPUT	junc_detail_d	0.2684
INPUT	speed_limit	0.1950
INPUT	weath_con_d	0.1576
INPUT	road_type_d	0.1056
INPUT	num_of_vehi	0.0933
INPUT	day_of_week	0.0927
INPUT	light_con_d	0.0572
INPUT	first_road_class	0.0226

Fig. 42 Random Features Variables Importance

Models Comparison, Conclusion, and Recommendations

Based on the below results and previous discussions, RF emerges as the champion model because of its robustness to bootstrapping and highest KS score; in contrast to neural networks where the extremely high KS score was due to overfitting. Additionally, RF effectively captures non-linear and spatial relationships that logistic regression failed to do.

Algorithm Name	Pipeline Name	KS (Youden)	Number of Observations
Neural Network	neural network	0.8877	1,055
Forest	Random Forest	0.8316	1,055
Decision Tree	Decision Tree	0.7253	1,055
Logistic Regression	Logisitic Regression	0.0000	1,055

Fig. 42 All Models Fit Statistics Comparison

The table below highlights the strengths and weaknesses of each of the models examined.

Models Comparison		
Model	**Strengths**	**Weaknesses in this context**
Logistic Regression	Simple and interpretable	Linear assumption failed for complex interaction
Neural Network	Captures complex patterns	Overfits for the medium sized bootstrapped by replacement dataset (data leakage issue)
Decision Tree	Handles non-linear and spatial relationships, robust to bootstrapping data balancing	Strong model; KS and misspecification rate could be enhanced using ensemble of decision trees
Random Forest	Encapsulates strengths of decision tree with a higher KS score	Computationally heavier than the decision tree, the misspecification rate did not improve compared to Decision Tree (could be due to the relatively small size of the dataset)

Table 9 Predictive Models Comparison

Below are the most important features that were common between all models. Based on this, such predictive models could be used to allocate resources such as speed cameras, signage, and patrols based on high-risk months (especially non-holiday period) , junction types, and speeding patterns. Safety campaigns and road improvements should also be focused on single and dual carriageway roads.

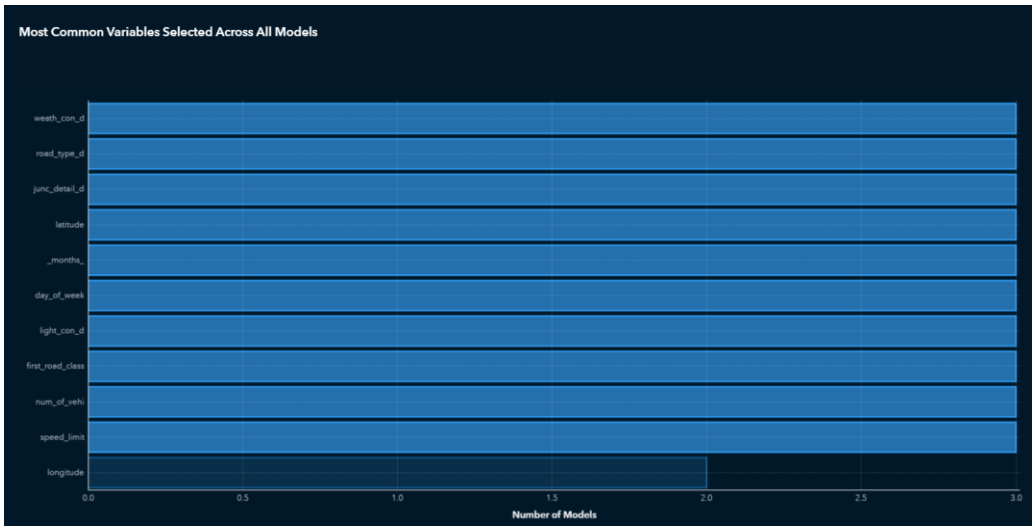


Fig. 43 Most Common Variables Selected Across All Models

Task 3: *Text Analysis of Tweets*

Details about the pipeline for all nodes: ***Task 3_Text Analytics All Nodes Pipeline (including Sentiment Analysis Code).pdf***

Details about exploration text analysis using Explore and Visualize app: ***Task 3_Tweets Road Accident Exploratory (Explore and Visualize)***

This task focuses on analyzing a dataset consisting of Tweets about road traffic accidents in Surrey using text mining. The data contains 598 observations under the variable "Text"; a new column is generated that assigns a unique ID to each tweet that acts as a key reference before uploading to SAS Viya.

The figure below shows the pipeline for this project.

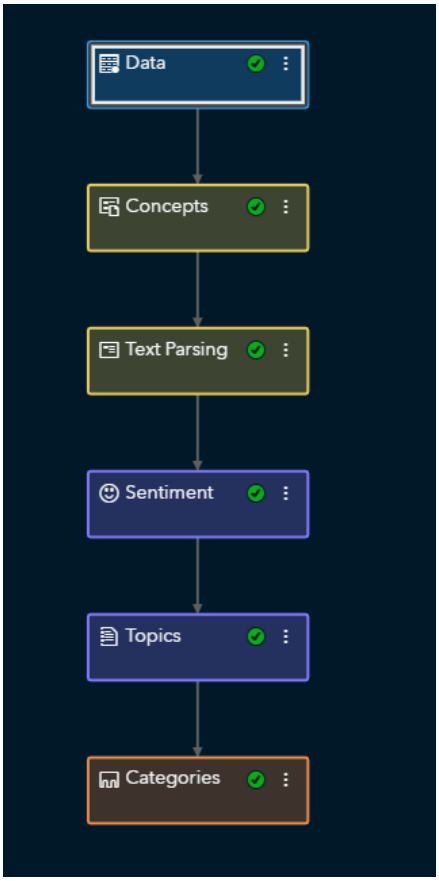


Fig. 44 Text Analytics Pipeline

Concepts and Exploratory Analysis

After a brief overview of the excel file and identifying key variables from predictive model, a set of custom concepts have been defined as shown below using regex expressions. Classifiers proved inefficient due to imprecise nature of dataset's text; making regex a more effective approach.

Custom Concepts Definition	
Concept Name	**Regex Expression**
surrey_roads	REGEX:(?:[AMBC]\d+)
surrey_junctions	REGEX:(?:Junction|Juntion|J)[\s-]*(?:\d+)
vehicle_type	REGEX:(?:car|lorry|truck|van|bus|motorcycle|bicycle|taxi|emergency\svehicle)
months	REGEX:(?:January|February|March|April|May|June|July|August|September|October|November|December)
severity_terms	REGEX:(?:fatal|serious|slight)\s(?:injury|injuries|collision)
day_of_week	REGEX:(?:Monday|Tuesday|Wednesday|Thursday|Friday|Saturday|Sunday|weekend|weekday)
time_of_day	REGEX:(?:morning|afternoon|evening|night|rush\s*hour|midnight|noon|overnight)
accident_types	REGEX: (?:head-on|rear-end|pedestrian|cyclist)
weather_conditions	REGEX:(?:fog|rain|snow|ice|icy|slippery|wet\spavement|hail|storm|flood)

Table 10 Custom Concepts Definition

Figures 45 and 46 show the frequency of the predefined and custom concepts. The custom concepts with the highest number of matches are surrey_roads, surrey_junctions, vehicle_type, and months. People have been mentioning specific surrey roads and junctions and this aligns with how these two variabels are crucial to the predicve models explored earlier. Vehicle_types being the third top concept is extermely insightful because the primary dataset lacked any information regarding the vehicles involved. Morover, the word cloud in figure 47 shows the frequency of the concepts' keywords and there is a huge emphasis on roads and junctions.

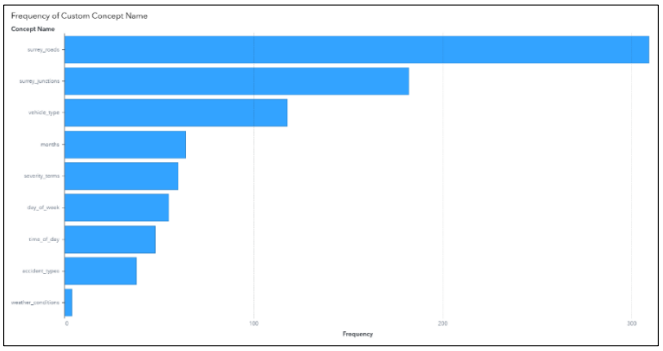


Fig. 45 Frequency of Custom Concepts

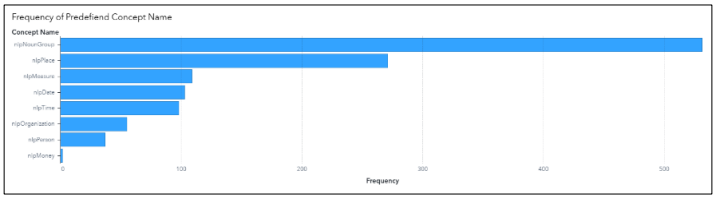


Fig. 46 Frequency of Predefined Concepts

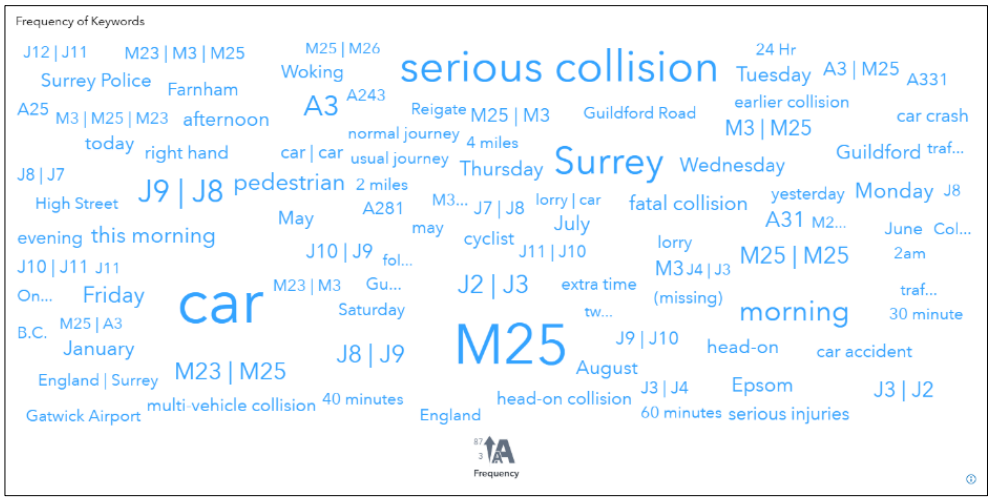


Fig. 47 Word Cloud for Concepts' Keywords

Text Parsing

Frequency is key for deciding which terms to keep or drop for meaningful impact. Although identifying vehicle types involved is crucial, some of the terms appeared only once; hence the model dropped them. Similarly, date/months did not provide any insights due to low frequency in documents.

Focusing on “severity_terms” concept, a tree map shows that “serious collision” which amounts to high frequency, is strongly associated with two roads: A3 and A331. Additionally, terms indicating “overnight” and “late” imply that serous collisions resulting in injuries take place in A3 and A331 at night as demonstrated by figure 48. This is further supported by the terms maps for roads A3 and A331 where serious collisions and overnight are strongly associated (figures 49 and 50)

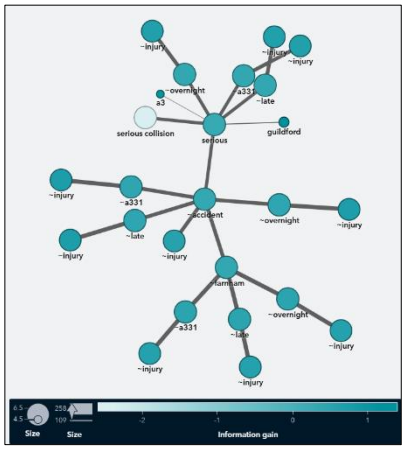


Fig. 50 "Serious Collision" Term Map

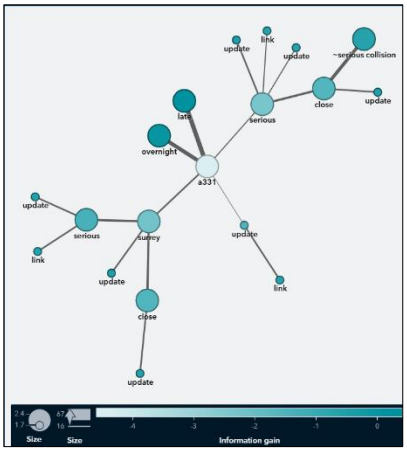


Fig. 49 "A3" Term Map

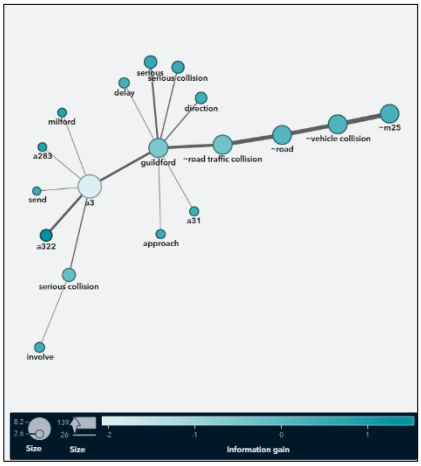


Fig. 48 "A331" Term Map

Another interesting finding is junction 9 tree map; it is strongly associated with Leatherhead town which is strongly associated with collisions. This suggests that people heading to Leatherhead through junction 9 are more prone to collisions/accidents.

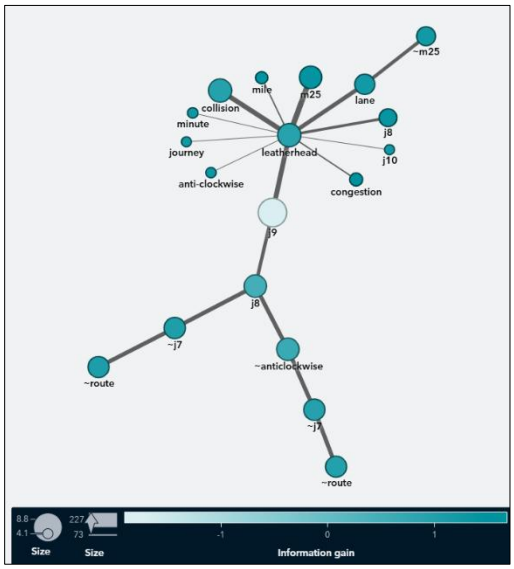


Fig. 51 "Junction 9" Term Map

Sentiment Analysis

Details about SAS code using SAS Program: ***Text Analytics All Nodes Pipeline (including Sentiment Analysis Code).pdf***

Following text parsing, the sentiment analysis node analyzed the text and generated a score code that reflects the emotional tone of the tweets. After extracting the score code and adding it to a SAS program, a table was generated that was visualized as shown below. There are three classes: negative, positive, and neutral. The dominating sentiment is "negative" accounting for 552 tweets which is almost 92% of the data.

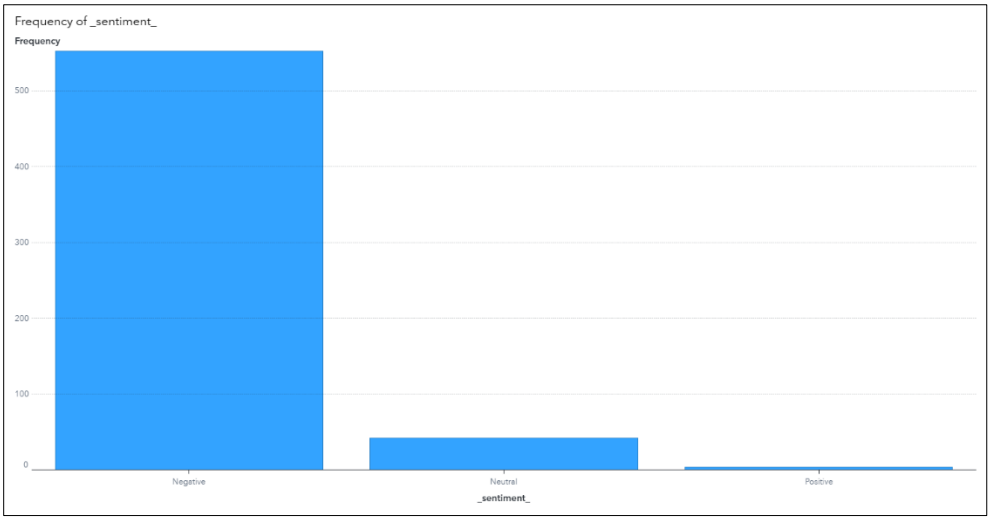


Fig. 52 Sentiment Analysis

Topic Modelling

SAS automatically generated 25 topics, from which 4 user-defined topics were created by merging related SAS-generated topics. The table below is a detailed description of the top 4 important topics (mix of autogenerated and user created) that will be promoted as categories. All the remaining topics can be seen in figure 53.

Topic Modelling			
Topic Given Name	**System Generated/User Created through merging**	**Raw SAS generated topics merged**	**Notes**
A3 and A331 Collisions	User Created	• guildford, a3, serious collision, +avoid, +direction • serious, surrey, serious collision, +late, a331	2 of the system genertaed topics support the earlier deductions that were made from the term maps. They are merged to encapsulate serious collision accidents on roads A3 and A331 happening at night.
Accident Hotspots: M25 J8–J9 (Leatherhead Area)	System Generated	• m25, leatherhead, j9, j10, accident	
"Incident-Induced Traffic Delays and Closures	User Created	• +mile, congestion, anti-clockwise,time,approx • usual, usual journey, +long, +journey, at least • +time, allow, extra, +please, extra time,service • rd,avenue,st,+avoid	Another common topic is the congestion and delays due to collisions. Even though these are consequences of collisions rather than factors, the Tweets are dense with this information, and it could be relevant in devising faster emergency handling methods to free up the accident location and let traffic through. Congestion and delays have drastic impacts on the productivity of a country in general
Farnham Head-On Collision	User Created	• farnham, a31, road, pedestrian, farnham road • head-on collision, head-on, farnham, +injure, woman	

Table 11 Summary of Topic Modelling

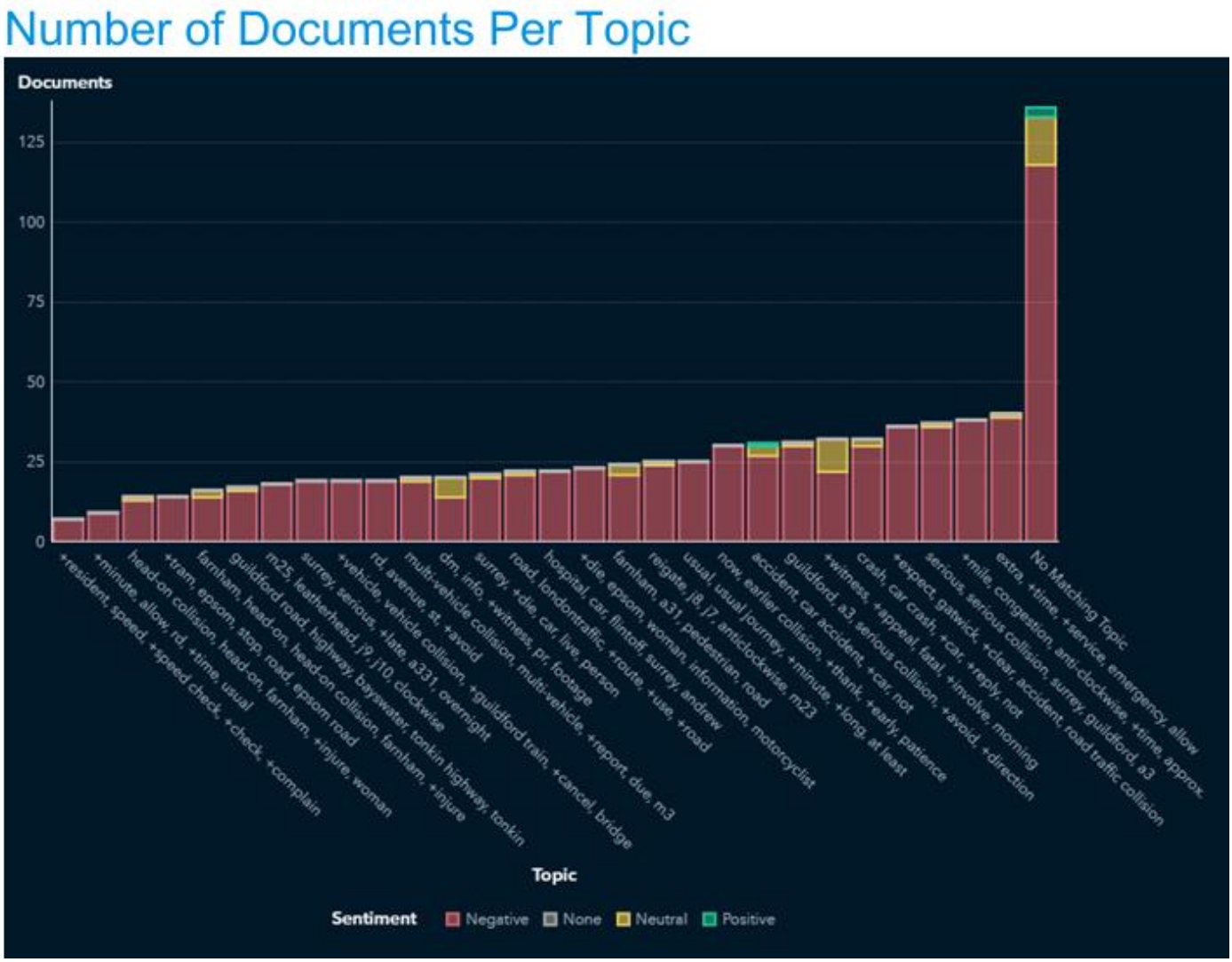


Fig. 53 Topic Modelling

Categories

Figure 54 shows that overall, the topics promoted to categories perform well given that large portion is true positive with smaller percentages relatively false negative or false positive. Further room for enhancement can be made to improve the count of false positives in the “A3 and A331 Collisions” by adding more specific operators to narrow the scope of rule definition to reduce the number of incorrect matches.

Regarding the metrics, most of the categories are performing well with an average F1 score of 0.77 which shows a good balance between recall and precision. The best performing category is "A3 and A331 Collisions" while the worst is "incident induced traffic delays and closure".

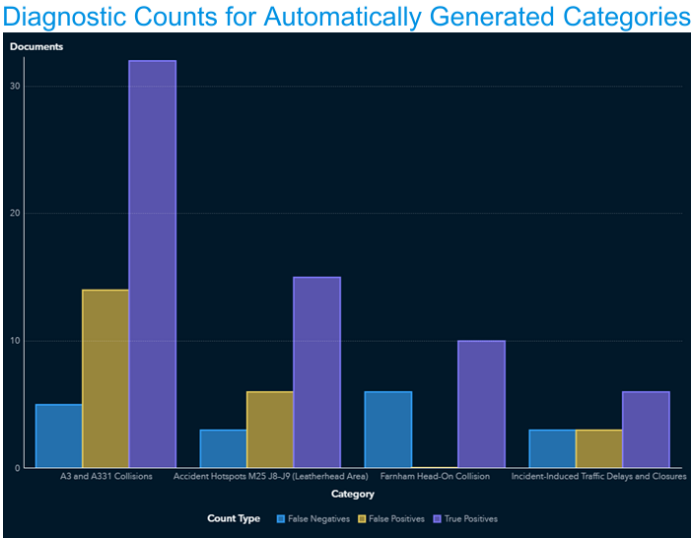


Fig. 54 Diagnostic Counts for Categories

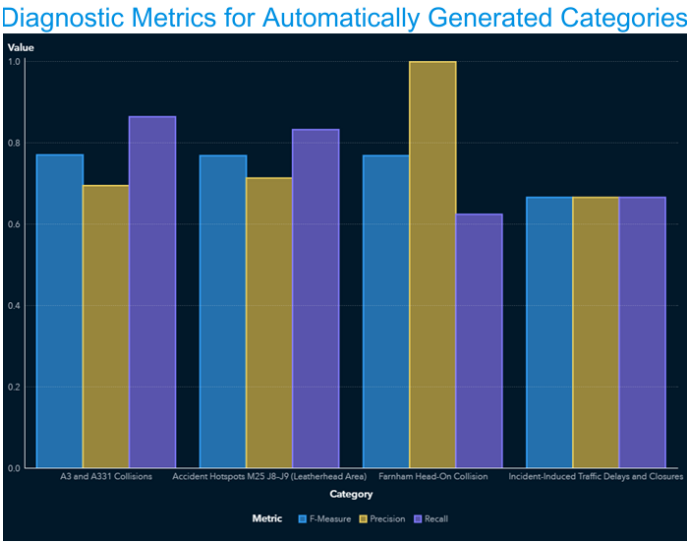


Fig. 55 Diagnostic Metrics for Categories

Key Insights

Twitter users frequently posted about specific Surrey roads and junctions reinforcing the predictive model's findings that location, road types and junctions are critical factors in accident severity. Notably, vehicle types were often mentioned as well adding valuable information that was missing from the primary dataset. Patterns revealed that roads A3 and A331 are associated with serious collisions particularly at night. Additionally, junction 9, Leatherhead, and Farnham are accident hotspots. Topic analysis revealed not only recurring themes related to junctions and roads but also reports of traffic delays and road closures due to accidents. Such insights can be leveraged to devise strategies for accident hotspot areas that suffer greatly from traffic halts.

References List

Ma, Z., Mei, G., & Cuomo, S. (2020). An analytic framework using deep learning for prediction of traffic accident injury severity based on contributing factors. Accident Analysis & Prevention, 146, 105711. https://doi.org/10.1016/j.aap.2020.105711

2999 words
(excluding figures, tables, and headings)

Decision-Maker's Summary and Recommendations

Based on

Exploring Road Traffic Accident Data and Text Analytics

Task 4: *Decision-Maker's Summary and Recommendations*

This report presents key findings from a comprehensive analysis of road traffic accidents in Surrey from 2021 and social media data. The objective is to support data-driven decisions for improving road safety.

1. **About the Dataset**
   - The primary dataset includes detailed records of road collisions in Surrey in 2021, capturing information related to road and environment, time, accident circumstances, location, and accident severity.
   - A secondary dataset of tweets related to accidents in Surrey was analyzed to support the findings and address potential gaps in the primary data mining project.

2. **Key Insights from Exploratory Data Analysis**

   Accident frequency varies noticeably throughout the year demonstrating clear seasonal trends. It decreases from November to January, rises until June, declines in July and August, and increases again into November. These fluctuations correspond with holiday periods, reflecting changes in traffic and commuting patterns.

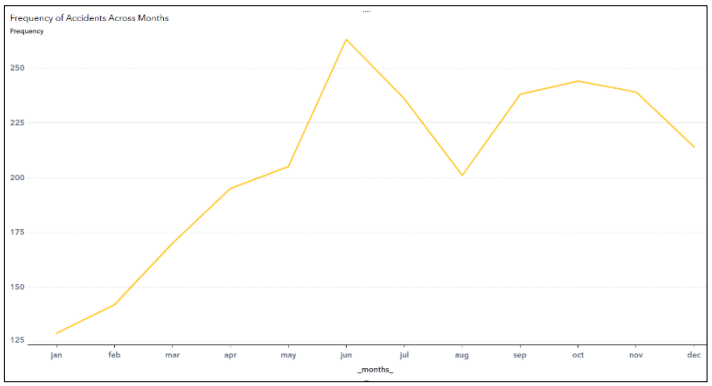


Figure 1 Accidents Across Months

Junctions and road types are significant factors in accident occurrence. Most of the incidents are reported on single and dual carriageway roads. Notably, the absence of junctions and presence of T or staggered junctions are associated with higher accident rates. In contrast, slip roads and mini-roundabout experience very few accidents.

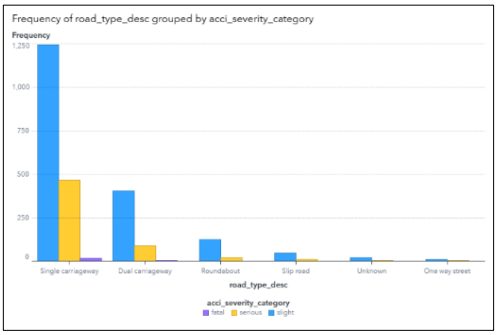


Figure 3 Accidents Across Road Types

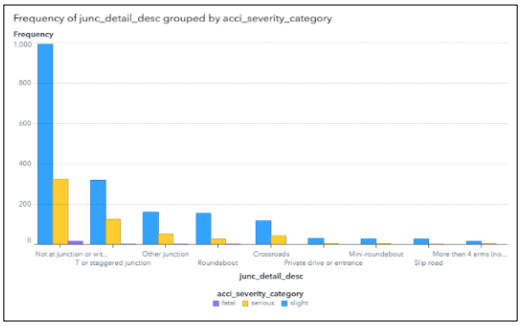


Figure 2 Accidents Across Junction Types

3. **Key Insights from Predictive Model**

   Four models were tested – logistic regression, decision trees, neural networks, and random forest (RF) to predict accident severity. RF was the best performer given its robust algorithm with the dataset

involved and strong discriminatory power to classify severity as fatal/serious/or slight. Months, location, junction details and speed are top risk factors and important variables in determining severity as supported by the exploratory data analysis.

4. Insights from Social Media (Text Analytics)

- *Relevant Mentions:* Twitter users frequently referred to specific Surrey roads, junctions, and vehicle types. This complements the earlier findings of the predictive model that location, junction and road types are critical factors in accident severity.
- *Unique Value:* Vehicle types were often mentioned, filling a gap left by the primary dataset, which lacked this piece of information.
- *Accident Patterns:* Roads A3 and A331 are strongly associated with serious collisions, particularly late hours. Additionally, Junction 9 and the Leatherhead area were notably referenced as accident hotspots.
- *Topic Analysis:* The common themes included major incidents on A3/A331, collisions around M25 Junctions 8-9, and recurrent reports of traffic delays and roads closures due to accidents.

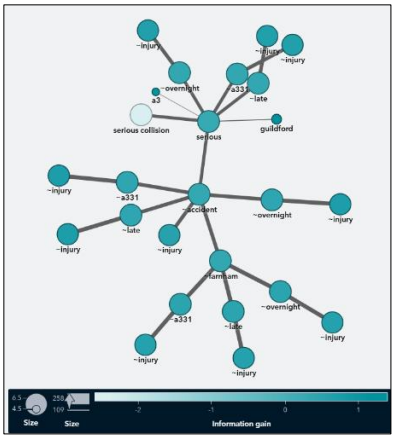


Figure 5 "Serious Collision" Term Map

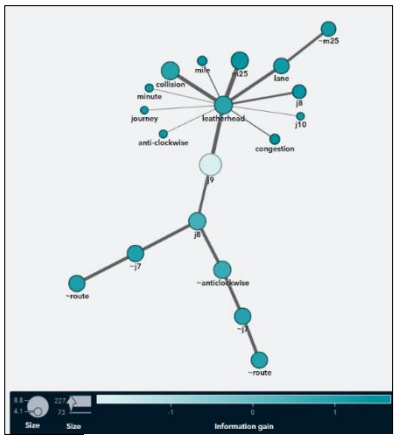


Figure 4 "Junction 9" Term Map

Targeted Recommendations Based on the Findings

- Focus engineering upgrades and targeted patrols in high-risk roads and junctions particularly in A3 and A331 roads; as well as M25 junctions 8-9 heading to Leatherhead especially during late hours. Implementing traffic signal control at Junction 9 is also strongly recommended given it is currently not signalized
- Use predictive model to allocate resources such as speed cameras, signage, and patrols based on high-risk months (especially non-holiday period) , junction types, and speeding pattern.
- Focus safety campaigns and make road improvements on single and dual carriageway roads
- Improve future data collection by including driver details (e.g. occupation, gender), crash types, detailed vehicle info as this will enhance the performance of the predictive model even further

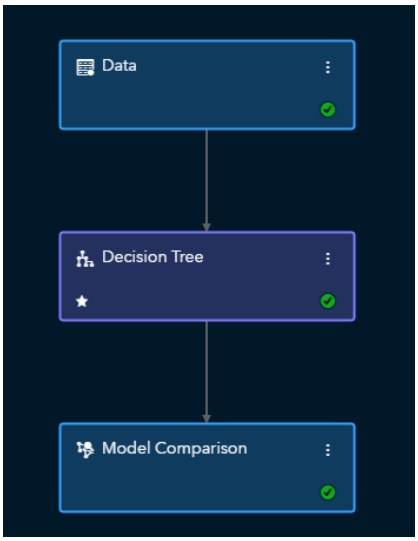
Appendix

acci_severity_category	Minimum	Lower Whisker	First Quartile	Average	Median	Third Quartile	Upper Whisker	Maximum	Std Dev	Count
fatal	1	1	1	1.70833	1	2	9	9	1.70623	24
serious	1	1	1	1.32998	1	1	6	6	0.7685	597
slight	1	1	1	1.25822	1	1	7	7	0.645	1,855

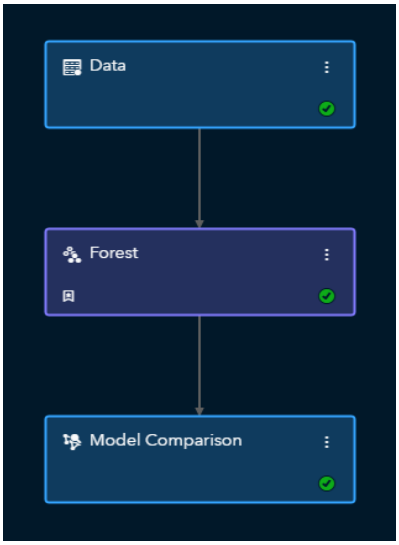
Appendix Fig 1 Number of Causality Distribution

acci_severity_category	Minimum	Lower Whisker	First Quartile	Average	Median	Third Quartile	Upper Whisker	Maximum	Std Dev	Count
fatal	30	30	30	43.75	40	60	70	70	15.2693	24
serious	20	20	30	38.3082	30	40	70	70	13.0815	597
slight	20	20	30	38.9704	30	40	70	70	14.1103	1,855

Appendix Fig 3 Speed Limit Distribution



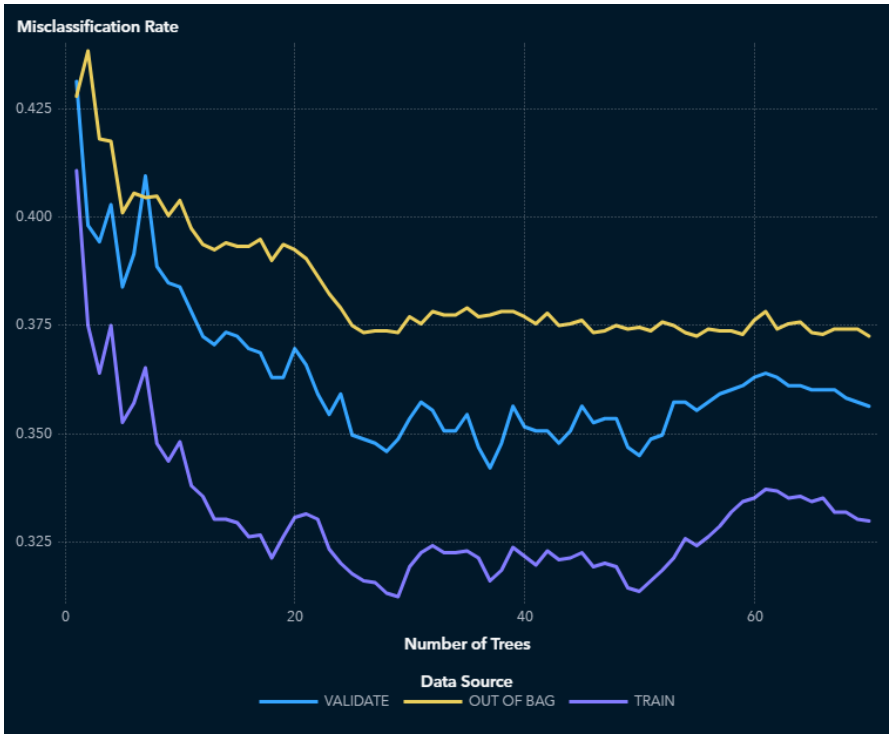
Appendix Fig 4 Decision Tree Pipeline



Appendix Fig 5 Randon Forest Pipeline

Variable Name	Training Importance	Training Relative Importance	Validation Rel...	Count	↓ Validation Import...
months	243.5914	1	1	10	86.2291
speed_limit	49.9083	0.2049	0.3764	4	32.4535
latitude	111.2987	0.4569	0.3100	11	26.7304
junc_detail_d	57.1999	0.2348	0.2453	4	21.1547
road_type_d	30.3278	0.1245	0.1776	3	15.3166
day_of_week	55.9951	0.2299	0.1437	8	12.3884
longitude	62.0510	0.2547	0.1243	9	10.7155
weath_con_d	43.4392	0.1783	0.0799	6	6.8865
num_of_vehi	12.0205	0.0493	0.0477	4	4.1103
first_road_class	7.1313	0.0293	0.0369	3	3.1844
light_con_d	19.6281	0.0806	0.0355	5	3.0603

Appendix Fig 6 Decision Tree Important Variables



Appendix Fig 7 RF misspecification rate: rate did not improve compared to Decision Tree (could be due to the relatively small size of the dataset)

Concept Name	Frequency ▼
surrey_roads	309
surrey_junctions	182
vehicle_type	118
months	64
severity_terms	60
day_of_week	55
time_of_day	48
accident_types	38
weather_conditions	4

Appendix Fig 8 Frequency Tables for Custom Concepts

```
 /* Save the new dataset in the casuser library */
 %let use_training_only = 1;
%if &use_training_only = 1 %then %do;
data new_dataset;
   set &dm_data;
   if acci_severity not in (1 ,2 ,3) then delete; /* Remove rows where acci_severity not equal to 1, 2, or 3 'var_name' equals 'value' */
   if IMP_police_force ne 45 then IMP_police_force = 45; /* changes police force that was imputated to 45 */
   if loc_auth_highw ne "E10000030" then loc_auth_highw= "E10000030" ;/* replace missing variable with this value (cnstant value)
/* Remove rows where did_poli_att not equal to 1, 2, or 3 'var_name' equals 'value' */
   if did_poli_offi_att not in (1 ,2 ,3, -1) then delete;
 /* Remove rows where longitude, latitude, loc_east_osgr, or loc_nor_osgr are missing' */
   if cmiss(latitude,longitude,loc_nor_osgr,loc_east_osgr) > 0 then delete;
 /* Adds random date to emplty date field with the condition of keeping the year 2021 */
   if missing(date) then do; /* Replace 'date_var' with your variable name */
      /* Define start and end dates for 2021 */
      start_date = '01JAN2021'd; /* SAS numeric date value for 01/01/2021 */
      end_date = '31DEC2021'd;   /* SAS numeric date value for 12/31/2021 */
      /* Generate a random date between start_date and end_date */
      random_days = floor((end_date - start_date + 1) * rand('uniform')); /* Days between 0 and 364 */
      random_date_num = start_date + random_days; /* Numeric SAS date */
      /* Convert numeric date to character with mm/dd/yyyy format */
      date = put(random_date_num, mmddyy10.); /* Format as mm/dd/yyyy */
   end;
   /* Ensure the variable remains character type */
   format date $10.; /* Explicitly set length to match mmddyy10. format */
/* create new day_of_week variable that ensures all the day of week are accurate and actually represent the corresponding date because some blank
dates have a dummy day of the week */
day_of_week_Amended = weekday(input(date, mmddyy10.)); /* Sunday=1 to Saturday=7  */
/* dropping variables not needed or redundant */
drop Row random_date_num  random_days  end_date   start_date  day_of_week;
run;
%end;
/* Step 1: Calculate the mode for each column (excluding -1 values) */
proc freq data=new_dataset noprint;
   where carri_haz ne -1;
   tables carri_haz / out=carri_haz_mode;
run;
proc sort data=carri_haz_mode;
   by descending count;
run;
data _null_;
```

Appendix Fig 9 Data Cleaning SAS Code

```
data work.output_table;
set road_acci_table;
length _months_ $3;
length time 8;
format time time8.;
/* Convert the date variable to SAS date value */
sas_date = mdy(month(date), day(date), year(date));
/* Check month ranges and assign labels accordingly */
if ('01JAN2021'd <= sas_date <= '31JAN2021'd) then _months_= 'jan';
else if ('01FEB2021'd <= sas_date <= '28FEB2021'd) then _months_= 'feb';
else if ('01MAR2021'd <= sas_date <= '31MAR2021'd) then _months_ =
'mar';
else if ('01APR2021'd <= sas_date <= '30APR2021'd) then _months_= 'apr';
else if ('01MAY2021'd <= sas_date <= '31MAY2021'd) then _months_=
'may';
else if ('01JUN2021'd <= sas_date <= '30JUN2021'd) then _months_= 'jun';
else if ('01JUL2021'd <= sas_date <= '31JUL2021'd) then _months_= 'jul';
else if ('01AUG2021'd <= sas_date <= '31AUG2021'd) then _months_= 'aug';
else if ('01SEP2021'd <= sas_date <= '30SEP2021'd) then _months_= 'sep';
else if ('01OCT2021'd <= sas_date <= '31OCT2021'd) then _months_= 'oct';
else if ('01NOV2021'd <= sas_date <= '30NOV2021'd) then _months_= 'nov';
else if ('01DEC2021'd <= sas_date <= '31DEC2021'd) then _months_= 'dec';
else _months_= 'other';
format sas_date date9.; /* Optional: Set the format for display purposes */
/* Extract the hour from the time variable */
hour_of_day = hour(time);
/* Categorize time into different periods */
if 0 <= hour_of_day < 6 then time_category = 'night';
else if 6 <= hour_of_day < 12 then time_category = 'morning';
else if 12 <= hour_of_day < 18 then time_category = 'afternoon';
else if 18 <= hour_of_day <= 23 then time_category = 'evening';
else time_category = 'other';
run;
```

Appendix Fig 10 SAS Program Code in Task 1 Data Cleaning Stagge 2 Preprocessing Flow

```
%let NumSamples1 = 70;
%let NumSamples2 = 3;
%let NumSamples3 = 1;
proc sort data=WORK.IMPORT;
by acci_severity;
run;
proc surveyselect data=WORK.IMPORT NOPRINT out=BalancedData1
method=urs
seed=12345
samprate=(1 0 0); /* Adjust the samprate to balance the strata */
strata acci_severity;
run;
data BalancedData_with_constant1;
set BalancedData1;
constant = 1;
run;
proc surveyselect data=BalancedData_with_constant1 NOPRINT seed=1
method=urs
samprate=1
OUTHITS
reps=&NumSamples1(repname=row)
out=BootSamp1;
strata acci_severity;
run;
proc freq data=BootSamp1;
tables acci_severity;
run;
proc surveyselect data=WORK.IMPORT NOPRINT out=BalancedData2
method=urs
seed=12345
samprate=(0 1 0); /* Adjust the samprate to balance the strata */
strata acci_severity;
run;
data BalancedData_with_constant2;
set BalancedData2;
constant = 1;
run;
proc surveyselect data=BalancedData_with_constant2 NOPRINT seed=1
method=urs
samprate=1
OUTHITS
reps=&NumSamples2(repname=row)
out=BootSamp2;
strata acci_severity;
run;
proc surveyselect data=WORK.IMPORT NOPRINT out=BalancedData3
method=urs
seed=12345
samprate=(0 0 0.9); /* Adjust the samprate to balance the strata */
strata acci_severity;
run;
data BalancedData_with_constant3;
set BalancedData3;
constant = 1;
run;
proc surveyselect data=BalancedData_with_constant3 NOPRINT seed=1
method=urs
samprate=1
OUTHITS
reps=&NumSamples3(repname=row)
out=BootSamp3;
strata acci_severity;
run;
proc freq data=BootSamp3;
tables acci_severity;
run;
data BootSamp;
set BootSamp1 BootSamp2 BootSamp3;
run;
proc freq data=BootSamp;
tables acci_severity;
run;
```

Appendix Fig 11 Task 2 Stage 1 Data Balancing Code

```
/* specifies CAS library information for the CAS table that you would like to score. You must modify the value to provide
the name of the library that contains the table to be scored. */
%let input_caslib_name = "CASUSER";
/* specifies the CAS table you would like to score. You must modify the value to provide the name of the input table, such
as "MyTable". Do not include an extension. */
%let input_table_name = "TWEETS";

/* specifies the column in the CAS table that contains a unique document identifier. You must modify the value to provide
the name of the document identifer column in the table. */
%let key_column = "ID";

/* specifies the column in the CAS table that contains the text data to score. You must modify the value to provide the
name of the text column in the table. */
%let document_column = "Text";

/* specifies the CAS library to write the score output tables. You must modify the value to provide the name of the library
that will contain the output tables that the score code produces. */
%let output_caslib_name = "CASUSER";

/* specifies the sentiment output CAS table to produce */
%let output_sentiment_table_name = "out_sentiment_1";

/* specifies the matches output CAS table to produce */
%let output_matches_table_name = "out_sent_matches_2";

/* specifies the features output CAS table to produce */
%let output_features_table_name = "out_sent_features_3";

/* specifies the language of the associated SAS Visual Text Analytics project. This should be set automatically to the
language you selected when you created your project. */
%let language = "ENGLISH";
/* specifies the hostname for the CAS server. This should be set automatically to the host for the associated SAS Visual
Text Analytics project. */
%let cas_server_hostname = "sas-cas-server-default-client";
/* specifies the port for the CAS server. This should be set automatically to the host for the associated SAS Visual Text
Analytics project. */
%let cas_server_port = 5570;

/* creates a session */
cas sascas1 host=&cas_server_hostname port=&cas_server_port;
libname sascas1 cas sessref=sascas1 datalimit=all;
/* calls the scoring action */
proc cas;
  session sascas1;
  loadactionset "sentimentAnalysis";
  action applySent;
    param
      table={caslib=&input_caslib_name, name=&input_table_name}
      docId=&key_column
      text=&document_column
      language=&language
      casOut={caslib=&output_caslib_name, name=&output_sentiment_table_name, replace=TRUE}
      matchOut={caslib=&output_caslib_name, name=&output_matches_table_name, replace=TRUE}
      featureOut={caslib=&output_caslib_name, name=&output_features_table_name, replace=TRUE}
    ;
  run;
quit;
```

Appendix Fig 12 Task 3 Sentiment Analysis Code